


RESEARCH

Open Access



Mean square cross error: performance analysis and applications in non-Gaussian signal processing

Yunxiang Zhang¹, Yuyang Zhao², Gang Wang¹ and Rui Xue^{3*} 

* Correspondence: xuerui@buaa.edu.cn

³School of Electronics and Information Engineering, Beihang University, Beijing, P. R. China
Full list of author information is available at the end of the article

Abstract

Most of the cost functions of adaptive filtering algorithms include the square error, which depends on the current error signal. When the additive noise is impulsive, we can expect that the square error will be very large. By contrast, the cross error, which is the correlation of the error signal and its delay, may be very small. Based on this fact, we propose a new cost function called the mean square cross error for adaptive filters, and provide the mean value and mean square performance analysis in detail. Furthermore, we present a two-stage method to estimate the closed-form solutions for the proposed method, and generalize the two-stage method to estimate the closed-form solution of the information theoretic learning methods, including least mean fourth, maximum correntropy criterion, generalized maximum correntropy criterion, and minimum kernel risk-sensitive loss. The simulations of the adaptive solutions and closed-form solution show the effectivity of the new method.

Keywords: Adaptive filter, Mean square error (MSE), Maximum correntropy criterion (MCC), Least mean fourth (LMF), Mean square cross error (MSCE)

1 Introduction

The mean square error (MSE) is probably the most widely used cost function for adaptive linear filters [1–5]. The MSE relies heavily on Gaussianity assumptions and performs well for Gaussian noise. Recently, information theoretic learning (ITL) has been proposed to process non-Gaussian noise. ITL uses the higher-order moments of the probability density function and may work well for non-Gaussian noise. Inspired by ITL, some cost functions, such as the maximum correntropy criterion (MCC) [6–11], improved least sum of exponentials (ILSE) [12], least mean kurtosis (LMK) [13], least mean fourth (LMF) [14–19], generalized MCC (GMCC) [20], and minimum kernel risk-sensitive loss (MKRSL) criterion [21, 22] have been presented.

The LMK and LMF are robust to sub-Gaussian noise. One typical sub-Gaussian distribution is the uniform distribution. The MCC and ILSE are robust to larger outliers or impulsive noise, which often take relatively more often values that are very close to zero or very large. This means that impulsive noise has a super-Gaussian distribution [23, 24].

Altogether, the distribution of the additive noise in linear filtering can be divided into three types: Gaussian, super-Gaussian, and sub-Gaussian. Super-Gaussian noise and sub-Gaussian noise are both non-Gaussian.

From the viewpoint of performance, for example, the steady error, the MSE, MCC, and LMF work well for Gaussian, super-Gaussian, and sub-Gaussian noise, respectively. The MSE demonstrates similar performance for the three types of noise under the same signal-to-noise ratio (SNR). For Gaussian noise, all the algorithms have similar steady errors. For super-Gaussian noise, the steady error comparison under the same SNR is $MCC < MSE < LMF$. For sub-Gaussian noise, the comparison is $LMF < MSE < MCC$.

Note that the cost functions of the above algorithms all include the square error, which is the correlation of the error signal. When impulsive noise is involved, we can expect that the square error will be very large. By contrast, the cross error (CE), which is the correlation of the error signal itself and its delay, may be very small for impulsive noise.

In our early work [25–27], we proposed the mean square cross prediction error to extract the desired signal in blind source separation (BSS), where the square cross prediction error was much smaller than the square prediction error. In this paper, we propose a new cost function called the mean square CE (MSCE) for adaptive filtering to process non-Gaussian noise. We expect that the proposed MSCE algorithm will perform well for non-Gaussian noise.

Note that the ITL methods can capture higher-order statistics of data. Thus, it is hard to directly obtain the corresponding closed-form solutions. We present a two-stage method to estimate the closed-form solutions for the LMF, MCC, GMCC, MKRSL, and MSCE.

The contributions of this paper are summarized as follows:

- i) We present a new cost function, that is, the MSCE, for adaptive filters, and provide the mean value and mean square performance analysis in detail.
- ii) We propose a two-stage method to estimate the closed-form solution of the proposed MSCE algorithm.
- iii) We generalize the two-stage method to estimate the closed-form solution of the LMF MCC, GMCC, and MKRSL algorithms.

The paper is organized as follows: In Section 2, the problem statement is explained in detail. In Section 3, the MSCE algorithm is presented with the adaptive algorithm and closed-form solution. In Section 4, the closed-form solution of the LMF, MCC, GMCC, and MKRSL are estimated. In Section 5, the mean behavior and mean square behavior of MSCE are analyzed. Simulations are provided in Section 6. Lastly, a conclusion is provided in Section 7.

2 Problem formulation

The absolute value of the normalized kurtosis may be considered as one measure of non-Gaussianity of the error signal. Several definitions for a random variable of zero means are presented as follows:

Definition 1 (normalized kurtosis)

The normalized kurtosis of random variable x is defined as

$$\kappa_4 = \frac{E\{|x|^4\}}{E^2\{|x|^2\}} - 3 \quad (1)$$

where x has zero mean.

Definition 2 (sub-Gaussian or platykurtic)

A distribution with negative normalized kurtosis is called sub-Gaussian, platykurtic, or short-tailed (e.g., uniform).

Definition 3 (super-Gaussian or leptokurtic)

A distribution with positive normalized kurtosis is called super-Gaussian, leptokurtic, or heavy-tailed (e.g., Laplacian).

Definition 4 (mesokurtic)

A zero-kurtosis distribution is called mesokurtic (e.g., Gaussian).

When the linear filtering problem is considered, there is an input vector $\mathbf{u} \in \mathbb{R}^M$, with unknown parameter $\mathbf{w}_o \in \mathbb{R}^M$ and desired response $d \in \mathbb{R}^1$. Data $d(i)$ are observed at each time point i by the linear regression model:

$$d(i) = \mathbf{w}_o^T \mathbf{u}(i) + v(i), \quad i = 1, 2, \dots, L, \quad (2)$$

where v is zero-mean background noise with variance σ_v^2 and L is the length of the sequence. The error signal for the linear filter is defined as

$$e(i) = d(i) - \mathbf{w}^T \mathbf{u}(i), \quad (3)$$

where \mathbf{w} is the estimate of \mathbf{w}_o . The distribution of the additive noise in linear filtering can be divided into three types: Gaussian, super-Gaussian, and sub-Gaussian. Super-Gaussian noise and sub-Gaussian noise are both non-Gaussian.

In this research, we made the following assumptions:

A1) The additive noise is white, that is,

$$E\{v(i)v(j)\} = 0, i \neq j. \quad (4)$$

A2) Inputs $\mathbf{u}(t)$ at different time moments (i, j) are uncorrelated:

$$E\{\mathbf{u}^H(i)\mathbf{u}(j)\} = E\{\mathbf{u}^T(i)\mathbf{u}(j)\} = 0, i \neq j. \quad (5)$$

A3) The inputs and additive noise at different time moments (i, j) are uncorrelated:

$$E\{\mathbf{u}^H(i)v(j)\} = 0, i \neq j. \quad (6)$$

The linear filtering algorithms of the MSE, MCC, and LMF are as follows: the cost function based on the MSE is given by

$$J_{MSE}(\mathbf{w}) = E\{e^2\}, \quad (7)$$

where E denotes the expectation operator. The gradient is defined as

$$\frac{\partial J_{MSE}}{\partial \mathbf{w}} = -E\{e\mathbf{u}\}. \quad (8)$$

At the stationary point, $E\{e\mathbf{u}\} = 0$. The closed-form solution denoted by \mathbf{w}_{MSE} is given by the Wiener–Hopf equation:

$$\begin{aligned} \mathbf{w}_{MSE} &= [\mathbf{R}(\mathbf{u})]^{-1}[\mathbf{R}_{du}], \\ \mathbf{R}(\mathbf{u}) &= E\{\mathbf{u}\mathbf{u}^H\}, \\ \mathbf{R}_{du} &= E\{d\mathbf{u}\}. \end{aligned} \quad (9)$$

The corresponding stochastic gradient descent, or LMS, algorithm is

$$\mathbf{w}_{LMS}(i+1) = \mathbf{w}_{LMS}(i) + \mu e(i)\mathbf{u}(i) \quad (10)$$

where μ denotes the step size and $\mu > 0$.

The cost function based on the LMF is given by

$$J_{LMF}(\mathbf{w}) = E\{e^4\}. \quad (11)$$

The corresponding stochastic gradient descent algorithm is

$$\mathbf{w}_{LMF}(i+1) = \mathbf{w}_{LMF}(i) + \mu e^3(i)\mathbf{u}(i). \quad (12)$$

The cost function based on the correntropy of the error, also called the MCC, is given by

$$J_{MCC}(\mathbf{w}) = E\left\{\exp\left(-\frac{e^2}{2\sigma^2}\right)\right\} \quad (13)$$

where σ denotes the kernel bandwidth. The corresponding stochastic gradient descent algorithm is

$$\mathbf{w}_{MCC}(i+1) = \mathbf{w}_{MCC}(i) + \mu \exp\left(-\frac{e^2(i)}{2\sigma^2}\right)e(i)\mathbf{u}(i). \quad (14)$$

The cost function based on the GMCC is given by

$$J_{GMCC}(\mathbf{w}) = \gamma_{\alpha, \beta} \{1 - E[\exp(-\lambda|e(i)|^\alpha)]\}. \quad (15)$$

The corresponding stochastic gradient descent algorithm is

$$\mathbf{w}_{GMCC}(i+1) = \mathbf{w}_{GMCC}(i) + \mu\lambda\alpha \exp(-\lambda|e(i)|^\alpha)|e(i)|^{\alpha-1} \text{sign}(e(i))\mathbf{u}(i). \quad (16)$$

The cost function based on the MKRSL is given by

$$J_{MKRSL}(\mathbf{w}) = \frac{1}{L\lambda} \sum_{i=1}^L \exp(\lambda(1-\kappa_\sigma(e(i)))),$$

$$\kappa_\sigma(e(i)) = \exp\left(-\frac{e^2(i)}{2\sigma^2}\right). \quad (17)$$

The corresponding stochastic gradient descent algorithm is

$$\mathbf{w}_{MKRSL}(i+1) = \mathbf{w}_{MKRSL}(i) + \frac{\mu}{\sigma^2} \exp(\lambda(1-\kappa_\sigma(e(i))))\kappa_\sigma(e(i))e(i)\mathbf{u}(i). \quad (18)$$

3 Methods

3.1 Adaptive algorithm of the MSCE

The CE can be expressed as $e(i)e(i-q)$, where q denotes the error delay. Because the CE may be negative, we provide a new cost function, that is, the MSCE, as

$$J_{MSCE}(\mathbf{w}) = \frac{1}{2} E\{e^2(i)e^2(i-q)\} \quad (19)$$

where

$$\begin{aligned} e(i) &= d(i) - \mathbf{w}^T \mathbf{u}(i), \\ e(i-q) &= d(i-q) - \mathbf{w}^T \mathbf{u}(i-q). \end{aligned} \quad (20)$$

The gradient of the MSCE can be derived as

$$\frac{\partial J_{MSCE}}{\partial \mathbf{w}} = -E\{e^2(i-q)[e(i)\mathbf{u}(i)] + e^2(i)[e(i-q)\mathbf{u}(i-q)]\}. \quad (21)$$

Then, the corresponding stochastic gradient descent algorithm is

$$\mathbf{w}_{MSCE}(i+1) = \mathbf{w}_{MSCE}(i) + \mu e^2(i-q)e(i)\mathbf{u}(i) + \mu e^2(i)e(i-q)\mathbf{u}(i-q). \quad (22)$$

Equation (19) may not be robust against outliers. We provide the generalized MSCE (GMSCE) as

$$J_{MSCE}(\mathbf{w}) = E\{G[e(i)]G[e(i-q)]\} \quad (23)$$

where $G(x)$ is an x^2 -like function. The stochastic gradient descent algorithm for the GMSCE is

$$\begin{aligned} \mathbf{w}_{GMSCE}(i+1) &= \mathbf{w}_{GMSCE}(i) \\ &+ \mu G[e(i)]g[e(i-q)]\mathbf{u}(i-q) \\ &+ \mu G[e(i-q)]g[e(i)]\mathbf{u}(i) \end{aligned} \quad (24)$$

where $g(\cdot)$ is the derivative of $G(\cdot)$.

3.2 A suitable criterion for cost function

Combining the references of ICA [23, 24, 28–30] with those of the MCC [6–11], ILSE [12], and LMF [14–19], we determined that there are three cost functions in the fast ICA [30] algorithm:

$$G_1(u) = \frac{1}{\alpha_1} \log \cosh(\alpha_1 u), \quad (25)$$

$$G_2(u) = -\frac{1}{\alpha_2} \exp\left(-\frac{\alpha_2 u^2}{2\sigma^2}\right), \quad (26)$$

$$G_3(u) = \frac{1}{4} u^4. \quad (27)$$

$G_2(u)$ is used to separate the super-Gaussian source in ICA, and works as the cost function of the MCC. $G_3(u)$ is used to separate the sub-Gaussian source in ICA when there are no outliers, and works as the cost function of the LMF. $G_1(u)$ has not been used in adaptive filtering, but $\cosh(\alpha_1 u)$ works as the cost function of the ILSE.

This motivated us to explore ICA or BSS algorithms to determine a suitable criterion for adaptive filtering. Here, we use $G_1(u)$ in the proposed GMSCE algorithm:

$$G(x) = \frac{1}{\alpha} \log \cosh(\alpha x) \quad (28)$$

where $1 \leq \alpha \leq 2$, and $\alpha = 1$ in the simulations. The derivative of $G(x)$ is

$$g(x) = \tanh(\alpha x). \quad (29)$$

Substituting (28)–(29) into (24) with $\alpha = 1$, we obtain

$$\begin{aligned} \mathbf{w}_{GMSCE}(i+1) &= \mathbf{w}_{GMSCE}(i) \\ &+ \mu \log \cosh[e(i)] \tanh[e(i-q)] \mathbf{u}(i-q) \\ &+ \mu \log \cosh[e(i-q)] \tanh[e(i)] \mathbf{u}(i). \end{aligned} \quad (30)$$

3.3 Closed-form solution of the MSCE and GMSCE

We can estimate the closed-form solution of the MSCE from the stationary point $\partial J_{MSCE} / \partial \mathbf{w} = 0$:

$$\frac{\partial J_{MSCE}}{\partial \mathbf{w}} = -E\{e^2(i-q)[e(i)\mathbf{u}(i)] + e^2(i)[e(i-q)\mathbf{u}(i-q)]\} = 0. \quad (31)$$

Substituting (20) into (31), we obtain

$$\begin{aligned} \frac{\partial J_{MSCE}}{\partial \mathbf{w}} &= -E\{e^2(i-q)[d(i)\mathbf{u}(i) - \mathbf{u}(i)\mathbf{u}^T(i)\mathbf{w}]\} \\ &- E\{e^2(i)[d(i-q)\mathbf{u}(i-q) - \mathbf{u}(i-q)\mathbf{u}^T(i-q)\mathbf{w}]\} = 0. \end{aligned} \quad (32)$$

It is difficult to solve \mathbf{w} from (32) because $e^2(i)$ contains the second-order term of \mathbf{w} . We present a two-stage method to estimate \mathbf{w} .

In the first stage, we estimate $e(i)$ and $e(i-q)$ from (9):

$$\begin{aligned} \mathbf{w}_{MSE} &= [R(\mathbf{u})]^{-1} [R_{du}], \\ \hat{e}(i-q) &= d(i-q) - \mathbf{w}_{MSE}^T \mathbf{u}(i-q), \\ \hat{e}(i) &= d(i) - \mathbf{w}_{MSE}^T \mathbf{u}(i), i = 1, 2, \dots, L \end{aligned} \quad (33)$$

where $\hat{e}(i)$ and $\hat{e}(i-q)$ are the estimates of $e(i)$ and $e(i-q)$, respectively.

In the second stage, we estimate \mathbf{w} from (32).

If we define

$$F_1(\mathbf{w}) = \frac{\partial J_{MSCE}}{\partial \mathbf{w}},$$

then we can rewrite (33) as

$$\begin{aligned} F_1(\mathbf{w}) = & -E\{\hat{e}^2(i-q)[d(i)\mathbf{u}(i)-\mathbf{u}(i)\mathbf{u}^T(i)\mathbf{w}]\} \\ & -E\{\hat{e}^2(i)[d(i-q)\mathbf{u}(i-q)-\mathbf{u}(i-q)\mathbf{u}^T(i-q)\mathbf{w}]\} = 0 \end{aligned} \quad (34)$$

where $F_1(\mathbf{w})$ is the estimate of $F_1(\mathbf{w})$.

Note that the expectation can be estimated by averaging over the samples using

$$E\{f(x)\} \approx \frac{1}{L} \sum_{i=1}^L f(x_i). \quad (35)$$

Equation (34) can be estimated by

$$\begin{aligned} F_1(\mathbf{w}) \approx & -\frac{1}{L} \sum_{i=q+1}^{L+q} \{\hat{e}^2(i-q)[d(i)\mathbf{u}(i)-\mathbf{u}(i)\mathbf{u}^T(i)\mathbf{w}]\} \\ & -\frac{1}{L} \sum_{i=q+1}^{L+q} \{\hat{e}^2(i)[d(i-q)\mathbf{u}(i-q)-\mathbf{u}(i-q)\mathbf{u}^T(i-q)\mathbf{w}]\} \\ = & -\frac{1}{L} \sum_{i=q+1}^{L+q} \{\hat{e}^2(i-q)d(i)\mathbf{u}(i) + \hat{e}^2(i)d(i-q)\mathbf{u}(i-q)\} \\ & +\frac{1}{L} \sum_{i=q+1}^{L+q} \{\hat{e}^2(i-q)\mathbf{u}(i)\mathbf{u}^T(i) + \hat{e}^2(i)\mathbf{u}(i-q)\mathbf{u}^T(i-q)\}\mathbf{w} = 0. \end{aligned} \quad (36)$$

Furthermore, we have

$$\begin{aligned} & \frac{1}{L} \sum_{i=q+1}^{L+q} \{\hat{e}^2(i-q)d(i)\mathbf{u}(i) + \hat{e}^2(i)d(i-q)\mathbf{u}(i-q)\} = \\ & \frac{1}{L} \sum_{i=q+1}^{L+q} \{\hat{e}^2(i-q)\mathbf{u}(i)\mathbf{u}^T(i) + \hat{e}^2(i)\mathbf{u}(i-q)\mathbf{u}^T(i-q)\}\mathbf{w}. \end{aligned} \quad (37)$$

If we define $\mathbf{R}_{du}(q)$ and $\mathbf{R}_{uu}(q)$ as

$$\begin{aligned} \mathbf{R}_{du}(q) &= E\{e^2(i-q)d(i)\mathbf{u}(i) + e^2(i)d(i-q)\mathbf{u}(i-q)\}, \\ \mathbf{R}_{uu}(q) &= E\{e^2(i-q)\mathbf{u}(i)\mathbf{u}^T(i) + e^2(i)\mathbf{u}(i-q)\mathbf{u}^T(i-q)\}, \end{aligned} \quad (38)$$

then we can estimate them as

$$\begin{aligned} \mathbf{R}_{du}(q) &= \frac{1}{L} \sum_{i=q+1}^{L+q} \{\hat{e}^2(i-q)d(i)\mathbf{u}(i) + \hat{e}^2(i)d(i-q)\mathbf{u}(i-q)\}, \\ \mathbf{R}_{uu}(q) &= \frac{1}{L} \sum_{i=q+1}^{L+q} \{\hat{e}^2(i-q)\mathbf{u}(i)\mathbf{u}^T(i) + \hat{e}^2(i)\mathbf{u}(i-q)\mathbf{u}^T(i-q)\}, \end{aligned} \quad (39)$$

where $\mathbf{R}_{du}(q)$ and $\mathbf{R}_{uu}(q)$ are the estimates of $\mathbf{R}_{du}(q)$ and $\mathbf{R}_{uu}(q)$, respectively.

We can estimate the closed-form solution of the MSCE as

$$\mathbf{w}_{close_MSCE} = [\mathbf{R}_{uu}(q)]^{-1}[\mathbf{R}_{du}(q)]. \quad (40)$$

Note that $\tanh(x) \approx x$ when x is small. We can estimate the closed-form solution of the GMSCE in the same way.

If we define $\mathbf{R}_{Gdu}(q)$ and $\mathbf{R}_{Guu}(q)$ as

$$\begin{aligned} \mathbf{R}_{Gdu}(q) &= E\{G(e(i-q))d(i)\mathbf{u}(i) + G(e(i))d(i-q)\mathbf{u}(i-q)\}, \\ \mathbf{R}_{Guu}(q) &= E\{G(e(i-q))\mathbf{u}(i)\mathbf{u}^T(i) + G(e(i))\mathbf{u}(i-q)\mathbf{u}^T(i-q)\}, \end{aligned} \quad (41)$$

then we can estimate them as

$$\begin{aligned} \mathbf{R}_{Gdu}(q) &= \frac{1}{L} \sum_{i=q+1}^{L+q} \{G(\hat{e}(i-q))d(i)\mathbf{u}(i) + G(\hat{e}(i))d(i-q)\mathbf{u}(i-q)\}, \\ \mathbf{R}_{Guu}(q) &= \frac{1}{L} \sum_{i=q+1}^{L+q} \{G(\hat{e}(i-q))\mathbf{u}(i)\mathbf{u}^T(i) + G(\hat{e}(i))\mathbf{u}(i-q)\mathbf{u}^T(i-q)\}. \end{aligned} \quad (42)$$

We can estimate the closed-form solution of the GMSCE as

$$\mathbf{w}_{close_GMSCE} = [\mathbf{R}_{Guu}(q)]^{-1}[\mathbf{R}_{Gdu}(q)]. \quad (43)$$

4 Closed-form solution of the LMF, MCC, GMCC, and MKRSL

Based on the two-stage methods, we can also estimate the closed solution of the LMF, MCC, GMCC, and MKRSL algorithms as follows.

4.1 Closed-form solution of the LMF

We can estimate the closed-form solution of the LMF from the stationary point $\partial J_{LMF} / \partial \mathbf{w} = 0$:

$$\frac{\partial J_{LMF}}{\partial \mathbf{w}} = -E\{e^2(i)[d(i)\mathbf{u}(i) - \mathbf{w}^T \mathbf{u}(i)]\mathbf{u}(i)\}. \quad (44)$$

In the first stage, we estimate $e(i)$ from (9):

$$\begin{aligned} \mathbf{w}_{MSE} &= [R(\mathbf{u})]^{-1}[R_{du}] \\ \hat{e}(i) &= d(i) - \mathbf{w}_{MSE}^T \mathbf{u}(i), i = 1, 2, \dots, L. \end{aligned} \quad (45)$$

In the second stage, we estimate \mathbf{w} from (44).

If we define

$$F_2(\mathbf{w}) = \frac{\partial J_{LMF}}{\partial \mathbf{w}}, \quad (46)$$

then we can rewrite (44) as

$$F_2(\mathbf{w}) = -E\{\hat{e}^2(i)[d(i)\mathbf{u}(i) - \mathbf{u}(i)\mathbf{u}^T(i)\mathbf{w}]\} \quad (47)$$

where $F_2(\mathbf{w})$ is the estimate of $F_2(\mathbf{w})$.

With the help of (35), Eq. (47) can be estimated by

$$F_1(\mathbf{w}) \approx -\frac{1}{L} \sum_{i=1}^L \{\hat{e}^2(i)[d(i)\mathbf{u}(i) - \mathbf{u}(i)\mathbf{u}^T(i)\mathbf{w}]\}. \quad (48)$$

If we define \mathbf{R}_{du2} and \mathbf{R}_{uu2} as

$$\begin{aligned}\mathbf{R}_{du2} &= E\{e^2(i)d(i)\mathbf{u}(i)\}, \\ \mathbf{R}_{uu2} &= E\{e^2(i)\mathbf{u}(i)\mathbf{u}^T(i)\},\end{aligned}\quad (49)$$

then we can estimate them as

$$\begin{aligned}\hat{\mathbf{R}}_{du2} &= \frac{1}{L} \sum_{i=1}^L \{\hat{e}^2(i)d(i)\mathbf{u}(i)\}, \\ \hat{\mathbf{R}}_{uu2} &= \frac{1}{L} \sum_{i=1}^L \{\hat{e}^2(i)\mathbf{u}(i)\mathbf{u}^T(i)\},\end{aligned}\quad (50)$$

where $\hat{\mathbf{R}}_{du2}$ and $\hat{\mathbf{R}}_{uu2}$ are the estimates of \mathbf{R}_{du2} and \mathbf{R}_{uu2} , respectively.

We can estimate the closed-form solution of the LMF as

$$\mathbf{w}_{close_LMF} = [\hat{\mathbf{R}}_{uu2}]^{-1}[\hat{\mathbf{R}}_{du2}]. \quad (51)$$

4.2 Closed-form solution of the MCC

We can estimate the closed-form solution of the MCC from the stationary point $\partial J_{MCC}/\partial \mathbf{w} = 0$:

$$\frac{\partial J_{MCC}}{\partial \mathbf{w}} = \frac{1}{\sigma^2} E\{\exp(-e^2(i)/2\sigma^2)[d - \mathbf{w}^T \mathbf{u}(i)]\mathbf{u}(i)\} = 0. \quad (52)$$

In the first stage, we estimate $e(i)$ from (45). In the second stage, we estimate \mathbf{w} from (52).

If we define

$$F_3(\mathbf{w}) = \sigma^2 \frac{\partial J_{MCC}}{\partial \mathbf{w}}, \quad (53)$$

then we can rewrite (52) as

$$F_3(\mathbf{w}) = E\{\exp(-\hat{e}^2(i)/2\sigma^2)[d(i)\mathbf{u}(i) - \mathbf{u}(i)\mathbf{u}^T(i)\mathbf{w}]\} \quad (54)$$

where $F_3(\mathbf{w})$ is the estimate of $F_3(\mathbf{w})$.

With the help of (35), Eq. (54) can be estimated by

$$F_1(\mathbf{w}) \approx -\frac{1}{L} \sum_{i=1}^L \{\exp(-\hat{e}^2(i)/2\sigma^2)[d(i)\mathbf{u}(i) - \mathbf{u}(i)\mathbf{u}^T(i)\mathbf{w}]\}. \quad (55)$$

Denote by

$$\begin{aligned}\hat{\mathbf{R}}_{du3} &= \frac{1}{L} \sum_{i=1}^L \{\exp(-\hat{e}^2(i)/2\sigma^2)d(i)\mathbf{u}(i)\}, \\ \hat{\mathbf{R}}_{uu3} &= \frac{1}{L} \sum_{i=1}^L \{\exp(-\hat{e}^2(i)/2\sigma^2)\mathbf{u}(i)\mathbf{u}^T(i)\}.\end{aligned}\quad (56)$$

We can estimate the closed-form solution of the MCC as

$$\mathbf{w}_{close_MCC} = [\hat{\mathbf{R}}_{uu3}]^{-1}[\hat{\mathbf{R}}_{du3}] \quad (57)$$

4.3 Closed-form solution of the GMCC

The closed-form solution of the GMCC is given by [20]:

$$\begin{aligned}\mathbf{w}_{close_GMCC} &= [E(h_1(e(i))\mathbf{u}(i)\mathbf{u}^T(i))]^{-1}[E(h_1(e(i))d(i)\mathbf{u}(i))], \\ h_1(e(i)) &= \exp(-\lambda|e(i)|^\alpha)|e(i)|^{\alpha-2}.\end{aligned}\quad (58)$$

In the first stage, we estimate $e(i)$ from (45), and we have

$$h_1(\hat{e}(i)) = \exp(-\lambda|\hat{e}(i)|^\alpha)|\hat{e}(i)|^{\alpha-2}. \quad (59)$$

In the second stage, we estimate \mathbf{w} from (59). Denote by

$$\begin{aligned}\mathbf{R}_{du4} &= \frac{1}{L} \sum_{i=1}^L \{h_1(\hat{e}(i))d(i)\mathbf{u}(i)\}, \\ \mathbf{R}_{uu4} &= \frac{1}{L} \sum_{i=1}^L \{h_1(\hat{e}(i))\mathbf{u}(i)\mathbf{u}^T(i)\}.\end{aligned}\quad (60)$$

We can estimate the closed-form solution of the GMCC as

$$\mathbf{w}_{close_GMCC} = [\mathbf{R}_{uu4}]^{-1}[\mathbf{R}_{du4}] \quad (61)$$

4.4 Closed-form solution of the MKRSL

The closed-form solution of the MKRSL is given by [22]:

$$\begin{aligned}\mathbf{w}_{close_MKRSL} &= [E(h_2(e(i))\mathbf{u}(i)\mathbf{u}^T(i))]^{-1}[E(h_2(e(i))d(i)\mathbf{u}(i))], \\ h_2(e(i)) &= \exp(\lambda(1-\kappa_\sigma(e(i))))\kappa_\sigma(e(i)).\end{aligned}\quad (62)$$

In the first stage, we estimate $e(i)$ from (45), and we have

$$h_2(\hat{e}(i)) = \exp(\lambda(1-\kappa_\sigma(\hat{e}(i))))\kappa_\sigma(\hat{e}(i)). \quad (63)$$

In the second stage, we estimate \mathbf{w} from (63). Denote by

$$\begin{aligned}\mathbf{R}_{du5} &= \frac{1}{L} \sum_{i=1}^L \{h_2(\hat{e}(i))d(i)\mathbf{u}(i)\}, \\ \mathbf{R}_{uu5} &= \frac{1}{L} \sum_{i=1}^L \{h_2(\hat{e}(i))\mathbf{u}(i)\mathbf{u}^T(i)\}.\end{aligned}\quad (64)$$

We can estimate the closed-form solution of the MKRSL as

$$\mathbf{w}_{close_MKRSL} = [\mathbf{R}_{uu5}]^{-1}[\mathbf{R}_{du5}]. \quad (65)$$

5 Performance analysis of MSCE

5.1 Mean value behavior

To compare the performance of the MSE and MSCE, we define the total weight error as

$$E\{|\varepsilon(i)|^2\} = E\{\varepsilon^T(i)\varepsilon(i)\} \quad (66)$$

where

$$\varepsilon(i) = \mathbf{w}_o - \mathbf{w}_{MSCE}(i). \quad (67)$$

Substituting (22) into (67), we obtain

$$\begin{aligned}
\varepsilon(i+1) &= \varepsilon(i) - \mu e^2(i-q) [\mathbf{u}(i) \mathbf{u}^T(i) \varepsilon(i) + \mathbf{u}(i) v(i)] \\
&\quad - \mu e^2(i) [\mathbf{u}(i-q) \mathbf{u}^T(i-q) \varepsilon(i) + \mathbf{u}(i-q) v(i-q)] \\
&= [\mathbf{I} - \mu e^2(i) \mathbf{u}(i-q) \mathbf{u}^T(i-q) - \mu e^2(i-q) \mathbf{u}(i) \mathbf{u}^T(i)] \varepsilon(i) \\
&\quad - [\mu e^2(i) v(i-q) \mathbf{u}(i-q) + \mu e^2(i-q) v(i) \mathbf{u}(i)]
\end{aligned} \tag{68}$$

where \mathbf{I} is the identity matrix.

Substituting (39) into (67), we obtain

$$\begin{aligned}
E\{\varepsilon(i+1)\} &= [\mathbf{I} - \mu \mathbf{R}_{uu}(q)] E\{\varepsilon(i)\} \\
&\quad - \mu E[e^2(i) v(i-q) \mathbf{u}(i-q) + e^2(i-q) v(i) \mathbf{u}(i)].
\end{aligned} \tag{69}$$

Note that $\mathbf{R}_{uu}(q)$ is positive definite; thus, (69) is stable for a sufficiently small step size μ .

The eigenvalue decomposition of $\mathbf{R}_{uu}(q)$ is $\mathbf{R}_{uu}(q) = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$. Then, we have the first-order moment of $\varepsilon(i)$

$$\begin{aligned}
E\{\varepsilon(i)\} &\approx \mathbf{U} [\mathbf{I} - \mu \mathbf{\Lambda}]^i \mathbf{U}^T \varepsilon(0), \\
\mathbf{\Lambda} &= \text{diag}\{d_1, d_2, \dots, d_M\}.
\end{aligned} \tag{70}$$

Let d_{\max} denote the maximum eigenvalue of $\mathbf{R}_{uu}(q)$. The step size should be selected as

$$0 < \mu < 2/d_{\max} \tag{71}$$

so that the iterations will converge.

5.2 Mean square behavior

If we define

$$\begin{aligned}
\delta(i) &= \mathbf{U}^T \varepsilon(i), \\
\mathbf{U} &= [\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_M],
\end{aligned} \tag{72}$$

where \mathbf{U}_m is the m th column of \mathbf{U} , then we can rewrite (56) as

$$\begin{aligned}
\delta(i+1) &\approx [\mathbf{I} - \mu \mathbf{\Lambda}] \delta(i) \\
&\quad - \mu \mathbf{U}^T [e^2(i) v(i-q) \mathbf{u}(i-q) + e^2(i-q) v(i) \mathbf{u}(i)],
\end{aligned} \tag{73}$$

which is composed of M decoupled difference equations:

$$\begin{aligned}
\delta_m(i+1) &\approx (1 - \mu d_m) \delta_m(i) \\
&\quad - \mu \mathbf{U}_m^T [e^2(i) v(i-q) \mathbf{u}(i-q) + e^2(i-q) v(i) \mathbf{u}(i)], \\
m &= 1, 2, \dots, M.
\end{aligned} \tag{74}$$

The second-order moment of $\delta_m(i+1)$ can be derived from (74) as

$$\begin{aligned}
E\{\delta_m^2(i+1)\} &\approx (1 - \mu d_m)^2 E\{\delta_m^2(i)\} \\
&\quad + \mu^2 E\{e^2(i) v^2(i-q) \mathbf{U}_m^T e^2(i) \mathbf{u}(i-q) \mathbf{u}^T(i-q) \mathbf{U}_m\} \\
&\quad + \mu^2 E\{e^2(i-q) v^2(i) \mathbf{U}_m^T e^2(i-q) \mathbf{u}(i) \mathbf{u}^T(i) \mathbf{U}_m\}.
\end{aligned} \tag{75}$$

From (2), (3), and (67), we obtain

$$\begin{aligned}
e(i) &= \varepsilon^T(i) \mathbf{u}(i) + v(i), \\
e(i-q) &= \varepsilon^T(i) \mathbf{u}(i-q) + v(i-q).
\end{aligned} \tag{76}$$

Thus, we have

$$\begin{aligned} e^2(i) &= \mathbf{u}^T(i) \varepsilon(i) \varepsilon^T(i) \mathbf{u}(i) + v^2(i) \approx v^2(i), \\ e^2(i-q) &= \mathbf{u}^T(i-q) \varepsilon(i) \varepsilon^T(i) \mathbf{u}(i-q) + v^2(i-q) \approx v^2(i-q). \end{aligned} \quad (77)$$

Substituting (77) into (75) yields

$$\begin{aligned} E\{\delta_m^2(i+1)\} &\approx (1-\mu d_m)^2 E\{\delta_m^2(i)\} \\ &+ \mu^2 E\{e^2(i) e^2(i-q) \mathbf{U}_m^T e^2(i) \mathbf{u}(i-q) \mathbf{u}^T(i-q) \mathbf{U}_m\} \\ &+ \mu^2 E\{e^2(i-q) e^2(i) \mathbf{U}_m^T e^2(i-q) \mathbf{u}(i) \mathbf{u}^T(i) \mathbf{U}_m\}. \end{aligned} \quad (78)$$

Let

$$r_{uu}(q) = e^2(i-q) \mathbf{u}(i) \mathbf{u}^T(i) + e^2(i) \mathbf{u}(i-q) \mathbf{u}^T(i-q), \quad (79)$$

then we have

$$\mathbf{R}_{uu}(q) = E\{r_{uu}(q)\}. \quad (80)$$

Substituting (79) into (80) yields

$$\begin{aligned} E\{\delta_m^2(i+1)\} &\approx (1-\mu d_m)^2 E\{\delta_m^2(i)\} \\ &+ \mu^2 E\{e^2(i) e^2(i-q) \mathbf{U}_m^T [r_{uu}(q)] \mathbf{U}_m\} \\ &\approx (1-\mu d_m)^2 E\{\delta_m^2(i)\} \\ &+ \mu^2 E\{e^2(i) e^2(i-q)\} E\{\mathbf{U}_m^T [r_{uu}(q)] \mathbf{U}_m\} \\ &= (1-\mu d_m)^2 E\{\delta_m^2(i)\} \\ &+ \mu^2 E\{e^2(i) e^2(i-q)\} \mathbf{U}_m^T \mathbf{R}_{uu}(q) \mathbf{U}_m. \end{aligned} \quad (81)$$

Note that $\mathbf{U}_m^T \mathbf{R}_{uu}(q) \mathbf{U}_m = d_m$, and we have

$$\begin{aligned} E\{\delta_m^2(i+1)\} &\approx (1-\mu d_m)^2 E\{\delta_m^2(i)\} \\ &+ \mu^2 E\{e^2(i) e^2(i-q)\} d_m. \end{aligned} \quad (82)$$

Then the second-order moment of $\delta_m(i+1)$ can be rewritten as

$$\begin{aligned} E\{\delta_m^2(i)\} &\approx (1-\mu d_m)^{2i} E\{\delta_m^2(0)\} \\ &+ \sum_{j=0}^i (1-\mu d_m)^{2j} \mu^2 d_m E\{e^2(i) e^2(i-q)\} \\ &= (1-\mu d_m)^{2i} E\{\delta_m^2(0)\} \\ &+ \frac{1-(1-\mu d_m)^{2i}}{(2-\mu d_m)\mu d_m} \mu^2 d_m E\{e^2(i) e^2(i-q)\} \\ &= (1-\mu d_m)^{2i} E\{\delta_m^2(0)\} \\ &+ \frac{1-(1-\mu d_m)^{2i}}{(2-\mu d_m)} \mu E\{e^2(i) e^2(i-q)\} \\ &\approx (1-\mu d_m)^{2i} E\{\delta_m^2(0)\} \\ &+ \frac{1-(1-\mu d_m)^{2i}}{(2-\mu d_m)} \mu E\{v^2(i) v^2(i-q)\}. \end{aligned} \quad (83)$$

Thus, the steady state error of the MSCE algorithm is given by

$$\lim_{i \rightarrow \infty} \sum_{m=1}^M E\{\delta_m^2(i)\} \approx \sum_{m=1}^M \frac{1}{(2-\mu d_m)} \mu E\{v^2(i) v^2(i-q)\}. \quad (84)$$

When $\mu < 1/d_{\max}$, we have

$$\lim_{i \rightarrow \infty} \sum_{m=1}^M E\{\delta_m^2(i)\} \approx \frac{1}{2} \mu ME\{v^2(i)v^2(i-q)\}. \quad (85)$$

For comparison, we can write the steady state error of the MSE algorithm as

$$\lim_{i \rightarrow \infty} \sum_{m=1}^M E\{\delta_m^2(i)\} \approx \frac{1}{2} \mu ME\{v^2(i)\}. \quad (86)$$

For impulsive noise, its variance $E\{v^2(i)\}$ may be very large, but its MSCE $E\{v^2(i)v^2(i-q)\}$ may be small. Thus, the proposed MSCE algorithm may have a smaller steady error than the MSE for impulsive noise.

5.3 Selection of delay q

After obtaining the estimate of $e(i)$ using (33), we can estimate the MSCE for $q = 1, 2, \dots, Q$:

$$J_{MSCE}(\mathbf{w}, q) \approx \frac{1}{2L} \sum_{i=q+1}^{L+q} \{\hat{e}^2(i-q)\hat{e}^2(i)\} \quad (87)$$

where J_{MSCE} is the estimate of J_{MSCE} . Because the mean-square performance of the MSCE algorithm is proportional to $E\{v^2(i)v^2(i-q)\}$ according to (85), we should select q with the smallest J_{MSCE} in (87).

6 Simulation results and discussion

In this section, the performance of the MSE, MSCE, GMSCE, LMF, MCC, GMCC, and MKRSL will be evaluated by simulations. All the simulation points were averaged over 100 independent runs. The performance of the adaptive solution was estimated by the steady-state mean-square deviation (MSD)

$$MSD = \lim_{i \rightarrow \infty} E\{\|\mathbf{w}_o - \mathbf{w}(i)\|_2^2\}. \quad (88)$$

The performance of the closed-form solution is

$$MSD = \|\mathbf{w}_o - \mathbf{w}\|_2^2. \quad (89)$$

We concluded that the smaller the MSD, the better the performance.

6.1 Closed-form solutions comparison

The closed-form solutions of the MSE, MSCE, GMSCE, LMF, MCC, GMCC, and MKRSL are expressed by (9), (40), (43), (51), (57), (61), and (65), respectively. The GMCC with $\alpha = 2, 4$, and 6 are denoted by GMCC1, GMCC2, and GMCC3, respectively. The MKRSL with $\lambda = 0.1$ and 32 are denoted by MKRSL1 and MKRSL2, respectively.

In the experiments, we compared the MSDs of the closed-form solutions of the ten algorithms with different non-Gaussian noises. The input filter order was $M = 5$, and the sample size had length $L = 3000$. When the SNRs are ranged from -20 to 20 dB, we obtain similar performance comparisons. Here the SNR was set to 6 dB.

Figures 1 and 2 partly show the four types of sub- and super-Gaussian noise, respectively.

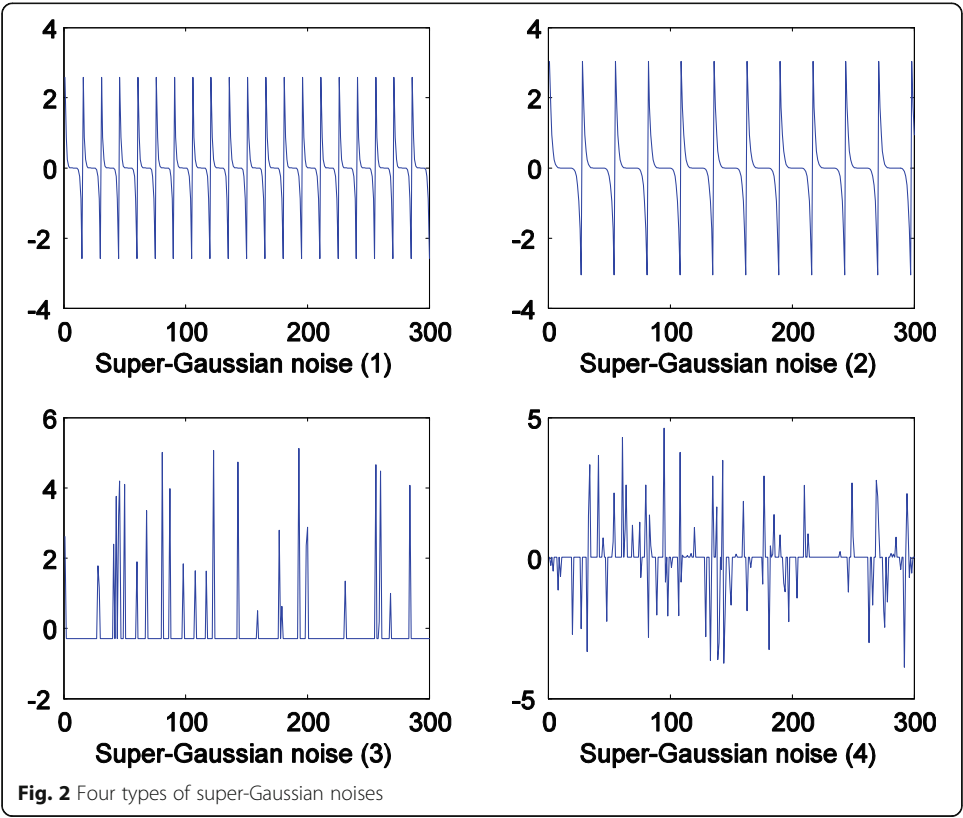
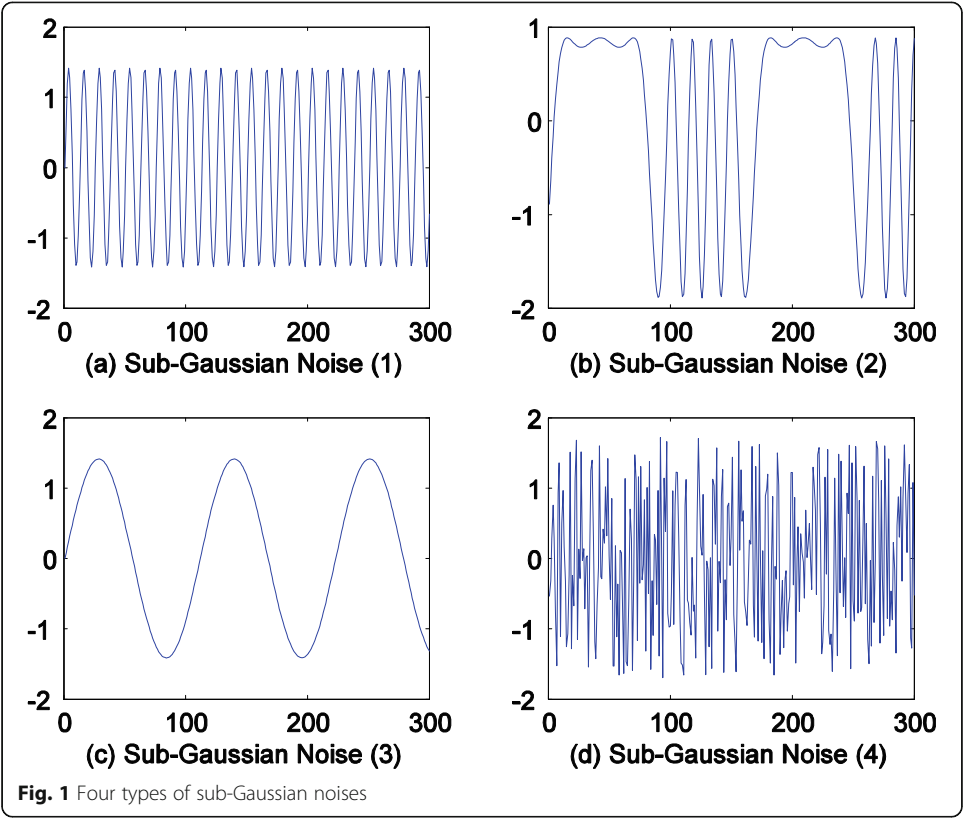


Figure 1a–c shows the periodic noises, and Fig. 1d shows the noise with uniform distribution. The kurtoses of the noises shown in Fig. 1a–d are -1.5 , -1.0 , -1.4 , and -1.2 , respectively.

Figure 2a, b shows the periodic super-Gaussian noises, and Fig. 2c, d shows impulsive noise. The impulsive noise $v(i)$ is generated as $v(i)=b(i)*G(i)$, where $b(i)$ is Bernoulli process with a probability of success $P\{b(i)=1\}=p$. $G(i)$ in Fig. 2c is zero-mean Rayleigh noise, and $G(i)$ in Fig. 2d is zero-mean Gaussian noise. The kurtoses of the noises shown in Fig. 2a–d are 3.0, 4.1, 14.4, and 7.3, respectively.

The MSDs of the closed-form solutions for sub- and super-Gaussian noise were shown in Tables 1 and 2, respectively. From the above two tables, we can observe the following three conclusions: firstly, the existing algorithms (MSE, LMF, MCC, GMCC and MKRSL) do not perform better than the MSE method for sub and super-Gaussian noise simultaneously. The MCC, GMCC1, MKRSL1 and MKRSL2 performs better (worse) than the MSE method for super-Gaussian (sub-Gaussian) noise, whereas the LMF and GMCC2 perform better (worse) than the MSE for sub-Gaussian (super-Gaussian) noise. Simulations demonstrated that the proposed MSCE and GMSCE algorithm may perform better than the MSE algorithm both for sub and super-Gaussian noise. Secondly, the MCC performs as well as the MKRSL, whose parameters, λ and σ , did not influence the MSDs of the closed-form solution. Thirdly, the parameters, λ and α , have great influence on the GMCC. When $\alpha = 2$ and $\lambda = 0.031$, GMCC1 performs better than the MSE for super-Gaussian noise. When $\alpha = 4$ and $\lambda = 0.005$, GMCC2 performs better than the MSE for sub-Gaussian noise.

6.2 Adaptive solution for sub-Gaussian noise

In the simulation, the input filter order was $M = 5$, the sample size had length $L = 10,000$ and SNR was set to 6 dB. The proposed algorithms (22) and (30) are denoted by MSCE and GMSCE, respectively.

Table 1 The MSDs (dB) of the closed-form solutions of the MSCE, GMSCE, MSE, LMF, and MCC with different sub-Gaussian noises at SNR = 6 dB ($L = 3000$)

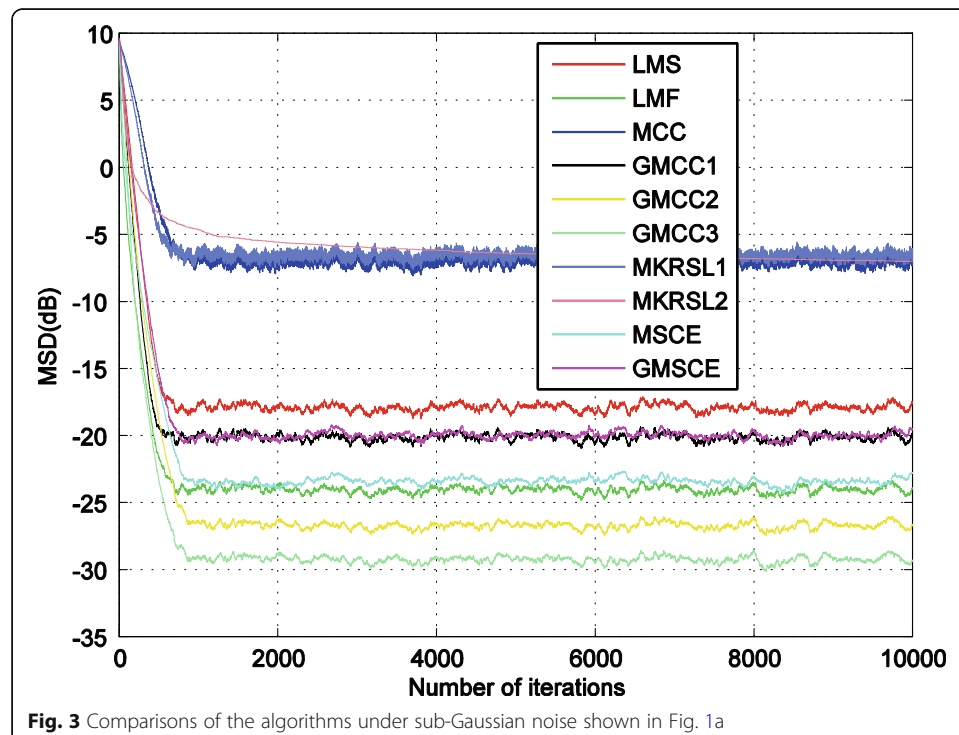
Algorithms	Sub-G noise (1)	Sub-G noise (2)	Sub-G noise (3)	Sub-G noise (4)	Averaged MSDs
MSCE	-31.0	-28.4	-31.4	-26.9	-29.4
GMSCE	-31.0	-28.4	-31.1	-27.1	-29.4
MSE	-28.4	-28.6	-28.6	-28.2	-28.5
LMF	-31.3	-26.7	-31.4	-30.2	-29.9
MCC ($\sigma = 1$)	-24.7	-25.7	-24.9	-25.7	-25.3
GMCC1 ($\alpha = 2$, $\lambda = 0.031$)	-28.0	-28.4	-28.2	-28.0	-28.2
GMCC2 ($\alpha = 4$, $\lambda = 0.005$)	-32.0	-27.1	-31.7	-30.5	-30.3
GMCC3 ($\alpha = 6$, $\lambda = 0.001$)	-20.7	-19.2	-20.7	-21.1	-20.4
MKRSL1 ($\lambda = 0.1$, $\sigma = 1$)	-24.7	-25.8	-24.9	-25.7	-25.3
MKRSL2 ($\lambda = 32$, $\sigma = 1.7$)	-24.7	-25.8	-24.9	-25.7	-25.3

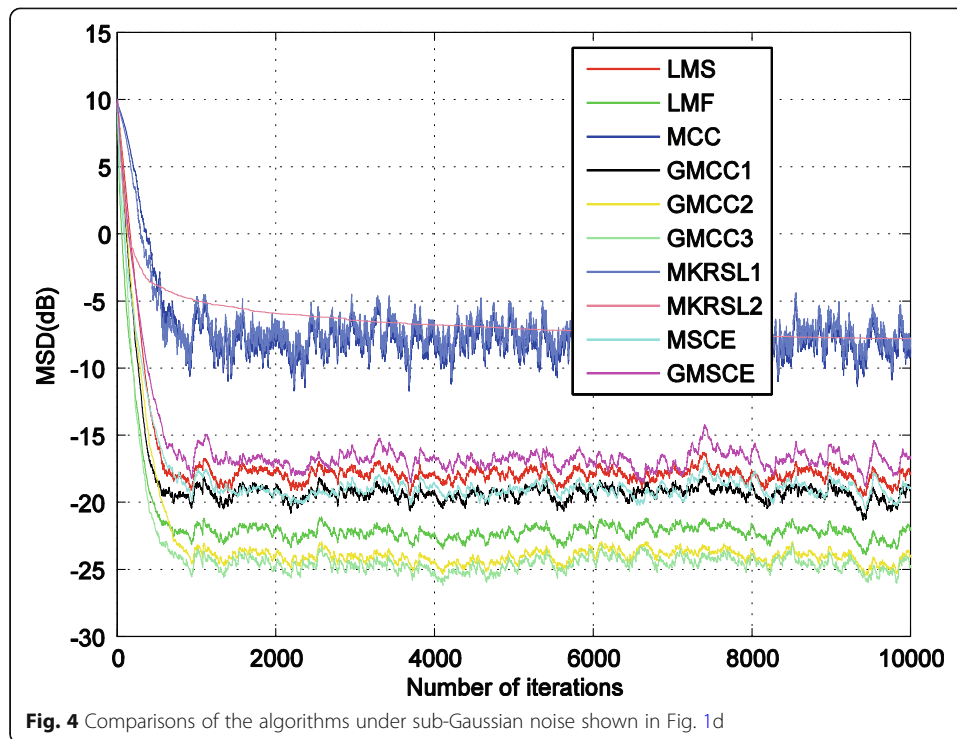
Table 2 The MSDs of the closed-form solutions of the MSCE, GMSCE, MSE, LMF, and MCC with different super-Gaussian noises at SNR = 6 dB ($L = 3000$)

Algorithms	Super-G noise (1)	Super-G noise (2)	Super-G noise (3)	Super-G noise (4)	Averaged MSDs
MSCE	− 78.4	− 73.2	− 22.2	− 22.6	− 49.1
GMSCE	− 74.4	− 71.7	− 23.6	− 23.8	− 48.4
MSE	− 28.0	− 28.2	− 28.2	− 27.5	− 28.0
LMF	− 14.9	− 12.4	− 1.62	− 7.13	− 9.01
MCC ($\sigma = 1$)	− 36.5	− 36.5	− 35.9	− 37.1	− 36.5
GMCC1 ($\alpha = 2, \lambda = 0.031$)	− 28.9	− 29.1	− 31.1	− 29.6	− 29.7
GMCC2 ($\alpha = 4, \lambda = 0.005$)	− 14.4	− 12.4	− 9.12	− 12.0	− 12.0
GMCC3 ($\alpha = 6, \lambda = 0.001$)	− 14.4	− 9.12	− 5.68	− 9.63	− 9.71
MKRSL1 ($\lambda = 0.1, \sigma = 1$)	− 36.5	− 36.5	− 35.9	− 37.1	− 36.5
MKRSL2 ($\lambda = 32, \sigma = 1.7$)	36.5	− 36.5	− 35.9	− 37.1	− 36.5

For sub-Gaussian noise shown in Fig. 1a, d, we compared the performance of the LMS, LMF, MCC, GMCC1-3, MKRSL1-2, MSCE, and GMSCE. The step-sizes were chosen such that all the algorithms had almost the same initial convergence speed, and other parameters (if any) for each algorithm were experimentally selected to achieve desirable performance.

The comparisons were shown in Figs. 3 and 4. From the two figures we can observe:





First, the GMCC1-3, LMF, and MSCE performed better than LMS for sub-Gaussian noises. GMCC1 and GMCC2 perform best among the algorithms.

Second, the MKRSL1-2 and MCC performed worse than the LMS. The performance curves of MKRSL1 and MCC were almost overlapped.

Third, the performance of the adaptive solution was not always consistent with that of the closed-form solution. Table 1 showed that the closed-form solution of GMCC3 was worse than MSE, but the adaptive solution of GMCC3 was better than MSE. It may be hard for each algorithm to achieve a good tradeoff between the same initial convergence speed and the desirable performance (steady-state error).

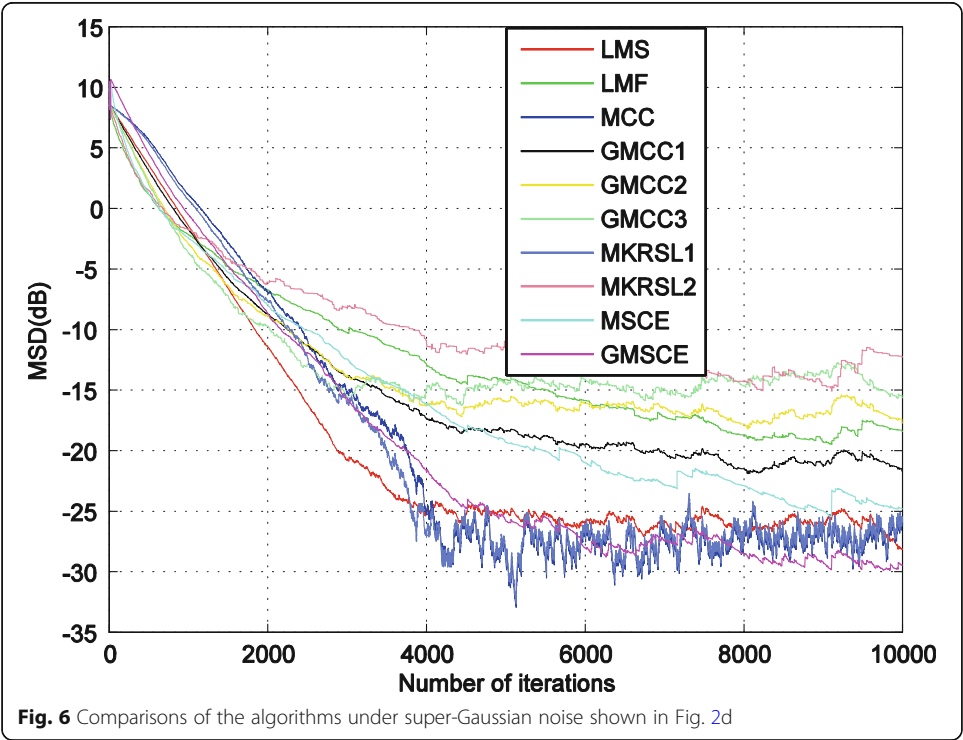
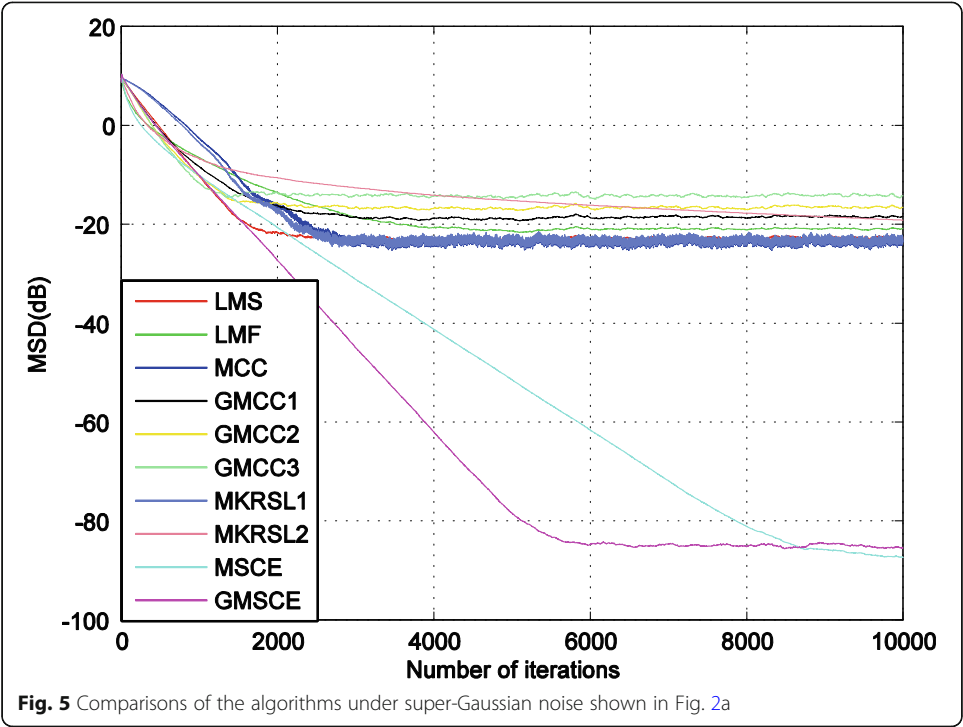
6.3 Adaptive solution for super-Gaussian noise

In the simulations, the input filter order was $M = 5$, the sample size had length $L = 10,000$ and SNR was set to 6 dB. The step-sizes were chosen such that all the algorithms had almost the same initial convergence speed.

For the super-Gaussian noise shown in Fig. 2a, d, we compared the performance of the LMS, LMF, MCC, GMCC1-3, MKRSL1-2, MSCE, and GMSCE. The comparisons were shown in Figs. 5 and 6. From the two figures we can observe:

First, the proposed MSCE and GMSCE performed much better than other algorithms for the periodic super-Gaussian noise shown in Fig. 2a. The MSCE performed a little better than the LMS for impulsive noise shown in Fig. 2d.

Second, the MKRSL1 and MCC had almost the same performance, the two algorithms performed a little better than the LMS.



Third, the LMF, GMCC1-3, and MKRSL2 performed worse than the LMS, though the closed-form solutions of the GMCC2 and MKRSL2 performed better than the LMS.

Combining the above simulations in this section, we can find that each algorithm may have its good points, and no algorithms can perform best for all kinds of noise. Dividing the additive noise into three types will be helpful to select the suitable algorithm for real applications.

7 Conclusions

This paper proposes a new cost function called the MSCE for adaptive filters, and provided the mean value and mean square performance analysis in detail. We have also presented a two-stage method to estimate the closed-form solutions for the MSCE method, and generalize the two-stage method to estimate the closed-form solution of the information theoretic learning methods, such as LMF, MCC, GMCC, and MKRSL.

The additive noise in adaptive filtering is divided into three types: Gaussian, sub-Gaussian, and super-Gaussian. The existing algorithms do not perform better than the mean square error method for sub and super-Gaussian noise simultaneously. The MCC, GMCC1, MKRSL1 and MKRSL2 performs better (worse) than the MSE method for super-Gaussian (sub-Gaussian) noise, whereas the LMF and GMCC2 perform better (worse) than the MSE for sub-Gaussian (super-Gaussian) noise. Simulations demonstrated that the proposed MSCE and GMSCE algorithm may perform better than the MSE algorithm both for sub and super-Gaussian noise.

In the future work, the MSCE algorithm may be extended to Kalman filtering, complex-valued filtering, distributed estimation, and non-linear filtering.

Abbreviations

MSE: Mean square error; ITL: Information theoretic learning; MCC: Maximum correntropy criterion; ILSE: Improved least sum of exponentials; LMK: Least mean kurtosis; LMF: Least mean fourth; GMCC: Generalized maximum correntropy criterion; MKRSL: Minimum kernel risk-sensitive loss; SNR: Signal-to-noise ratio; CE: Cross error; BSS: Blind source separation; MSCE: Mean square CE Cross error

Acknowledgements

Thanks to the anonymous reviewers and editors for their hard work.

Authors' contributions

Rui Xue and Gang Wang proposed the original idea of the full text. Rui Xue designed the experiment. Yunxiang Zhang, Yuyang Zhao and Gang Wang performed the experiment and analyzed the results. Yunxiang Zhang and Yuyang Zhao drafted the manuscript. Rui Xue and Gang Wang wrote the manuscript. All authors read and approved this submission.

Funding

This work was supported by the National Key Research and Development Program of China (Project No. 2017YFB0503400), National Natural Science Foundation of China under Grants 61371182 and 41301459.

Availability of data and materials

The datasets used during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

All authors agree to publish the submitted paper in this journal.

Competing interests

The authors declare that they have no competing interests.

Author details

¹School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, P.R. China. ²School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu 611731, P.R. China. ³School of Electronics and Information Engineering, Beihang University, Beijing, P. R. China.

Received: 29 January 2021 Accepted: 6 May 2021

Published online: 20 May 2021

References

1. X. Li, T. Adali, Complex-valued linear and widely linear filtering using MSE and Gaussian entropy. *IEEE Trans. Signal Process.* **60**(11), 5672–5684 (2012)
2. T. Adali, P.J. Schreier, Optimization and estimation of complex valued signals: theory and applications in filtering and blind source separation. *IEEE Signal Process. Mag.* **31**(5), 112–128 (2014)
3. S.Y. Huang, C.G. Li, Y. Liu, Complex-valued filtering based on the minimization of complex-error entropy. *IEEE Trans. Signal Process.* **24**(5), 695–708 (2013)
4. A.H. Sayed, Adaptive networks. *Proc. IEEE* **102**, 460–497 (2014)
5. J. Chen, A.H. Sayed, Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process.* **60**(8), 4289–4305 (2011)
6. W. Liu, P.P. Pokharel, J.C. Principe, Correntropy: properties and applications in non-Gaussian signal processing. *IEEE Trans. Signal Process.* **55**(11), 5286–5298 (2007)
7. A. Singh, J.C. Principe, in *Proceedings of the International Joint Conference on Neural Networks 2009*, Using correntropy as a cost function in linear adaptive filters (IEEE, Atlanta, 2009), pp. 2950–2955. <https://doi.org/10.1109/IJCNN.2009.5178823>
8. R. He, W.S. Zheng, B.G. Hu, Maximum correntropy criterion for robust face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(8), 1561–1576 (2011)
9. A.I. Fontes, A.M. de Martins, L.F. Silveira, J. Principe, Performance evaluation of the correntropy coefficient in automatic modulation classification. *Expert Syst. Appl.* **42**(1), 1–8 (2015)
10. W. Ma, B. Chen, J. Duan, H. Zhao, Diffusion maximum correntropy criterion algorithms for robust distributed estimation. *Digit. Signal Process.* **58**, 10–19 (2016)
11. J.P.F. Guimaraes, A.I.R. Fontes, J.B.A. Rego, A.M. de Martins, J.C. Principe, Complex correntropy: probabilistic interpretation and application to complex-valued data. *IEEE Signal Process. Lett.* **24**(1), 42–25 (2017)
12. S. Wang, Y. Zheng, S. Duan, L. Wang, C.K. Tse, A class of improved least sum of exponentials algorithms. *Signal Process.* **128**, 340–349 (2016)
13. N.J. Bershad, J.C.M. Bermudez, Stochastic analysis of the least mean kurtosis algorithm for Gaussian inputs. *Digit. Signal Process.* **54**, 35–45 (2016)
14. E. Walach, B. Widrow, The least mean fourth (LMF) adaptive algorithm and its family. *IEEE Trans. Inf. Theory* **30**, 275–283 (1984)
15. P.I. Hubscher, J.C.M. Bermudez, An improved statistical analysis of the least mean fourth (LMF) adaptive algorithm. *IEEE Trans. Signal Process.* **51**(3), 664–671 (2003)
16. E. Eweda, Global stabilization of the least mean fourth algorithm. *IEEE Trans. Signal Process.* **60**(3), 1473–1477 (2012)
17. E. Eweda, Mean-square stability analysis of a normalized least mean fourth algorithm for a Markov plant. *IEEE Trans. Signal Process.* **62**(24), 6545–6553 (2014)
18. E. Eweda, Dependence of the stability of the least mean fourth algorithm on target weights non-stationarity. *IEEE Trans. Signal Process.* **62**(7), 1634–1643 (2014)
19. W. Wang, H. Zhao, Performance analysis of diffusion least mean fourth algorithm over network. *Signal Process.* **141**, 32–47 (2017)
20. B. Chen, L. Xing, H. Zhao, N. Zheng, J.C. Principe, Generalized correntropy for robust adaptive filtering. *IEEE Trans. Signal Process.* **64**(13), 3376–3387 (2016)
21. B. Chen, R. Wang, Risk-sensitive loss in kernel space for robust adaptive filtering. *Proc. 2015 IEEE Int. Conf. Digit. Signal Process.*, 921–925 (2015)
22. B. Chen, L. Xing, B. Xu, H. Zhao, N. Zheng, J.C. Principe, Kernel risk-sensitive loss: definition, properties and application to robust adaptive filtering. *IEEE Trans. Signal Process.* **65**(11), 2888–2901 (2017)
23. A. Cichocki, S. Amari, *Adaptive blind signal and image processing, learning algorithms and applications* (Wiley, New York, 2002)
24. A. Hyvärinen, J. Karhunen, E. Oja, *Independent component analysis* (Wiley, New York, 2001)
25. G. Wang, Y. Zhang, B. He, K.T. Chong, A framework of target detection in hyperspectral imagery based on blind source extraction. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **9**(2), 835–844 (2016)
26. G. Wang, C. Li, L. Dong, Noise estimation using mean square cross prediction error for speech enhancement. *IEEE Trans. Circuits Syst. I Reg. Pap.* **57**, 1489–1499 (2010)
27. G. Wang, N. Rao, S. Shepherd, C. Beggs, Extraction of desired signal based on AR model with its application to atrial activity estimation in atrial fibrillation. *EURASIP J. Adv. Signal Process.* **9**, 728409 (2008)
28. A. Hyvärinen, Blind source separation by nonstationarity of variance: a cumulant-based approach. *IEEE Trans. Neural Netw.* **12**(6), 1471–1474 (2001)
29. A. Hyvärinen, Sparse code shrinkage: denoising of nonGaussian data by maximum likelihood estimation. *Neural Comput.* **11**(7), 1739–1768 (1999)
30. A. Hyvärinen, E. Oja, Fast and robust fixed-point algorithm for independent component analysis. *IEEE Trans. Neural Netw.* **10**(3), 626–634 (1999)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.