Signal Processing 140 (2017) 116–126

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Constrained maximum correntropy adaptive filtering

Siyuan Peng[a], Badong Chen[b,*], Lei Sun[c], Wee Ser[a], Zhiping Lin[a]

[a] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore
[b] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China
[c] School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China

ARTICLE INFO

Article history:
Received 8 February 2017
Revised 5 May 2017
Accepted 6 May 2017
Available online 16 May 2017

Keywords:
Constrained adaptive filtering
Maximum correntropy criterion
Non-Gaussian signal processing
Convergence analysis

ABSTRACT

Constrained adaptive filtering algorithms have been extensively studied in many applications. Most existing constrained adaptive filtering algorithms are developed under the mean square error (MSE) criterion, which is an ideal optimality criterion under Gaussian noises. This assumption however fails to model the behavior of non-Gaussian noises found in practice. Motivated by the robustness and simplicity of maximum correntropy criterion (MCC) for non-Gaussian impulsive noises, this paper proposes a new adaptive filtering algorithm called constrained maximum correntropy criterion (CMCC). Specifically, CMCC incorporates a linear constraint into a MCC filter to solve a constrained optimization problem explicitly. The proposed adaptive filtering algorithm is easy to implement, has low computational complexity, and can significantly outperform those MSE based constrained adaptive algorithms in heavy-tailed impulsive noises. Additionally, the mean square convergence behaviors are studied under energy conservation relation, and a sufficient condition to ensure the mean square convergence and the steady-state mean square deviation (MSD) of the CMCC algorithm are obtained. Simulation results confirm the theoretical predictions under both Gaussian and non-Gaussian noises, and demonstrate the excellent performance of the novel algorithm by comparing it with other conventional methods.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Constrained adaptive filtering algorithms have been successfully applied in domains of signal processing and communications, such as system identification, blind interference suppression, array signal processing, and spectral analysis [1,2]. The main advantage of constrained adaptive filters is that they have an error-correcting feature that can prevent the accumulation of errors (e.g., the quantization errors in a digital implementation). As a well-known linearly-constrained adaptive filtering algorithm, the *constrained least mean square* (CLMS) [3] is a simple stochastic-gradient based adaptive algorithm, originally conceived as an adaptive solution to a linearly-constrained minimum-variance (LCMV) filtering problem in antenna array processing [4]. Although the CLMS is simple and computationally efficient, it obviously suffer from low convergence speed especially when the input signal is correlated. In order to improve the convergence rate, the *constrained recursive least squares* (CRLS) algorithm was derived in [5], at the cost of higher computational complexity. Some improvements of the CRLS can be found in [6,7]. Several *constrained affine projection* (CAP) algorithms were also developed [8,9].

Most of the existing constrained adaptive filtering algorithms have been developed based on the common mean square error (MSE) criterion due to its attractive features, such as mathematical tractability, computational simplicity and optimality under Gaussian assumption [10]. However, Gaussian assumption does not always hold in real-world environments, even though it is justified for many natural noises. When the signals are disturbed by non-Gaussian noises, the MSE based algorithms may perform poorly or encounter the instability problem [11,12]. From a statistical viewpoint, the MSE is insufficient to capture all possible information in non-Gaussian signals. In practical situations, non-Gaussian noises are frequently encountered. For example, some sources of non-Gaussian impulsive noises are ill synchronization in digital recording, motor ignition noise in internal combustion engines, and lighting spikes in natural phenomena [13,14].

To deal with the non-Gaussian noise problem (which usually causes large outliers), maximum correntropy criterion (MCC) has been successfully applied to replace the traditional MSE criterion due to its simplicity and robustness [11,12,15–19]. As a nonlinear and local similarity measure directly related to the probability of how similar two random variables are in the bisector neighborhood of the joint space controlled by the kernel bandwidth, cor-

* Corresponding author.
E-mail address: chenbd@mail.xjtu.edu.cn (B. Chen).

http://dx.doi.org/10.1016/j.sigpro.2017.05.009
0165-1684/© 2017 Elsevier B.V. All rights reserved.

rentropy is insensitive to large outliers, and is frequently used as a powerful method to handle non-Gaussian impulsive noises in various applications of engineering. For instance, Singh et al. [20] and Zhao et al. [16] utilized the correntropy as a cost function to develop robust adaptive filtering algorithm for signal processing, and Chen et al. extended the original correntropy by using the generalized Gaussian density (GGD) function as the kernel, and proposed a generalized correntropy for robust adaptive filtering [21]. He et al. presented a MCC-based rotationally invariant principal component analysis (PCA) algorithm for image processing [22], and also incorporated the correntropy induced metric (CIM) into MCC to develop an effective sparse representation algorithm for robust face recognition [23]. Bessa et al. adopted MCC to train neural networks for wind prediction in power system [24]. Hasanbelliu et al. utilized information theoretic measures (entropy and correntropy) to develop two algorithms that can deal with both rigid and non-rigid point set registration with different computational complexities and accuracies [25]. However, constrained adaptive filtering based on MCC has not been studied yet in the literature.

In this work, a constrained maximum correntropy criterion (CMCC) adaptive filtering algorithm is proposed for signal processing especially in presence of heavy-tailed impulsive noises. The main contributions in this paper are summarized as follows:

- First, we develop the CMCC adaptive filtering algorithm by incorporating a linear constraint into the MCC to solve a constrained optimization problem explicitly. The computational complexity analysis is also presented.
- Second, based on the energy conservation relation [26–28], we analyze the mean square convergence behaviors of the proposed algorithm, and present particularly a sufficient condition to guarantee the mean square convergence and the steady-state mean square deviation (MSD) in the cases of Gaussian and non-Gaussian noises.
- Finally, we confirm the validity of theoretical expectations experimentally, and illustrate the desirable performance (e.g., lower MSD) of CMCC by comparing it with other methods in linear-phase system identification and beamforming application.

The rest of the paper is organized as follows. In Section 2, after briefly reviewing the MCC, we develop the CMCC algorithm and analyze the computational complexity. In Section 3, we study the mean square convergence of the proposed algorithm. Simulation results are then presented in Section 4. Finally, Section 5 gives the conclusion.

Notation: In this paper, capital boldface letters, small boldface letters, and normal font are respectively used to denote matrices, vectors, and scalars, e.g., $\mathbf{C}$, $\mathbf{w}$, and e. All vectors are column vectors, and the time instant for vectors and scalars is placed as a subscript, for example, $\mathbf{w}_n$, and e_n. In addition, the notation $\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$ stands for the squared Euclidean norm of a vector $\mathbf{w}$, and accordingly, the weighted squared Euclidean norm can be written as $\|\mathbf{w}\|_\Sigma^2 = \mathbf{w}^T\Sigma\mathbf{w}$. Other notations will be given in the rest of the paper if necessary.

2. CMCC Algorithm

2.1. Maximum correntropy criterion

As a similarity measure between two random variables X and Y, correntropy is defined by [12,18,20,21]

$$V(X,Y) = E[\kappa(X,Y)] = \int \kappa(x,y)dF_{XY}(x,y) \tag{1}$$

where $E[\cdot]$ denotes the *expectation operator*, $\kappa(\cdot,\cdot)$ is a *shift-invariant Mercer kernel*, and $F_{XY}(x, y)$ stands for the joint distribution function of (X, Y). It takes the advantage of a kernel trick that nonlinearly maps the input space to a higher dimensional feature space. In the present work, without mentioning otherwise, the kernel function of correntropy $\kappa(\cdot,\cdot)$ is the Gaussian kernel, given by

$$\kappa_\sigma(x-y) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-y)^2}{2\sigma^2}\right) \tag{2}$$

where $\sigma > 0$ is the kernel bandwidth parameter. In most practical situations, the join distribution $F_{XY}(x, y)$ is usually unknown, and only a finite number of data samples $\{(x_n, y_n)\}_{n=1}^N$ are available. In these cases, the correntropy can be estimated by

$$\hat{V}_{N,\sigma} = \frac{1}{N}\sum_{n=1}^{N}\kappa_\sigma(x_n - y_n) \tag{3}$$

where $\hat{(\cdot)}$ is the estimator. Under the *maximum correntropy criterion* (MCC), an adaptive filter will be trained by maximizing the correntropy between the desired response and filter output, formulated by

$$\max_{\mathbf{w}} J_{MCC} = \frac{1}{N}\sum_{n=1}^{N}\kappa_\sigma(e_n) \tag{4}$$

where e_n is the error between the desired response and filter output, and $\mathbf{w}$ stands for the filter weight vector. Fig. 1 shows the MCC cost function $\kappa_\sigma(x-y)$ in the joint space of x and y. As one can see clearly, the MCC is a local similarity measure, whose value is heavily decided by the kernel function along the line $x = y$. Furthermore, from a view of geometric meaning, we can divide the space in three regions, namely Euclidean region, transition region and rectification region. The MCC behaves like 2-norm distance in the Euclidean region, similarly like a 1-norm distance in the transition region and eventually approaches a zero-norm in the rectification region, which also interprets the robustness of correntropy for outliers [12,18].

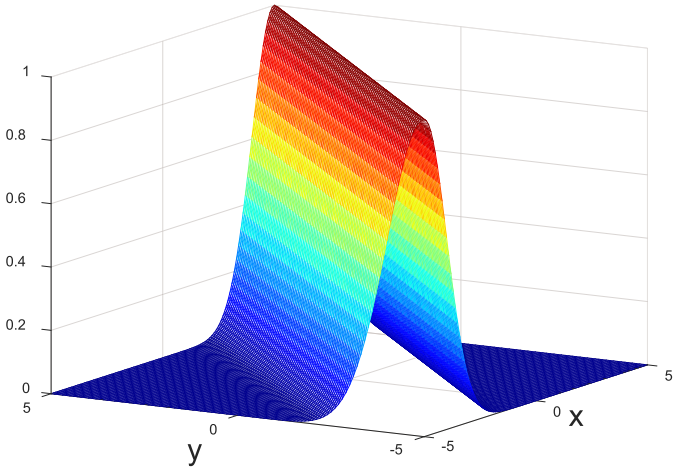

Fig. 1. MCC cost function in the joint space ($\sigma = 1.0$).

2.2. CMCC algorithm

Consider a linear unknown system, with an M-dimensional weight vector $\mathbf{w}^* = [w_1^*, w_2^*, \cdots, w_M^*]^T$ that needs to be estimated. The measured output d_n of the unknown system at instant n is assumed to be

$$d_n = y_n^* + \upsilon_n = \mathbf{w}^{*T}\mathbf{x}_n + \upsilon_n \tag{5}$$

where $y_n^* = \mathbf{w}^{*T}\mathbf{x}_n$ denotes the actual output of the unknown system, with $[\cdot]^T$ being the transpose operator, $\mathbf{x}_n = [x_{1,n}, x_{2,n}, \cdots, x_{M,n}]^T$ is the input vector, and υ_n stands

Table 1
Computational complexity of CMCC, CLMS, CAP and CRLS.

Algorithm	Computational Complexity
CMCC	$2M^2+5M+1+\Gamma_g$
CLMS	$2M^2+5M+1$
CAP	$2M^2+(2L+3)M+1$
CRLS	$7M^2+(6K^2+9K+5)M+3K$

for an interference or measurement noise. Suppose that the estimator is another M-dimensional linear filter, with an adaptive weight vector $\mathbf{w}_n$. Then the instantaneous prediction error at instant n is

$$e_n = d_n - y_n = d_n - \mathbf{w}_{n-1}^T\mathbf{x}_n \tag{6}$$

where $y_n = \mathbf{w}_{n-1}^T\mathbf{x}_n$ denotes the output of the adaptive filter. For a constrained adaptive filter, a linear constraint will be imposed upon the filter weight vector as

$$\mathbf{C}^T\mathbf{w} = \mathbf{f} \tag{7}$$

where $\mathbf{C}$ is an $M \times K$ constraint matrix, and $\mathbf{f}$ is a vector containing K constraint values. The CLMS algorithm is derived by solving the following optimization problem [3,27,29,30]:

$$\min_{\mathbf{w}} E\left[\left(d_n - \mathbf{w}_{n-1}^T\mathbf{x}_n\right)^2\right] \quad \textit{subject to} \quad \mathbf{C}^T\mathbf{w} = \mathbf{f} \tag{8}$$

leading to the following weight update equation:

$$\mathbf{w}_n = \mathbf{P}\left[\mathbf{w}_{n-1} + \eta(d_n - \mathbf{w}_{n-1}^T\mathbf{x}_n)\mathbf{x}_n\right] + \mathbf{q} \tag{9}$$

where η is the step-size parameter, $\mathbf{P} = \mathbf{I}_M - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T$ with $\mathbf{I}_M$ being an $M \times M$ identity matrix, and $\mathbf{q} = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$.

In this work, we use the MCC instead of MSE to develop a constrained adaptive filtering algorithm. Similar to (8), we propose the following CMCC optimization problem

$$\max_{\mathbf{w}} E\left[\kappa_\sigma\left(d_n - \mathbf{w}_{n-1}^T\mathbf{x}_n\right)\right] \quad \textit{subject to} \quad \mathbf{C}^T\mathbf{w} = \mathbf{f} \tag{10}$$

and accordingly, by defining the Lagrange function, the CMCC cost J_{CMCC} is

$$J_{CMCC} = E\left[\kappa_\sigma\left(d_n - \mathbf{w}_{n-1}^T\mathbf{x}_n\right)\right] + \boldsymbol{\xi}_n^T\left(\mathbf{C}^T\mathbf{w}_{n-1} - \mathbf{f}\right) \tag{11}$$

where $\boldsymbol{\xi}_n$ is a $K \times 1$ Lagrange multiplier vector. A stochastic-gradient based algorithm can thus be derived as (see Appendix A for a detailed derivation)

$$\mathbf{w}_n = \mathbf{P}\left[\mathbf{w}_{n-1} + \eta g(e_n)(d_n - \mathbf{w}_{n-1}^T\mathbf{x}_n)\mathbf{x}_n\right] + \mathbf{q} \tag{12}$$

where $g(e_n)$ is a nonlinear function of e_n, given by

$$g(e_n) = \exp\left(-\frac{e_n^2}{2\sigma^2}\right) \tag{13}$$

The above algorithm is referred to as the CMCC algorithm. It should be noted that, when $\sigma \to \infty$, we have $g(e_n) \to 1$, which implies that the CMCC algorithm is approximately equal to the CLMS algorithm.

2.3. Computational complexity

The computational complexity of the proposed CMCC algorithm and other constrained adaptive algorithms (e.g., CLMS, CAP and CRLS), in terms of the total number of required additions and multiplications at each iteration, are shown in Table 1, where Γ_g is a constant associated with the complexity of the nonlinear function $g(e_n)$. Obviously, the computational complexity of these algorithms is $O(M^2)$. Since Γ_g is usually not expensive, it can be seen that the proposed algorithm has lower computational cost than CRLS due to calculating the covariance matrix of the input vector per iteration for CRLS, also has lower computational cost than CAP (especially when the sliding window length L is large). Generally speaking, the computational complexity of CMCC is almost the same as that of the CLMS.

3. Convergence analysis

3.1. Assumptations

In this section, we analyze the mean square convergence behaviors of the proposed CMCC algorithm. First, we give the following assumptions:

1) The input sequence $\{\mathbf{x}_n\}$ is a independent, identically, distributed (i.i.d) multivariate Gaussian, with zero-mean and the positive-definite covariance matrix of the input sequence $\mathbf{R} = E[\mathbf{x}_n\mathbf{x}_n^T]$.
2) The noise $\{\upsilon_n\}$ is zero-mean, i.i.d, and independent of any other signals in the system.
3) The filter is long enough such that the a priori error e_n^a, to be defined later, is zero-mean Gaussian.
4) The error nonlinearity $g(e_n)$ is asymptotically uncorrelated with $\|\mathbf{x}_n\|^2$ at steady-state.

The independence assumptions 1) and 2) are very popular and have been frequently used in the literature for performance analysis of most adaptive algorithm [10,27,31,32]. When the filter is long enough, assumption 3) is reasonable in practical by central limit theorem, and also remains valid in the whole stage of adaptation. Assumption 4) will become realistic and valid especially when the weight vector get longer (see [26–28,33] for more detailed explanation about assumptions 3) and 4)).

3.2. Optimal solution

Setting $\frac{\partial J_{CMCC}}{\partial \mathbf{w}}|_{\mathbf{w}=\mathbf{w}_{n-1}} = \mathbf{0}_{M\times 1}$ (Here $\mathbf{0}_{M\times 1}$ denotes the $M \times 1$ zero vector), one can derive the optimal weight vector $\mathbf{w}_{opt}$ under the CMCC optimization problem as follows:

$$\begin{aligned}
&E\left[g(e_n)(d_n - \mathbf{w}_{opt}^T\mathbf{x}_n)\mathbf{x}_n\right] + \mathbf{C}\boldsymbol{\xi}_n = \mathbf{0}_{M\times 1}\\
&\Rightarrow E\left[g(e_n)\mathbf{x}_n\mathbf{x}_n^T\right]\mathbf{w}_{opt} = E[g(e_n)d_n\mathbf{w}_n] + \mathbf{C}\boldsymbol{\xi}_n\\
&\Rightarrow \mathbf{R}_g\mathbf{w}_{opt} = \mathbf{p}_g + \mathbf{C}\boldsymbol{\xi}_n\\
&\Rightarrow \mathbf{w}_{opt} = \mathbf{R}_g^{-1}\mathbf{p}_g + \mathbf{R}_g^{-1}\mathbf{C}\boldsymbol{\xi}_n
\end{aligned} \tag{14}$$

where $\mathbf{R}_g = E\left[g(e_n)\mathbf{x}_n\mathbf{x}_n^T\right]$ denotes a weighted autocorrelation matrix of the input vector, and $\mathbf{p}_g = E[g(e_n)d_n\mathbf{x}_n]$ is a weighted cross-correlation vector between the measured output and the input vector. Since

$$\begin{aligned}
&\mathbf{C}^T\mathbf{w}_{opt} = \mathbf{f}\\
&\Rightarrow \mathbf{C}^T\left[\mathbf{R}_g^{-1}\mathbf{p}_g + \mathbf{R}_g^{-1}\mathbf{C}\boldsymbol{\xi}_n\right] = \mathbf{f}\\
&\Rightarrow \boldsymbol{\xi}_n = \left[\mathbf{C}^T\mathbf{R}_g^{-1}\mathbf{C}\right]^{-1}\left(\mathbf{f} - \mathbf{C}^T\mathbf{R}_g^{-1}\mathbf{p}_g\right)
\end{aligned} \tag{15}$$

one can rewrite (14) as

$$\mathbf{w}_{opt} = \mathbf{R}_g^{-1}\mathbf{p}_g + \mathbf{R}_g^{-1}\mathbf{C}\left[\mathbf{C}^T\mathbf{R}_g^{-1}\mathbf{C}\right]^{-1}\left(\mathbf{f} - \mathbf{C}^T\mathbf{R}_g^{-1}\mathbf{p}_g\right) \tag{16}$$

Under the assumptions 1) and 2), we derive by using (5)

$$\begin{aligned}
&d_n = \mathbf{w}^{*T}\mathbf{x}_n + \upsilon_n\\
&\Rightarrow g(e_n)d_n\mathbf{x}_n^T = g(e_n)\mathbf{w}^{*T}\mathbf{x}_n\mathbf{x}_n^T + \upsilon_n g(e_n)\mathbf{x}_n^T\\
&\Rightarrow \mathbf{p}_g = \mathbf{R}_g\mathbf{w}^*\\
&\Rightarrow \mathbf{w}^* = \mathbf{R}_g^{-1}\mathbf{p}_g
\end{aligned} \tag{17}$$

Therefore, combining (16) and (17), we obtain

$$\mathbf{w}_{opt} = \mathbf{w}^* + \mathbf{R}_g^{-1}\mathbf{C}\left(\mathbf{C}^T\mathbf{R}_g^{-1}C\right)^{-1}\left(\boldsymbol{f} - \mathbf{C}^T\mathbf{w}^*\right) \tag{18}$$

Remark. Since the weighting in $\mathbf{R}_g$ and $\mathbf{p}_g$ is an exponential Gaussian function of the error, which directly relies on the weight vector $\mathbf{w}$ through the error (see (6)), the above optimal solution is not a closed-form solution, and actually, is a fixed-point equation [21]. Furthermore, the optimal weight vector in (18) is close to the optimal filter coefficient vector of the CLMS algorithm as $\sigma \to \infty$ (hence $g(e_n) \to 1$) [27].

3.3. Energy conservation relation

To derive the energy conservation relation of the error quantities, we first introduce the following two useful error measures:

$$\tilde{\mathbf{w}}_n = \mathbf{w}_n - \mathbf{w}_{opt} \tag{19}$$

$$e_n^a = (\mathbf{w}^* - \mathbf{w}_n)^T \mathbf{x}_n \tag{20}$$

where $\tilde{\mathbf{w}}_n$ denotes the weight error vector, and e_n^a stands for a priori error. Indeed, we also define

$$\boldsymbol{\varepsilon}_{\mathbf{w}} = \mathbf{w}^* - \mathbf{w}_{opt} \tag{21}$$

Substituting (5), (19) and (21) into (12) yields

$$\begin{aligned}\tilde{\mathbf{w}}_n =& \mathbf{P}\left[\mathbf{w}_{n-1} + \eta g(e_n)(d_n - \mathbf{w}_{n-1}^T\mathbf{x}_n)\mathbf{x}_n\right] + \mathbf{q} - \mathbf{w}_{opt} \\ =& \mathbf{P}\left[\mathbf{I}_M - \eta g(e_n)\mathbf{x}_n\mathbf{x}_n^T\right]\tilde{\mathbf{w}}_{n-1} + \eta g(e_n)\upsilon_n\mathbf{P}\mathbf{x}_n \\ &+ \eta g(e_n)\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\boldsymbol{\varepsilon}_{\mathbf{w}} + \mathbf{P}\mathbf{w}_{opt} - \mathbf{w}_{opt} + \mathbf{q}\end{aligned} \tag{22}$$

Due to $\mathbf{P}\mathbf{w}_{opt} - \mathbf{w}_{opt} + \mathbf{q} = \mathbf{0}_{M\times 1}$, we can rewrite (22) as

$$\tilde{\mathbf{w}}_n = \mathbf{P}\left[\mathbf{I}_M - \eta g(e_n)\mathbf{x}_n\mathbf{x}_n^T\right]\tilde{\mathbf{w}}_{n-1} + \eta g(e_n)\upsilon_n\mathbf{P}\mathbf{x}_n + \eta g(e_n)\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\boldsymbol{\varepsilon}_{\mathbf{w}} \tag{23}$$

Note that matrix $\mathbf{P}$ is idempotent, namely $\mathbf{P} = \mathbf{P}^2$ and $\mathbf{P} = \mathbf{P}^T$. Multiplying both sides of (23) by $\mathbf{P}$ and after some straightforward matrix manipulations, we can obtain

$$\mathbf{P}\tilde{\mathbf{w}}_n = \tilde{\mathbf{w}}_n \tag{24}$$

Combining (23) and (24), we have

$$\begin{aligned}\tilde{\mathbf{w}}_n =& \mathbf{P}\tilde{\mathbf{w}}_{n-1} - \eta g(e_n)\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\tilde{\mathbf{w}}_{n-1} + \eta g(e_n)\upsilon_n\mathbf{P}\mathbf{x}_n + \eta g(e_n)\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\boldsymbol{\varepsilon}_{\mathbf{w}} \\ =& \left(\mathbf{I}_M - \eta g(e_n)\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\mathbf{P}\right)\tilde{\mathbf{w}}_{n-1} + \eta g(e_n)\upsilon_n\mathbf{P}\mathbf{x}_n + \eta g(e_n)\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\boldsymbol{\varepsilon}_{\mathbf{w}}\end{aligned} \tag{25}$$

Under assumptions 1), 2) and 3), taking the expectations of the squared Euclidean norms of both sides of (25) leads to the following energy conservation relation:

$$\begin{aligned}E\left[\|\tilde{\mathbf{w}}_n\|^2\right] =& E\left[\|\tilde{\mathbf{w}}_{n-1}\|_{\mathbf{H}}^2\right] + \eta^2 E\left[g^2(e_n)\right]E\left[\upsilon_n^2\right]E\left[\mathbf{x}_n^T\mathbf{P}\mathbf{x}_n\right] \\ &+ \eta^2 E\left[g^2(e_n)\right]\boldsymbol{\varepsilon}_{\mathbf{w}}^T E\left[\mathbf{x}_n\mathbf{x}_n^T\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\right]\boldsymbol{\varepsilon}_{\mathbf{w}}\end{aligned} \tag{26}$$

where $E\left[\|\tilde{\mathbf{w}}_n\|^2\right]$ is called the *weight error power* (WEP) at iteration n, and

$$\mathbf{H} = \mathbf{I}_M - 2\eta E[g(e_n)]\mathbf{P}\mathbf{R}\mathbf{P} + \eta^2 E\left[g^2(e_n)\right]\mathbf{P}E[\mathbf{x}_n\mathbf{x}_n^T\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T]\mathbf{P}$$

3.4. Mean square stability

Since $\mathbf{P} = \mathbf{P}^2$, we derive

$$E\left[\mathbf{x}_n^T\mathbf{P}\mathbf{x}_n\right] = E\left[\mathbf{x}_n^T\mathbf{P}\mathbf{P}\mathbf{x}_n\right] = tr\{\mathbf{P}\mathbf{R}\mathbf{P}\} = tr\{\Upsilon\} \tag{27}$$

where $tr\{\cdot\}$ stands for the *trace operator*, and $\Upsilon = \mathbf{P}\mathbf{R}\mathbf{P}$. According to the Isserlis' theorem [34] for Gaussian vectors $\hbar_1$, $\hbar_2$, $\hbar_3$ and $\hbar_4$, we have

$$\begin{aligned}E\left[\hbar_1\hbar_2^T\hbar_3\hbar_4^T\right] =& E\left[\hbar_1\hbar_2^T\right]E\left[\hbar_3\hbar_4^T\right] + E\left[\hbar_1\hbar_3^T\right] \\ &\times E\left[\hbar_2\hbar_4^T\right] + E\left[\hbar_1\hbar_4^T\right]E\left[\hbar_2^T\hbar_3\right]\end{aligned} \tag{28}$$

With $\hbar_1 = \mathbf{x}_n$, $\hbar_2 = \mathbf{x}_n$, $\hbar_3 = \mathbf{P}\mathbf{x}_n$ and $\hbar_4 = \mathbf{x}_n$, we obtain

$$\begin{aligned}E\left[\mathbf{x}_n\mathbf{x}_n^T\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\right] &= \mathbf{R}\mathbf{P}\mathbf{R} + \mathbf{R}\mathbf{P}\mathbf{R} + E\left[\mathbf{x}_n^T\mathbf{P}\mathbf{x}_n\right]\mathbf{R} \\ &= 2\mathbf{R}\mathbf{P}\mathbf{R} + tr\{\Upsilon\}\mathbf{R}\end{aligned} \tag{29}$$

Since $\mathbf{P}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} = \mathbf{0}_{M\times 1}$, Substituting (27) and (29) into (26), we get

$$E\left[\|\tilde{\mathbf{w}}_n\|^2\right] = E\left[\|\tilde{\mathbf{w}}_{n-1}\|_{\mathbf{H}}^2\right] + \eta^2 E\left[g^2(e_n)\right]tr\{\Upsilon\}\left(\boldsymbol{\varepsilon}_{\mathbf{w}}^T\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} + E\left[\upsilon_n^2\right]\right) \tag{30}$$

and

$$\mathbf{H} = \mathbf{I}_M - 2\eta E[g(e_n)]\mathbf{P}\mathbf{R}\mathbf{P} + \eta^2 E\left[g^2(e_n)\right](tr\{\Upsilon\}\mathbf{P}\mathbf{R}\mathbf{P} + 2\mathbf{P}\mathbf{R}\mathbf{P}\mathbf{R}\mathbf{P})$$

Let $\lambda_i (i = 1, \ldots, M-K)$ be the eigenvalues of the matrix Y. A sufficient condition for the mean square stability can be obtained as [3,27,35]

$$\left|1 - 2\eta E[g(e_n)]\lambda_i + \eta^2 E\left[g^2(e_n)\right]tr\{\Upsilon\}\lambda_i + 2\eta^2 E\left[g^2(e_n)\right]\lambda_i^2\right| < 1 \qquad i = 1, \ldots, M-K \tag{31}$$

After some simple manipulations, we have

$$0 < \eta < \frac{2E[g(e_n)]}{[2\lambda_{max} + tr\{\Upsilon\}]E[g^2(e_n)]} \tag{32}$$

where $\lambda_{\max}$ denotes the largest eigenvalue of the matrix Y. Due to $E[g(e_n)] \geq E[g^2(e_n)] > 0$, one can obtain a stronger condition to guarantee the mean square stability:

$$0 < \eta \leq \frac{2}{2\lambda_{max} + tr\{\Upsilon\}} \tag{33}$$

Remark. Since we only derive (32) and (33) under the steady-state assumption, we cannot solve the problem of how to select the best step-size for a specific application. However, the condition provides a possible range for choosing a step-size for CMCC algorithm. Similar analysis were derived in several literatures [36].

3.5. Steady-state mean square deviation (MSD)

Before processing, we define the steady-state MSD (a performance measure) as follows:

$$S = \lim_{n\to\infty} E\left[\|\tilde{\mathbf{w}}_n\|^2\right] \tag{34}$$

Assume that $\mathbf{T}$ is an arbitrary symmetric nonnegative definite matrix. Under assumptions 1), 2) and 3), one can derive the following relation by taking the expectations of the squared -weighted Euclidean norms of both sides of (25):

$$\begin{aligned}E\left[\|\tilde{\mathbf{w}}_n\|_{\mathbf{T}}^2\right] =& E\left[\|\tilde{\mathbf{w}}_{n-1}\|_{\mathbf{U}}^2\right] + \eta^2 E\left[g^2(e_n)\right]E\left[\upsilon_n^2\right]E\left[\mathbf{x}_n^T\mathbf{P}\mathbf{T}\mathbf{P}\mathbf{x}_n\right] \\ &+ \eta^2 E\left[g^2(e_n)\right]\boldsymbol{\varepsilon}_{\mathbf{w}}^T E\left[\mathbf{x}_n\mathbf{x}_n^T\mathbf{P}\mathbf{T}\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\right]\boldsymbol{\varepsilon}_{\mathbf{w}}\end{aligned} \tag{35}$$

in which

$$\begin{aligned}\mathbf{U} =& E\left[\left(\mathbf{I}_M - \eta g(e_n)\mathbf{x}_n\mathbf{x}_n^T\right)\mathbf{P}\mathbf{T}\mathbf{P}\left(\mathbf{I}_M - \eta g(e_n)\mathbf{x}_n\mathbf{x}_n^T\right)\right] \\ =& \mathbf{P}\mathbf{T}\mathbf{P} - \eta E[g(e_n)]\mathbf{R}\mathbf{P}\mathbf{T}\mathbf{P} - \eta E[g(e_n)]\mathbf{P}\mathbf{T}\mathbf{P}\mathbf{R} \\ &+ \eta^2 E\left[g^2(e_n)\right]E\left[\mathbf{x}_n\mathbf{x}_n^T\mathbf{P}\mathbf{T}\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\right]\end{aligned} \tag{36}$$

In the same way as for (27) and (29), we derive

$$E\left[\mathbf{x}_n^T\mathbf{P}\mathbf{T}\mathbf{P}\mathbf{x}_n\right] = tr\{\mathbf{T}\Upsilon\} \tag{37}$$

$$E\left[\mathbf{x}_n\mathbf{x}_n^T\mathbf{P}\mathbf{T}\mathbf{P}\mathbf{x}_n\mathbf{x}_n^T\right] = tr\{\mathbf{T}\Upsilon\}\mathbf{R} + 2\mathbf{R}\mathbf{P}\mathbf{T}\mathbf{P}\mathbf{R} \tag{38}$$

Thus we can rewrite (36) as

$$\begin{aligned}\mathbf{U} =& (\mathbf{I}_M - \eta E[g(e_n)]\mathbf{R})\mathbf{P}\mathbf{T}\mathbf{P}(\mathbf{I}_M - \eta E[g(e_n)]\mathbf{R}) \\ &+ \eta^2 E\left[g^2(e_n)\right]tr\{\mathbf{T}\Upsilon\}\mathbf{R} + 2\eta^2 E\left[g^2(e_n)\right] \\ &\times \mathbf{R}\mathbf{P}\mathbf{T}\mathbf{P}\mathbf{R} - \eta^2 E^2[g(e_n)]\mathbf{R}\mathbf{P}\mathbf{T}\mathbf{P}\mathbf{R}\end{aligned} \tag{39}$$

From [37], some useful properties can be obtained, that is,

$$\mathbf{vec}\{\mathbf{BCD}\} = \left(\mathbf{D}^T \otimes \mathbf{B}\right)\mathbf{vec}\{\mathbf{C}\}\text{and}$$

$$tr\left\{\mathbf{B}^T\mathbf{C}\right\} = \mathbf{vec}^T\{\mathbf{C}\}\mathbf{vec}\{\mathbf{B}\}$$

where $\mathbf{vec}\{\cdot\}$ denotes the vectorization operator, $\otimes$ stands for the Kronecker product. With the vectorization and the above properties, we have

$$\mathbf{vec}\{\mathbf{U}\} = \mathbf{Ft} \tag{40}$$

where

$$\begin{aligned}\mathbf{F} =&(\mathbf{I}_M - \eta E[g(e_n)]\mathbf{R})\mathbf{P} \otimes (\mathbf{I}_M - \eta E[g(e_n)]\mathbf{R})\mathbf{P}\\ &+ 2\eta^2 E\left[g^2(e_n)\right](\mathbf{RP}\otimes\mathbf{RP}) + \eta^2 E\left[g^2(e_n)\right]\\ &\times \mathbf{vec}\{\mathbf{R}\}\mathbf{vec}\{\Upsilon\} - \eta^2 E^2[g(e_n)](\mathbf{RP}\otimes\mathbf{RP})\end{aligned}$$

and $\mathbf{t} = \mathbf{vec}\{\mathbf{T}\}$. Combining (37), (38) and (40), we can rewrite (35) as

$$\begin{aligned}E\left[\|\tilde{\mathbf{w}}_n\|^2_{\mathbf{t}}\right] =& E\left[\|\tilde{\mathbf{w}}_{n-1}\|^2_{\mathbf{Ft}}\right] + \eta^2 E\left[g^2(e_n)\right]\\ &\times \left(\boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} + E\left[\upsilon_n^2\right]\right)\mathbf{vec}^T\{\Upsilon\}\mathbf{t}\end{aligned} \tag{41}$$

Assume that the filter is stable and achieves the steady-state, i.e. $\lim_{n\to\infty} E\left[\|\tilde{\mathbf{w}}_n\|^2\right] = \lim_{n\to\infty} E\left[\|\tilde{\mathbf{w}}_{n-1}\|^2\right]$. By using (41), we have

$$\begin{aligned}\lim_{n\to\infty} E\left[\|\tilde{\mathbf{w}}_n\|^2_{(\mathbf{I}_{M^2}-\mathbf{F})\mathbf{t}}\right] =& \lim_{n\to\infty} \eta^2 E\left[g^2(e_n)\right]\\ &\times \left(\boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} + E\left[\upsilon_n^2\right]\right)\mathbf{vec}^T\{\Upsilon\}\mathbf{t}\end{aligned} \tag{42}$$

Therefore, by selecting an appropriate $\mathbf{t} = \left(\mathbf{I}_{M^2} - \mathbf{F}\right)^{-1}\mathbf{vec}\{\mathbf{I}_M\}$, we can obtain

$$\begin{aligned}S =&\eta^2\left(\boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} + E\left[\upsilon_n^2\right]\right)\mathbf{vec}^T\{\Upsilon\}\\ &\times \lim_{n\to\infty}(\mathbf{I}_{M^2} - \mathbf{F})^{-1}\mathbf{vec}\{\mathbf{I}_M\}E\left[g^2(e_n)\right]\end{aligned} \tag{43}$$

Based on assumption 3), we can rewrite (18) as following:

$$\mathbf{w}_{opt} = \mathbf{w}^* + \mathbf{R}^{-1}\mathbf{C}\left(\mathbf{C}^T\mathbf{R}^{-1}C\right)^{-1}\left(\boldsymbol{f} - \mathbf{C}^T\mathbf{w}^*\right) \tag{44}$$

and accordingly

$$\boldsymbol{\varepsilon}_w = \mathbf{R}^{-1}\mathbf{C}\left(\mathbf{C}^T\mathbf{R}^{-1}C\right)^{-1}\left(\mathbf{C}^T\mathbf{w}^* - \boldsymbol{f}\right) \tag{45}$$

In order to obtain the theoretical value of the steady-state MSD, we also need to evaluate the values of $\lim_{n\to\infty} E[g(e_n)]$ and $\lim_{n\to\infty} E\left[g^2(e_n)\right]$. We consider two cases below:

1. If υ_n is zero-mean Gaussian distributed with variance σ_υ^2, then

$$\lim_{n\to\infty} E[g(e_n)] \approx \frac{\sigma}{\sqrt{\sigma^2 + \boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} + \sigma_\upsilon^2}} \tag{46}$$

$$\lim_{n\to\infty} E\left[g^2(e_n)\right] \approx \frac{\sigma}{\sqrt{\sigma^2 + 2\boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} + 2\sigma_\upsilon^2}} \tag{47}$$

Thus

$$\begin{aligned}S \approx&\eta^2\left(\boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} + \sigma_\upsilon^2\right)\mathbf{vec}^T\{\Upsilon\}(\mathbf{I}_{M^2} - \mathbf{F})^{-1}\\ &\times \mathbf{vec}\{\mathbf{I}_M\}\frac{\sigma}{\sqrt{\sigma^2 + 2\boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} + 2\sigma_\upsilon^2}}\end{aligned} \tag{48}$$

2. If υ_n is non-Gaussian, then by Taylor expansion we have

$$\begin{aligned}\lim_{n\to\infty} E[g(e_n)] \approx& E\left[\exp\left(-\frac{\upsilon_n^2}{2\sigma^2}\right)\right] + \frac{1}{2}\boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}}\\ &\times E\left[\left(\frac{\upsilon_n^2}{\sigma^4} - \frac{1}{\sigma^2}\right)\exp\left(-\frac{\upsilon_n^2}{2\sigma^2}\right)\right]\end{aligned} \tag{49}$$

$$\begin{aligned}\lim_{n\to\infty} E\left[g^2(e_n)\right] \approx& E\left[\exp\left(-\frac{\upsilon_n^2}{\sigma^2}\right)\right] + \boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}}\\ &\times E\left[\left(\frac{2\upsilon_n^2}{\sigma^4} - \frac{1}{\sigma^2}\right)\exp\left(-\frac{\upsilon_n^2}{\sigma^2}\right)\right]\end{aligned} \tag{50}$$

It follows that

$$\begin{aligned}S \approx& \eta^2\left(\boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} + E\left[\upsilon_n^2\right]\right)\mathbf{vec}^T\{\Upsilon\}(\mathbf{I}_{M^2} - \mathbf{F})^{-1}\mathbf{vec}\{\mathbf{I}_M\}\\ &\times\left(E\left[\exp\left(-\frac{\upsilon_n^2}{\sigma^2}\right)\right] + \boldsymbol{\varepsilon}^T_{\mathbf{w}}\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}}E\right.\\ &\left.\times\left[\left(\frac{2\upsilon_n^2}{\sigma^4} - \frac{1}{\sigma^2}\right)\exp\left(-\frac{\upsilon_n^2}{\sigma^2}\right)\right]\right)\end{aligned} \tag{51}$$

Remark. It is worth noting that (48) and (51) have been derived by using the approximation $\mathbf{w}_n \approx \mathbf{w}_{opt}$ at the steady state. In addition, the theoretical value for non-Gaussian noise case has been derived by taking the Taylor expansion of $g(e_n)$ around υ_n and omitting the higher-order terms. If the noise power is very large, the approximation is not accurate and hence, the derived values at steady state may deviate seriously from the actual results. The detailed derivations for (46) to (51) can be found in Appendix B.

4. Simulation results

In this section, we present simulation results to confirm the theoretical conclusions drawn in the previous section, and illustrate the superior performance of the proposed CMCC algorithm compared with the traditional CLMS algorithm [3], CAP algorithm [8] and CRLS algorithm [5] in non-Gaussian noise. The selection of kernel bandwidth is also discussed in the end.

4.1. Non-Gaussian noise models

Generally speaking, the non-Gaussian noise distributions can be divided into two categories: light-tailed (e.g., binary, uniform, etc.) and heavy-tailed (e.g., Laplace, Cauchy, mixed Gaussian, alpha-stable, etc.) distributions [16,26,28,33,38,39]. In the following experiments, six common non-Gaussian noise models including binary noise, uniform noise, Laplace noise, Cauchy noise, mixed Gaussian noise, and alpha-stable noise, are selected for performance evaluation. Descriptions of these non-Gaussian noises are as following:

1. Binary noise model: Standard binary noise [26] takes the values of either $\upsilon = 1$ or $\upsilon = -1$, with probability mass function $Pr\{\upsilon = 1\} = Pr\{\upsilon = -1\} = 0.5$.
2. Uniform noise model: The uniform noise is distributed over $[-1, 1]$, whose PDF takes the form [28]:

$$p(\upsilon) = \begin{cases}\frac{1}{2} & -1 \le \upsilon \le 1,\\ 0 & \texttt{otherwise}.\end{cases} \tag{52}$$

3. Laplace noise model: The Laplace noise is distributed with probability density function (PDF)[33]:

$$p(\upsilon) = \frac{1}{2}\exp^{-|\upsilon|} \tag{53}$$

4. Cauchy noise model: The PDF of the Cauchy noise is [33]

$$p(\upsilon) = \frac{1}{\pi(1+\upsilon^2)} \tag{54}$$

5. Mixed Gaussian noise model: The mixed Gaussian noise model is given by [16]:

$$(1-\theta)\mathcal{N}\left(\lambda_1, \upsilon_1^2\right) + \theta\mathcal{N}\left(\lambda_2, \upsilon_2^2\right) \tag{55}$$

where $\mathcal{N}\left(\lambda_i, \upsilon_i^2\right)(i = 1, 2)$ denote the Gaussian distributions with mean values λ_i and variances υ_i^2, and θ is the mixture

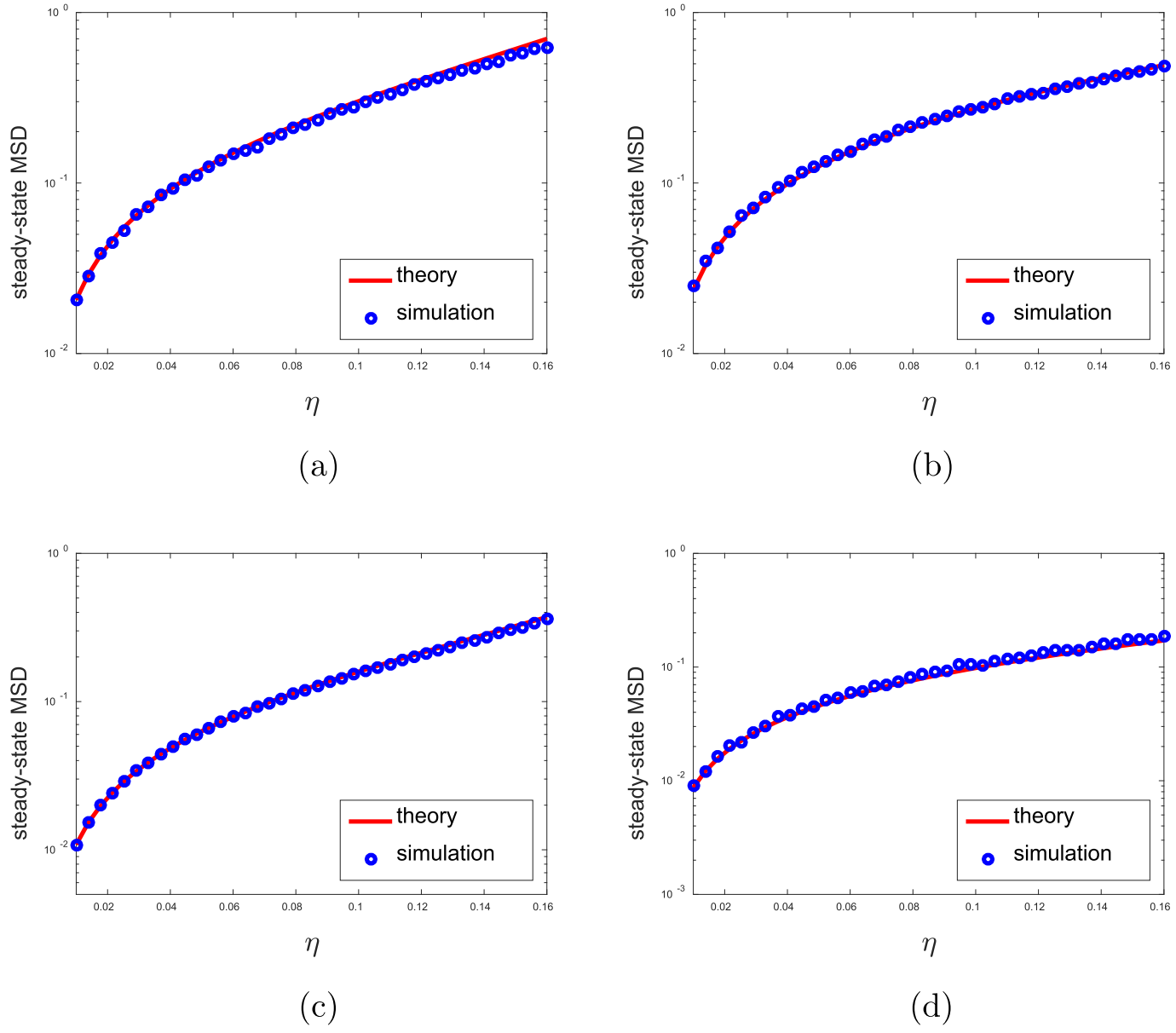

Fig. 2. Theoretical and simulated steady-state MSDs with different step-sizes η: (a) Gaussian noise ($\sigma = 8.0$, $\sigma_\upsilon^2 = 0.81$); (b) Binary noise ($\sigma = 2.0$, $\sigma_\upsilon^2 = 1.0$); (c) Uniform noise ($\sigma = 8.0$, $\sigma_\upsilon^2 = 0.33$); (d) Laplace noise ($\sigma = 1.0$, $\sigma_\upsilon^2 = 1.0$).

coefficient. Usually one can set θ to a small value and $\upsilon_2^2 \gg \upsilon_1^2$ to represent the impulsive noises (or large outliers). Therefore, we define the mixed Gaussian noise parameter vector as $V_{mix} = \left(\lambda_1, \lambda_2, \upsilon_1^2, \upsilon_2^2, \theta\right)$.

6. Alpha-stable noise model: The characteristic function of the alpha-stable noise is defined as [38,39]:

$$\psi(t) = \exp\{j\delta t - \gamma|t|^\alpha[1 + j\beta sgn(t)S(t,\alpha)]\} \tag{56}$$

in which

$$S(t,\alpha) = \begin{cases} \tan(\frac{\alpha\pi}{2}) & if \quad \alpha \neq 1 \\ \frac{2}{\pi}\log|t| & if \quad \alpha = 1 \end{cases} \tag{57}$$

From (56), one can observe that a stable distribution is completely determined by four parameters: 1) the characteristic factor α; 2) the symmetry parameter β; 3) the dispersion parameter γ; 4) the location parameter δ. So we define the alpha-stable noise parameter vector as $V_{alpha} = (\alpha, \beta, \gamma, \delta)$.

It is worth mentioning that, in the case of $\alpha = 2$, the alpha-stable distribution coincides with the Gaussian distribution, while $\alpha = 1, \delta = 0$ is the same as the Cauchy distribution.

4.2. Validation of steady-state MSD

In this experiment, we show the values of the theoretical and simulated steady-state MSDs of the CMCC in a linear channel with weight vector ($M = 7$)

$$\mathbf{w}^* = [0.332, -0.040, -0.094, 0.717, -0.652, -0.072, 0.580]^T \tag{58}$$

Assume that $K = 3$, $\mathbf{C}$ is full-rank, and the input covariance matrix $\mathbf{R}$ is positive-definite with $tr\{\mathbf{R}\} = M$ [6]. The input vectors are zero-mean multivrate Gaussian, and the disturbance noises considered include Gaussian noise, binary noise, uniform noise and Laplace noise. Fig. 2 shows the theoretical and simulated steady-state MSDs with different step-sizes, and Fig. 3 presents the theoretical and simulated steady-state MSDs with different noise vari-

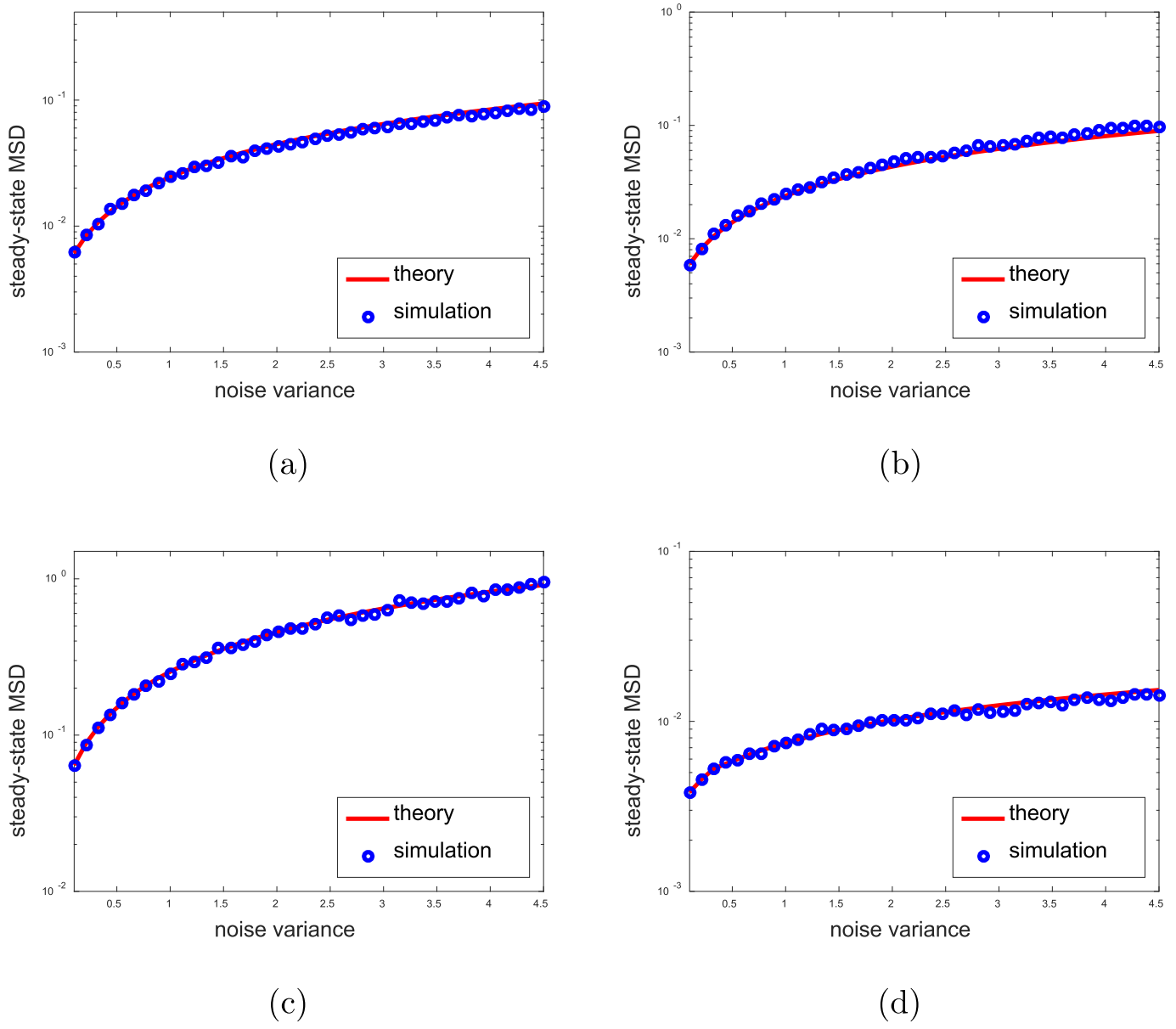

Fig. 3. Theoretical and simulated steady-state MSDs with noise variance σ_v^2: (a) Gaussian noise ($\eta = 0.01$, $\sigma = 8.0$); (b) Binary noise ($\eta = 0.01$, $\sigma = 6.0$); (c) Uniform noise ($\eta = 0.08$, $\sigma = 6.0$); (d) Laplace noise ($\eta = 0.01$, $\sigma = 0.8$).

ances. If not mentioned otherwise, simulation results are averaged over 500 independent Monte Carlo runs, and in each simulation, 5000 iterations are run to ensure the algorithms to reach the steady state, and the steady-state MSDs are obtained as averages over the last 200 iterations. Evidently, the steady-state MSDs are increasing with the step-size and noise variances increasing. In addition, the steady-state MSDs obtained from simulations match well with those theoretical results (computed by (48) for Gaussian noise and (51) for Non-Gaussian noise).

4.3. Linear system identification

We consider a linear system identification problem where the length of the adaptive filter is equal to that of the unknown system impulse response. Assume that the weight vector $\mathbf{w}^*$ of the unknown system, the constraint parameters $\mathbf{C}$ and $\mathbf{f}$, the input vectors, and the input covariance matrix $\mathbf{R}$ are the same as the previous experiment. In the simulations below, without mentioning otherwise, the sliding data length for CAP is set to 4, and the forgetting factor for CRLS is set to 0.998. The kernel bandwidth for CMCC is $\sigma = 2.0$.

First, we illustrate the performance of the proposed CMCC compared with CLMS, CAP and CRLS in four noise distributions. Simulation results are shown in Fig. 4. In the simulation, the mixed Gaussian noise parameters are set at $V_{mix} = (0, 0, 0.01, 100, 0.05)$, the alpha-stable noise parameters are set as $V_{alpha} = (1.5, 0, 0.4, 0)$, the laplace noise is zero-mean with standard deviation 5, and the cauchy noise is reduced to $\frac{1}{10}$. The step-sizes are chosen such that all the algorithms have almost the same initial convergence speed. As one can see clearly, the CMCC algorithm significantly outperforms other algorithms in terms of stability, and achieves much lower steady-state MSD.

Second, we demonstrate how the kernel bandwidth σ will influence the convergence performance of CMCC. Fig. 5 shows the convergence curves of CMCC with different σ, where the mixed Gaussian noise is chosen for measurement noise and

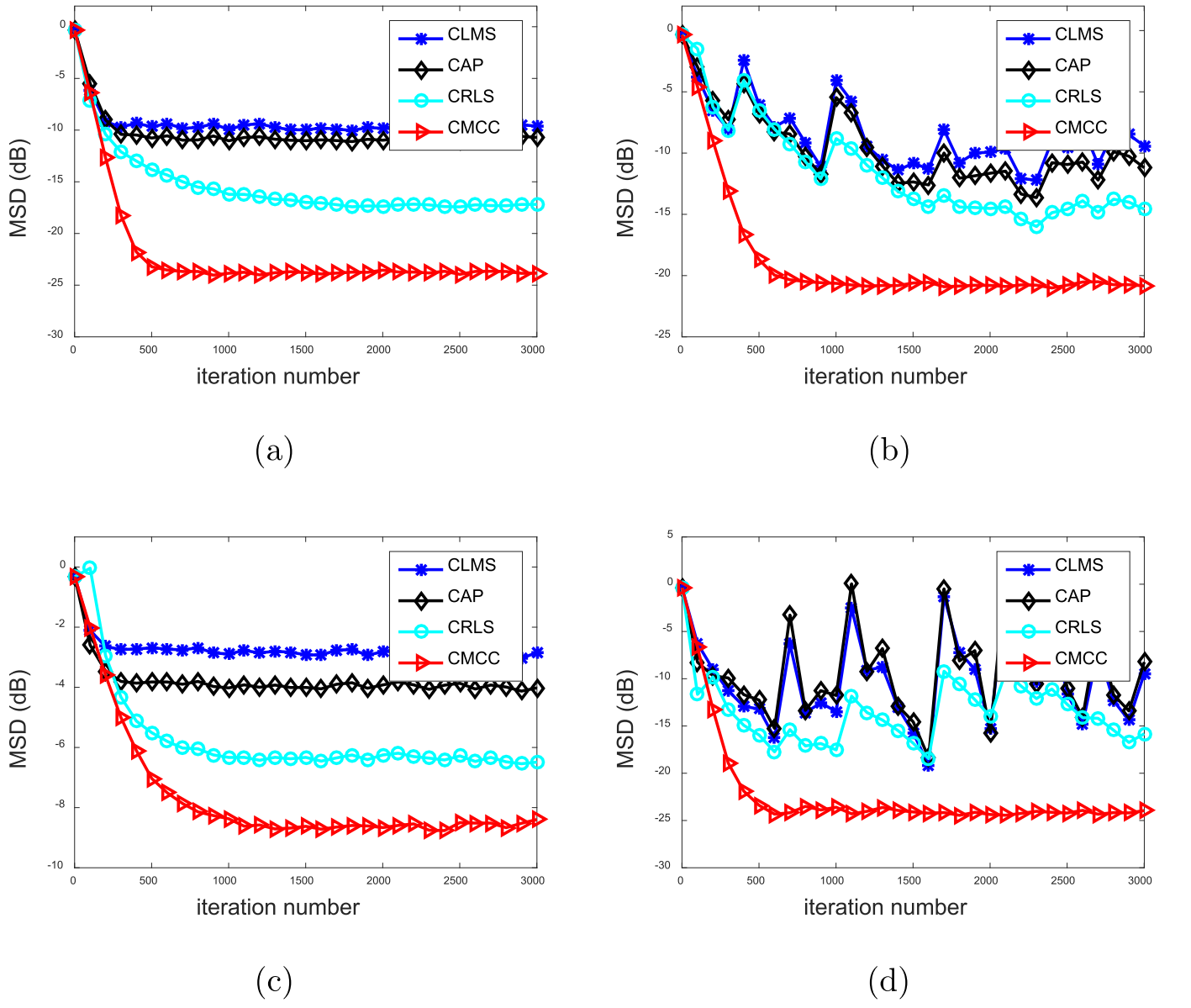

Fig. 4. Convergence curves of CLMS, CAP, CRLS and CMCC in different noises: (a) Mixed Gaussian noise; (b) Alpha-stable noise; (c) Laplace noise; (d) Cauchy noise.

the noise parameters are the same as the previous simulation. The step-sizes are set at $\eta = 0.06, 0.012, 0.01, 0.01, 0.01$ for $\sigma = 0.5, 2.0, 8.0, 16.0, 32.0$ respectively. Obviously, the kernel bandwidth has significant influence on the convergence behavior. In this example, the proposed algorithm achieves the lowest steady-state MSD when $\sigma = 2.0$. If the kernel bandwidth is too larger (e.g., $\sigma = 32.0$) or too small (e.g., $\sigma = 0.5$), the convergence performance of CMCC will become poor. We provide some useful properties later for kernel bandwidth selection in practical applications.

Third, we investigate the stability problem of the CMCC in different step-sizes. Fig. 6 illustrates the convergence performance with different step-sizes. The noise is still the mixed Gaussian noise with same parameters. Simulation results show that when the step-size is very large (such as $\eta \geq 0.5$), the CMCC will be divergent, which confirms the validity of the theoretical analysis of mean square stability in Section 3. Additionally, in this simulation, we calculate the value of $\frac{2}{2\lambda_{\max}+tr\{\Upsilon\}}$ (by (33)) to 0.278, not larger than 0.4, which also illustrates the effectiveness of (33).

4.4. Beamforming application

In this scenario, we consider a uniform linear array consisting of $M = 7$ omnidirectional sensors with an element spacing of half wavelength. We also assume that there are four users. Among them, the signal of one user is of interest, and is presumed to arrive at the direction-of-arrival (DOA) of $\varphi_d = 0^\circ$, while the other three signals are considered as interferers with DOAs of $\varphi_1 = -25^\circ$, $\varphi_2 = 30^\circ$, $\varphi_3 = 60^\circ$, respectively. We choose the constraint matrix $\mathbf{C} = [\mathbf{I}_{\frac{M-1}{2}}, \mathbf{0}, -\mathbf{J}_{\frac{M-1}{2}}]$ with $\mathbf{J}$ being a reversal matrix of size (an identity matrix with all rows in reversed order), and the response vector $\mathbf{f} = \mathbf{0}$ [6,8]. The measurement noise υ_n is the additive non-Gaussian noise, and the measured output of the unknown system is set to $d_n = \upsilon_n$. In the following simulations, the *signal-to-noise ratio* (SNR) is set to 0 dB, and the *interference-to-noise ratio* (INR) is set to 10 dB. The sliding data length for CAP is set to 4, and the forgetting factor for CRLS is set to 0.999. The kernel bandwidth σ is set at 20.

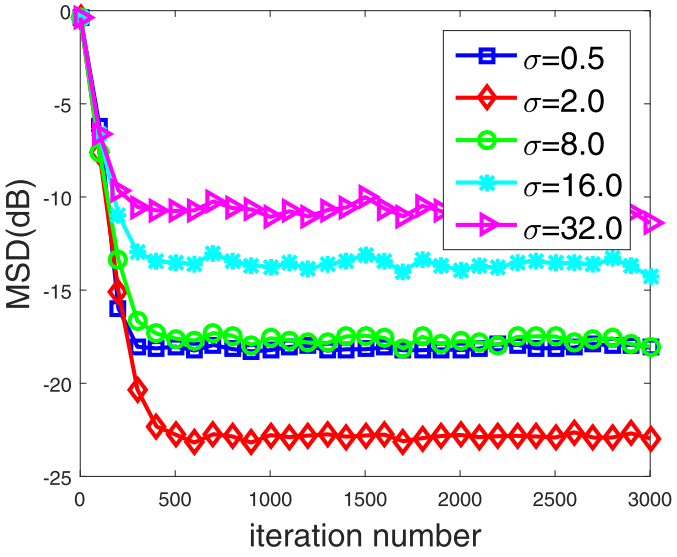

Fig. 5. Convergence curves of CMCC with different σ.

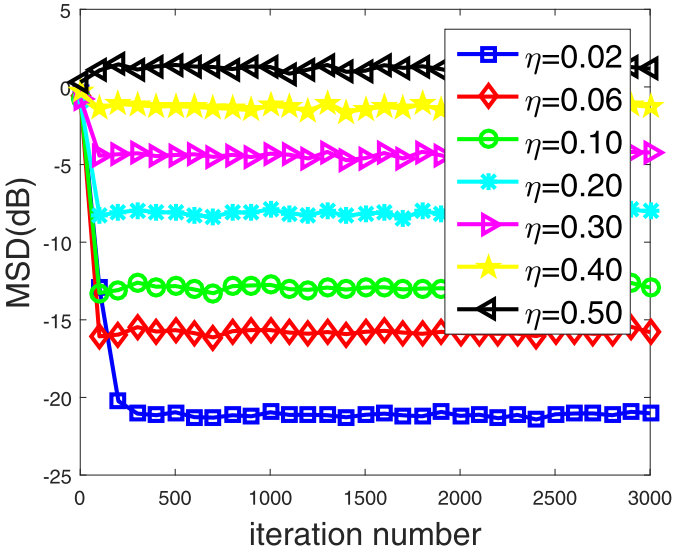

Fig. 6. Convergence curves of CMCC with different η.

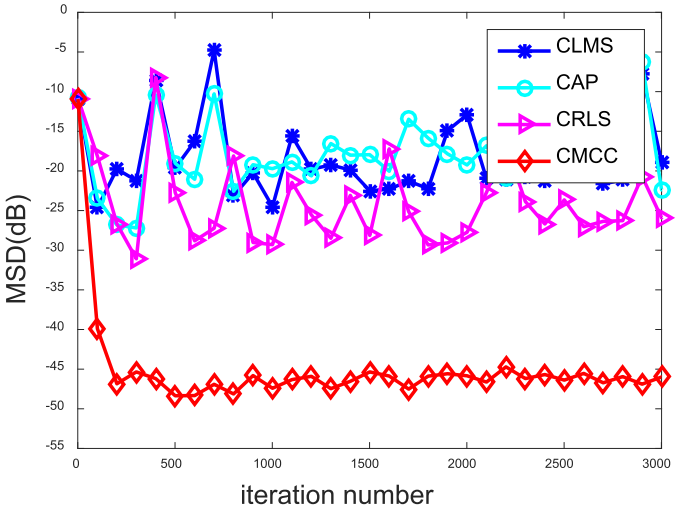

Fig. 7. Convergence curves of CLMS, CAP, CRLS and CMCC.

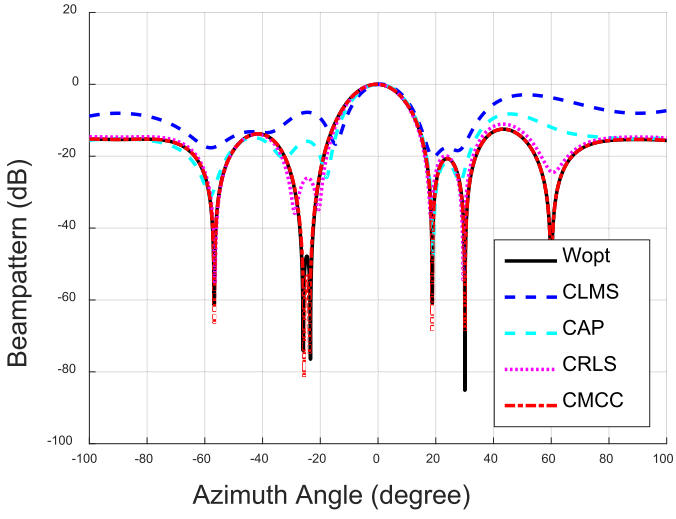

Fig. 8. Beampatterns of CLMS, CAP, CRLS and CMCC.

The convergence curves of CLMS, CAP, CRLS and CMCC in alpha-stable noise are illustrated in Fig. 7, and accordingly, the beampatterns of different methods are given in Fig. 8. The noise parameters are set at $V_{alpha} = (1.2, 0, 1.4, 0)$, and other parameters are chosen such that all algorithms have almost the same initial convergence rate. As one can see that the proposed algorithm performs best in all scenarios in term of MSD and beampattern shape. Furthermore, it has similar performance to the optimal beamformer after convergence.

Fig. 9 shows the steady-state MSDs of CLMS, CAP, CRLS and CMCC with different $\alpha = (0.6, 0.8, 1.0, 1.2, 1.4, 1.6)$ and different $\gamma = (1.2, 1.4, 1.6, 1.7, 1.8, 1.9)$ in 3-D space. Other parameters are the same as in the previous simulation for all algorithms. As expected, the proposed algorithm can achieve much better steady-state performance than CLMS, CAP and CRLS in all cases.

4.5. Parameter selection

The kernel bandwidth σ is an important free parameter in CMCC since it controls all robust properties of correntropy. An appropriate kernel bandwidth can provide an effective mechanism to eliminate the effect of outliers and noise.

According to the previous studies, some useful tricks for kernel bandwidth selection are as follows [12,17–19]:

1. If the data are plentiful, a small σ should be used so that high precision can be achieved; however, the kernel bandwidth must be selected to make a compromise between estimation efficiency and outlier rejection if the data are small.
2. As σ increases, the contribution of the higher-order moments decays faster, and the second-order moment plays a key role. Therefore, a large σ is frequently appropriate for Gaussian noises, while a small σ is usually adapt to non-Gaussian impulsive noises.
3. For a given noise environment, there is a relatively large range of σ that provides nearly optimal performance.

Currently, Silverman's rule, one of the most widely used methods in kernel density estimation, is often used to estimate σ. However, the limitation is that this method cannot obtain the best possible value. Therefore, in a practical application, σ is manually selected or optimized by scanning the performance.

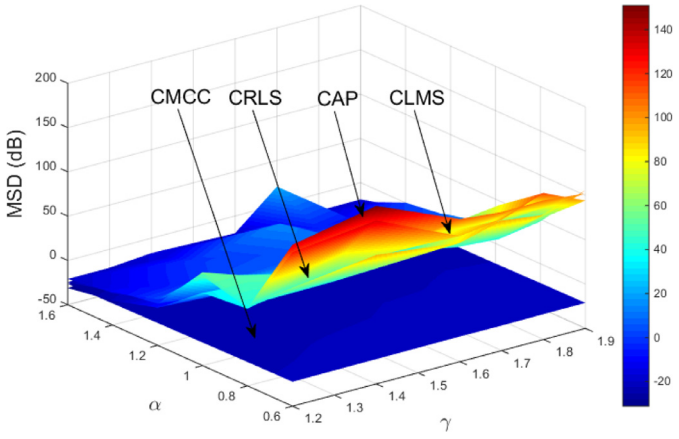

Fig. 9. Steady-state MSDs of CLMS, CAP, CRLS and CMCC in 3-D space.

5. Conclusion

In this paper, we have developed the constrained maximum correntropy criterion (CMCC) adaptive filtering algorithm by incorporating a linear constraint into the maximum correntropy criterion. We also studied the mean square convergence performance including the mean square stability and the steady-state mean square deviation of the proposed algorithm. Simulation results have confirmed the theoretical conclusions and shown that the new algorithm can significantly outperform the traditional methodes when the noise is of heavy-tailed non-Gaussian distribution.

Acknowledgments

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

Appendix A. Derivation of (12)

Based on (11), we can easily derive the following instantaneous weight update equation

$$\begin{aligned}\mathbf{w}_n =&\mathbf{w}_{n-1}+\eta\frac{\partial J_{CMCC}}{\partial \mathbf{w}}|_{\mathbf{w}=\mathbf{w}_{n-1}}\\ =&\mathbf{w}_{n-1}+\eta g(e_n)\left(d_n-\mathbf{w}_{n-1}^T\mathbf{x}_n\right)\mathbf{x}_n+\eta\mathbf{C}\boldsymbol{\xi}_n\end{aligned}\tag{A.1}$$

Due to

$$\mathbf{f}=\mathbf{C}^T\mathbf{w}_n=\mathbf{C}^T\left[\mathbf{w}_{n-1}+\eta\mathbf{C}\boldsymbol{\xi}_n+\eta g(e_n)\left(d_n-\mathbf{w}_{n-1}^T\mathbf{x}_n\right)\mathbf{x}_n\right]\tag{A.2}$$

we have

$$\boldsymbol{\xi}_n=\frac{1}{\eta}(\mathbf{C}^T\mathbf{C})^{-1}\left[\mathbf{f}-\mathbf{C}^T\mathbf{w}_{n-1}-\eta g(e_n)\left(d_n-\mathbf{w}_{n-1}^T\mathbf{x}_n\right)\mathbf{C}^T\mathbf{x}_n\right]\tag{A.3}$$

Substituting (A.3) into (A.1), and after some simple vector manipulations, we derive

$$\mathbf{w}_n=\mathbf{P}\left[\mathbf{w}_{n-1}+\eta g(e_n)(d_n-\mathbf{w}_{n-1}^T\mathbf{x}_n)\mathbf{x}_n\right]+\mathbf{q}\tag{A.4}$$

which is the CMCC algorithm.

Appendix B. Derivation of (46)~ (51)

Here we consider two cases below:

1. Gaussian noise case
Since $e_n=e_n^a+\upsilon_n$, in this case e_n is also zero-mean Gaussian. Let σ_e^2 be the variance of the error e_n. Then we have

$$\sigma_e^2=E\left[(e_n^a)^2\right]+\sigma_\upsilon^2\tag{B.1}$$

Using (21) and the approximation $\mathbf{w}_n\approx\mathbf{w}_{opt}$ at the steady-state, we obtain

$$e_n^a\approx(\mathbf{w}^*-\mathbf{w}_{opt})^T\mathbf{x}_n=\boldsymbol{\varepsilon}_\mathbf{w}^T\mathbf{x}_n\tag{B.2}$$

Therefore

$$\sigma_e^2\approx\boldsymbol{\varepsilon}_\mathbf{w}^T\mathbf{R}\boldsymbol{\varepsilon}_\mathbf{w}+\sigma_\upsilon^2\tag{B.3}$$

It follows that

$$\begin{aligned}\lim_{n\to\infty}E[g(e_n)]&=\lim_{n\to\infty}\frac{1}{\sqrt{2\pi}\sigma_e}\int_{-\infty}^{\infty}\exp\left(-\frac{e_n^2}{2\sigma^2}\right)\exp\left(-\frac{e_n^2}{2\sigma_e^2}\right)de_n\\&=\frac{\sigma}{\sqrt{\sigma^2+\sigma_e^2}}\approx\frac{\sigma}{\sqrt{\sigma^2+\boldsymbol{\varepsilon}_\mathbf{w}^T\mathbf{R}\boldsymbol{\varepsilon}_\mathbf{w}+\sigma_\upsilon^2}}\end{aligned}\tag{B.4}$$

$$\begin{aligned}\lim_{n\to\infty}E\left[g^2(e_n)\right]&=\lim_{n\to\infty}\frac{1}{\sqrt{2\pi}\sigma_e}\int_{-\infty}^{\infty}\exp\left(-\frac{e_n^2}{\sigma^2}\right)\exp\left(-\frac{e_n^2}{2\sigma_e^2}\right)de_n\\&=\frac{\sigma}{\sqrt{\sigma^2+2\sigma_e^2}}\approx\frac{\sigma}{\sqrt{\sigma^2+2\boldsymbol{\varepsilon}_\mathbf{w}^T\mathbf{R}\boldsymbol{\varepsilon}_\mathbf{w}+2\sigma_\upsilon^2}}\end{aligned}\tag{B.5}$$

Substituting (B.4) and (B.5) into (43), we obtain

$$\begin{aligned}S\approx&\eta^2\left(\boldsymbol{\varepsilon}_\mathbf{w}^T\mathbf{R}\boldsymbol{\varepsilon}_\mathbf{w}+\sigma_\upsilon^2\right)\mathbf{vec}^T\{\Upsilon\}(\mathbf{I}_{M^2}-\mathbf{F})^{-1}\\&\times\mathbf{vec}\{\mathbf{I}_M\}\frac{\sigma}{\sqrt{\sigma^2+2\boldsymbol{\varepsilon}_\mathbf{w}^T\mathbf{R}\boldsymbol{\varepsilon}_\mathbf{w}+2\sigma_\upsilon^2}}\end{aligned}\tag{B.6}$$

2. Non-Gaussian noise case
Taking the Taylor expansion of $g(e_n)$ with respect to e_n^a around υ_n, we have

$$g(e_n)=g(e_n^a+\upsilon_n)=g(\upsilon_n)+g'(\upsilon_n)e_n^a+\frac{1}{2}g''(\upsilon_n)(e_n^a)^2+o\left((e_n^a)^2\right)\tag{B.7}$$

where

$$g(\upsilon_n)=\exp\left(-\frac{\upsilon_n^2}{2\sigma^2}\right)\tag{B.8}$$

$$g'(\upsilon_n)=-\frac{\upsilon_n}{\sigma^2}\exp\left(-\frac{\upsilon_n^2}{2\sigma^2}\right)\tag{B.9}$$

$$g''(\upsilon_n)=\left(\frac{\upsilon_n^2}{\sigma^4}-\frac{1}{\sigma^2}\right)\exp\left(-\frac{\upsilon_n^2}{2\sigma^2}\right)\tag{B.10}$$

Thus

$$\begin{aligned}E[g(e_n)]\approx&E[g(\upsilon_n)]+\frac{1}{2}E\left[g''(\upsilon_n)\right]E\left[(e_n^a)^2\right]\\=&E\left[\exp\left(-\frac{\upsilon_n^2}{2\sigma^2}\right)\right]+\frac{1}{2}\boldsymbol{\varepsilon}_\mathbf{w}^T\mathbf{R}\boldsymbol{\varepsilon}_\mathbf{w}\\&\times E\left[\left(\frac{\upsilon_n^2}{\sigma^4}-\frac{1}{\sigma^2}\right)\exp\left(-\frac{\upsilon_n^2}{2\sigma^2}\right)\right]\end{aligned}\tag{B.11}$$

$$\begin{aligned}E\left[g^2(e_n)\right]\approx&E\left[g(\upsilon_n^2)\right]+E\left[(e_n^a)^2\right]E\left[g(\upsilon_n)g''(\upsilon_n)+g'^2(\upsilon_n))\right]\\=&E\left[\exp\left(-\frac{\upsilon_n^2}{\sigma^2}\right)\right]\\&+\boldsymbol{\varepsilon}_\mathbf{w}^T\mathbf{R}\boldsymbol{\varepsilon}_\mathbf{w}E\left[\left(\frac{2\upsilon_n^2}{\sigma^4}-\frac{1}{\sigma^2}\right)\exp\left(-\frac{\upsilon_n^2}{\sigma^2}\right)\right]\end{aligned}\tag{B.12}$$

Substituting (B.11) and (B.12) into (43) yields

$$S \approx \eta^2\left(\boldsymbol{\varepsilon}_{\mathbf{w}}^T\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}} + E\left[\upsilon_n^2\right]\right)\mathbf{vec}^T\{\Upsilon\}(\mathbf{I}_{M^2} - \mathbf{F})^{-1}\mathbf{vec}\{\mathbf{I}_M\} \times \left(E\left[\exp\left(-\frac{\upsilon_n^2}{\sigma^2}\right)\right] + \boldsymbol{\varepsilon}_{\mathbf{w}}^T\mathbf{R}\boldsymbol{\varepsilon}_{\mathbf{w}}E \times \left[\left(\frac{2\upsilon_n^2}{\sigma^4} - \frac{1}{\sigma^2}\right)\exp\left(-\frac{\upsilon_n^2}{\sigma^2}\right)\right]\right) \tag{B.13}$$

References

[1] M. de Campos, S. Werner, J. Apolinrio Jr., Constrained Adaptive Filters, Springer Berlin Heidelberg, 2004.
[2] Y. Li, C. Zhang, S. Wang, Low-complexity non-uniform penalized affine projection algorithm for sparse system identification, Circuits Syst. Signal Process. 35 (5) (2016) 1611–1624.
[3] O. Frost, An algorithm for linearly constrained adaptive array processing, Proc. IEEE 60 (8) (1972) 926–935.
[4] H. Van Trees, Detection, Estimation, and Modulation Theory Part IV: Optimum Array Processing, John Wiley & Sons, 2004.
[5] L. Resende, J. Romano, M. Bellanger, A fast least squares algorithm for linearly constrained adaptive filtering, IEEE Trans. Signal Process. 44 (5) (1996) 1168–1174.
[6] R. Arablouei, K. Dogancay, Reduced-complexity constrained recursive least-squares adaptive filtering algorithm, IEEE Trans. Signal Process. 60 (12) (2012) 6687–6692.
[7] R. Arablouei, K. Dogancay, Linearly-constrained recursive total least-squares algorithm, IEEE Signal Process. Lett. 19 (12) (2012) 821–824.
[8] S. Werner, J. Apolinrio Jr., M. de Campos, P.S.R. Diniz, Low-complexity constrained affine-projection algorithms, IEEE Trans. Signal Process. 53 (12) (2005) 4545–4555.
[9] K. Lee, Y. Baek, Y. Park, Nonlinear acoustic echo cancellation using a nonlinear postprocessor with a linearly constrained affine projection algorithm, IEEE Trans. Circuits Syst. II Exp. Briefs 62 (9) (2015) 881–885.
[10] A. Sayed, Fundamentals of Adaptive Filtering, John Wiley & Sons, 2003.
[11] B. Chen, Y. Zhu, J. Hu, J. Principe, System parameter identification: information criteria and algorithms, Newnes (2013).
[12] J. Principe, Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives, Springer, New York, NY, USA, 2010.
[13] K. Plataniotis, D. Androutsos, A. Venetsanopoulos, Nonlinear filtering of non–gaussian noise, J. Intell. Rob. Syst. 19 (2) (1997) 207–231.
[14] B. Weng, K. Barner, Nonlinear system identification in impulsive environments, IEEE Trans. Signal Process 53 (7) (2005) 2588–2594.
[15] D. Haddad, M. Petraglia, A. Petraglia, A unified approach for sparsity-aware and maximum correntropy adaptive filters, 2016 24th Eur. Signal Process. Conf. (EUSIPCO) (2016) 170–174.
[16] S. Zhao, B. Chen, J. Principe, Kernel adaptive filtering with maximum correntropy criterion, in Proc. Int. Joint Conf. Neural Netw. (IJCNN) (2011) 2012–2017.
[17] X. Zhang, K. Li, Z. Wu, Y. Fu, H. Zhao, B. Chen, Convex regularized recursive maximum correntropy algorithm, Signal Process. 129 (2016) 12–16.
[18] W. Liu, P. Pokharel, J. Principe, Correntropy: properties and applications in non-gaussian signal processing, IEEE Trans. Signal Processing 55 (11) (2007) 5286–5298.
[19] L. Shi, Y. Lin, Convex combination of adaptive filters under the maximum correntropy criterion in impulsive interference, IEEE Signal Process. Lett. 21 (11) (2014) 1385–1388.
[20] A. Singh, J. Principe, Using correntropy as a cost function in linear adaptive filters, in Proc. Int. Joint Conf. Neural Netw. (IJCNN) (2009) 2950–2955.
[21] B. Chen, L. Xing, H. Zhao, N. Zheng, J. Principe, Generalized correntropy for robust adaptive filtering, IEEE Trans. Signal Process. 64 (13) (2016) 3376–3387.
[22] R. He, B. Hu, W. Zheng, X. Kong, Robust principal component analysis based on maximum correntropy criterion, IEEE Trans. Image Process. 20 (6) (2011) 1485–1494.
[23] R. He, W. Zheng, B. Hu, Maximum correntropy criterion for robust face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1561–1576.
[24] R. Bessa, V. Miranda, J. Gama, Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting, IEEE Trans. Power Syst. 24 (4) (2009) 1657–1666.
[25] E. Hasanbelliu, L. Giraldo, J. Principe, Information theoretic shape matching, IEEE Trans. Pattern Anal. Mach. Intell. 36 (12) (2014) 2436–2451.
[26] B. Chen, Y. Zhu, J. Hu, Mean-square convergence analysis of ADALINE training with minimum error entropy criterion, IEEE Trans. Neural Netw. 21 (7) (2010) 1168–1179.
[27] R. Arablouei, K. Dogancay, On the mean-square performance of the constrained LMS algorithm, Signal Process. 117 (2015) 192–197.
[28] T. Al-Naffouri, A. Sayed, Adaptive filters with error nonlinearities: mean-square analysis and optimum design, EURASIP J. Appl. Signal Process. 4 (2001) 192–205.
[29] R. Arablouei, K. Dogancay, Linearly-constrained line-search algorithm for adaptive filtering, Electron. Lett. 48 (19) (2012) 1208–1209.
[30] Y. Li, Y. Wang, T. Jiang, Sparse-aware set-membership nlms algorithms and their application for sparse channel estimation and echo cancelation, Int. J. Electron. Commun. 70 (2016) 895–902.
[31] R. Arablouei, K. Dogancay, Performance analysis of linear-equality-constrained least squares estimation, IEEE Trans. Signal Process. 63 (14) (2015) 3802–3809.
[32] H. Lee, S. Yim, W. Song, z^2-proportionate diffusion LMS algorithm with mean square performance analysis, Signal Process. 131 (2017) 154–160.
[33] B. Chen, L. Xing, J. Liang, N. Zheng, J. Principe, Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion, IEEE Signal Process. Lett. 21 (7) (2014) 880–884.
[34] L. Isserlis, On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables, Biometrika 12 (1/2) (1918) 134–139.
[35] Y. Li, Y. Wang, T. Jiang, Norm-adaption penalized least mean square/fourth algorithm for sparse channel estimation, Signal Process. 128 (2016) 243–251.
[36] B. Lin, R. He, X. Wang, B. Wang, The steady-state mean-square error analysis for least mean p-order algorithm, IEEE Signal Process. Lett. 16 (3) (2009) 176–179.
[37] K. Abadir, J. Magnus, Matrix Algebra, NY: Cambridge University Press, 2005.
[38] J. Zhang, T. Qiu, A. Song, H. Tang, A novel correntropy based DOA estimation algorithm in impulsive noise environments, Signal Process. 104 (2014) 346–357.
[39] Z. Wu, S. Peng, B. Chen, H. Zhao, Robust hammerstein adaptive filtering under maximum correntropy criterion, Entropy 117 (10) (2015) 7149–7166.