electronics

MDPI

Article

Convex Regularized Recursive Minimum Error Entropy Algorithm

Xinyu Wang, Shifeng Ou and Ying Gao *

School of Physics and Electronic Information, Yantai University, Yantai 264005, China; zhangxueting0315@s.ytu.edu.cn (X.W.); ousfeng@ytu.edu.cn (S.O.)
* Correspondence: 201001000234@ytu.edu.cn

Abstract: It is well known that the recursive least squares (RLS) algorithm is renowned for its rapid convergence and excellent tracking capability. However, its performance is significantly compromised when the system is sparse or when the input signals are contaminated by impulse noise. Therefore, in this paper, the minimum error entropy (MEE) criterion is introduced into the cost function of the RLS algorithm in this paper, with the aim of counteracting the interference from impulse noise. To address the sparse characteristics of the system, we employ a universally applicable convex function to regularize the cost function. The resulting new algorithm is named the convex regularization recursive minimum error entropy (CR-RMEE) algorithm. Simulation results indicate that the performance of the CR-RMEE algorithm surpasses that of other similar algorithms, and the new algorithm excels not only in scenarios with sparse systems but also demonstrates strong robustness against pulse noise.

Keywords: minimum error entropy; recursive minimum error entropy; convex regularized recursive minimum error entropy

check for updates

Citation: Wang, X.; Ou, S.; Gao, Y. Convex Regularized Recursive Minimum Error Entropy Algorithm. *Electronics* **2024**, *13*, 992. https://doi.org/10.3390/electronics13050992

Academic Editor: Manohar Das

Received: 26 January 2024
Revised: 29 February 2024
Accepted: 3 March 2024
Published: 6 March 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Sparse adaptive filters have widespread applications in the field of signal processing, commonly employed in tasks such as acoustic echo cancellation [1,2] and communication channel identification [3]. Traditional adaptive filtering algorithms used for identifying sparse systems (referring to systems that contain many near-zero or a few large coefficients that need to be identified) often rely on prior knowledge of sparsity. In particular, many relevant algorithms have been proposed in this context: for the LMS algorithm, scholars have successively improved its cost function using the l_0-norm and l_1-norm regularization techniques [4]. This result significantly enhances the algorithm's convergence speed in sparse systems, reduces steady-state errors, and simultaneously improves its robustness. Subsequently, other researchers utilized a more general convex function to regularize the LMS algorithm [5], making the proposed algorithm more versatile compared to the former. In [6], the author applied convex regularization techniques to enhance the RLS algorithm. By incorporating a general regularization term into the RLS cost function, they derived a new type of sparse adaptive filtering algorithm, referred to as the convex regularized recursive least squares (CR-RLS) algorithm. This approach allows the use of any convex function for regularization. In recent years, the concept of sparsity has even been applied to kernel adaptive filtering, as mentioned in [7–10].

The aforementioned adaptive algorithms based on the mean-square error criterion perform well in Gaussian noise environments. However, in real-life scenarios, signals may be contaminated by impulsive noise, and the mean-square error criterion may not yield satisfactory results in the presence of impulsive noise [11]. With the development of information theoretic learning (ITL), the maximum correntropy criterion (MCC) [12–15] and the minimum error entropy criterion (MEE) [16–18] have gradually been introduced.

Due to their calculations involving higher-order statistical properties of the error signal, they exhibit less sensitivity to significant outliers, thus demonstrating a strong capability to effectively handle impulsive noise. Especially in non-Gaussian environments, information-theoretic criteria are notably superior to the most popular mean square error criterion. Furthermore, the literature indicates that under heavy-tailed noise distributions or when signals are contaminated by impulse noise or outliers, the MCC criterion and MEE criterion are significantly superior to the mean square error criterion.

In the present day, the maximum correlation entropy criterion and the minimum error entropy criterion have been successfully applied in adaptive filtering algorithms. For instance, a classic example is found in [19], where the correntropy-induced metric (CIM) penalized MCC criterion is utilized instead of the traditional minimum mean square error criterion for robust channel estimation against impulse noise. MCC is employed to combat impulse noise, while CIM is utilized to effectively exploit the sparsity of the channel. Hence, the proposed CIM-MCC algorithm demonstrates excellent performance. Furthermore, researchers have introduced the MCC criterion in kernel adaptive filtering [13], leading to the derivation of the kernel recursive maximum correntropy (KRMC) algorithm. Subsequent simulation results in alpha-stable noise environments for short-term chaotic time series prediction confirm the superior performance of KRMC. Similarly, the recursive minimum error entropy (RMEE) algorithm is proposed [20]. These algorithms exhibit strong suppression capabilities against pulse noise. Considering scenarios involving sparse systems, scholars have also proposed the convex regularized recursive maximum correntropy (CR-RMC) algorithm as mentioned in [21]. Inspired by the previous discussions, this paper introduces sparsity enhancements to the RMEE algorithm by incorporating a more general convex function as a sparse penalty term in the cost function. As a result, we obtain a sparse adaptive algorithm based on the MEE criterion, which is called the convex regularization recursive minimum error entropy (CR-RMEE) algorithm. Subsequent results reveal that this algorithm not only exhibits strong robustness in the presence of impulsive noise but also demonstrates good capabilities in identifying sparse systems.

The remaining part of the paper is organized as follows: In Section 2, relevant background knowledge is introduced, including basic concepts of adaptive filtering and minimum error entropy. We provide a detailed explanation of the proposed new algorithm in Section 3, including the derivation process and the selection of regularization factors. In Section 4, we conduct simulation experiments on the algorithm, considering both stationary and non-stationary systems and various noise environments, to test the algorithm's convergence, tracking performance, and robustness. An actual scenario of echo cancellation is also considered to thoroughly evaluate the algorithm's performance. Finally, the conclusion is presented in Section 5.

2. Basic Principles

When performing linear adaptive filtering, there is an input vector **u**, unknown tap parameter **w**, and desired response **d**. The desired response **d**(n) at each regression point can be obtained with the following expression:

$$\mathbf{d}(n) = \mathbf{w}_o^T \mathbf{u}(n) + \mathbf{v}(n), n = 1, 2, \cdots \tag{1}$$

where **v** is a zero-mean background noise with a variance of σ_v^2. In this context, the error signal can be represented as follows:

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{w}^T(n-1)\mathbf{u}(n) \tag{2}$$

where $\mathbf{w}(n-1)$ is the estimate of $\mathbf{w_o}$ at point $n-1$.

Additionally, in this study, we have made the following assumptions:

(a) The input signal is a white sequence with a mean of zero and is wide-sense stationary. The additive noise is zero-mean.
(b) The input sequences at different time instants and the noise are uncorrelated.

(c) The inputs at different time instants and the additive noise are uncorrelated.

$$E\left\{\mathbf{u}(m)\mathbf{u}^T(n)\right\} = \begin{cases} \sigma_u^2\mathbf{I}, m = n, \\ 0, m \neq n, \end{cases} \tag{3a}$$

$$E\left\{\mathbf{u}^T(m)\mathbf{u}(n)\right\} = \begin{cases} M\sigma_u^2, m = n, \\ 0, m \neq n, \end{cases} \tag{3b}$$

$$E\{v(m)v(n)\} = \begin{cases} \sigma_v^2, m = n, \\ 0, m \neq n. \end{cases} \tag{3c}$$

$$E\left\{\mathbf{u}^H(m)\mathbf{v}(n)\right\} = 0, m \neq n. \tag{4}$$

As a well-known learning criterion in Information Theoretic Learning (ITL), the Minimum Error Entropy was proposed by Alfred Renyi, building upon the foundation of Shannon entropy. Its expression can be derived from the Shannon entropy formula:

$$H_\alpha(X) = \frac{1}{1-\alpha}\log\left[\sum_{i=1}^{N} p_i^\alpha\right] = \frac{1}{1-\alpha}\log(V_\alpha(X)) \tag{5}$$

where α is a the order parameter, and you can change its value to alter the form of entropy expression. p_i represents the probability of occurrence of the $i-th$ data set; $V_\alpha(X)$ is referred to as the Information Potential (IP), and the second-order Renyi entropy with $\alpha = 2$ is more commonly used as follows:

$$V_2(X) = E\left[k_\sigma^2(X)\right] \tag{6}$$

where $\kappa_\sigma(X)$ is the Gaussian kernel with bandwidth σ. Kernel functions primarily describe the inner product of random variables in the feature space. The Gaussian kernel function is commonly used, and its expression is as follows:

$$\kappa_\sigma(x - x_o) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[\frac{(x-x_o)}{2\sigma^2}\right] \tag{7}$$

Due to the difficulty in obtaining the Probability Density Function (PDF) of a signal accurately in practical situations, it is common to estimate the PDF. The most common method for PDF estimation is based on the Parzen method. Assuming the input vector data are $\{u_1, u_2, \cdots, u_N\}$, the probability density after Parzen estimation can be represented as follows:

$$\hat{p}_{u,\sigma}(u) = \frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(u - u_i) \tag{8}$$

The estimate of its entropy can be derived by incorporating the kernel function into the second-order Renyi entropy formula:

$$\hat{H}_2(U) = -\log\left[\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\kappa_{\sqrt{2}\sigma}(u_i - u_j)\right] \tag{9}$$

where u_i and u_i represent input signals at different time instants i and j. When replacing the input in the kernel function with the error e, the error entropy $\hat{H}_2(e)$ of the system can be defined.

From the derived error entropy formula above, based on the definition of error entropy, it can be concluded that when the error entropy is minimized, the system error signal becomes more stable. This implies that the system's filtering algorithm has achieved

convergence. Therefore, the cost function for algorithms based on the minimum error entropy criterion can be expressed as follows:

$$\begin{aligned} J_{MEE}(\mathbf{w}) &= \min\left[\hat{H}_2(e)\right] \\ &= \min\left\{-\log\left[\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\kappa_{\sqrt{2}\sigma}(e_i - e_j)\right]\right\} \end{aligned} \tag{10}$$

Because this function is a logarithmic function and monotonically increasing, the cost formula can be simplified to

$$\begin{aligned} J_{MEE}(\mathbf{w}) &= \max\left[\hat{V}_{2,\sigma}(e)\right] \\ &= \max\left\{\left[\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\kappa_{\sqrt{2}\sigma}(e_i - e_j)\right]\right\} \end{aligned} \tag{11}$$

where $\mathbf{w}$ represents the tap parameters of the adaptive filter, and $\hat{V}_{2,\sigma}(e)$ represents the estimate of the second-order information potential.

3. Proposed Algorithm

In this chapter, we first proposed a new cost function and derived the novel CR-RMEE algorithm based on it. We also introduced the updated formula for the regularization factor. Finally, towards the end of this chapter, we conducted a mathematical analysis of the convergence of the new algorithm, demonstrating its theoretical convergence.

3.1. CR-RMEE Algorithm

The traditional MEE cost function, as shown in (11), leads to the cost function for the exponential forgetting factor $(0 < \lambda < 1)$ based on the MEE criterion as follows:

$$J(\mathbf{w}) = \frac{1}{L^2}\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\lambda^{i+j}G_\sigma(e_i - e_j) \tag{12}$$

where L is the length of the sliding window inside the filter, $G_\sigma(\cdot)$ is the Gaussian kernel function, and σ is the kernel width.

We incorporate a convex function containing the weight vector into it for regularization, which allows the regularized cost function to leverage some prior knowledge about the unknown system, such as sparsity. In this paper, the regularized MEE cost function is as follows:

$$J(\mathbf{w}) = \frac{1}{L^2}\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\lambda^{i+j}G_\sigma(e_i - e_j) - \gamma_n f(\mathbf{w}) \tag{13}$$

where $f(\cdot)$ is a convex function, γ_n is a variable constant known as the regularization factor. The role of γ_n is to maintain a balance between the MEE cost function and the regularization term. When γ_n increases or decreases, the impact of the regularization term on the adaptive algorithm also changes accordingly.

There are subgradients for the regularization convex function in the cost function. The collection of subgradients is called subdifferential, which is denoted by $\partial f(\mathbf{w})$. We denote a subgradient vector of $f(\cdot)$ at n with $\nabla^s f(n) \in \partial f(n)$.

Therefore, the gradient of the cost function (13) can be represented as follows:

$$\nabla J(\mathbf{w}) = \frac{1}{L^2\sigma^2}\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\lambda^{i+j}(\mathbf{u}_i - \mathbf{u}_j)G_\sigma(e_i - e_j)(e_i - e_j) - \gamma_n\{\nabla^s f(\mathbf{w})\} \tag{14}$$

To simplify the computation, at each time point n (the following derivations of equations are all under time point n) we have

$$\begin{cases} \mathbf{u}_i = \mathbf{u}(n-i), \\ d_i = d(n-i), \\ e_i = e(n-i), i = 0, 1, \cdots, L-1. \end{cases} \tag{15}$$

Then, we can obtain

$$\begin{cases} \mathbf{U}_L = [\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_{L-1}] = [\mathbf{u}_0, \mathbf{U}_{L-1}], \\ \mathbf{D}_L = [d_0, d_1, \cdots, d_{L-1}]^T = [d_0, \mathbf{D}_{L-1}]^T, \\ \varepsilon_L = [e_0, e_1, \cdots, e_{L-1}]^T = [e_0, \varepsilon_{L-1}]^T = \mathbf{D}_L - \mathbf{U}_L^T \mathbf{w}. \end{cases} \tag{16}$$

So, the gradient of the cost function (14) can be rewritten as follows:

$$\begin{aligned} \nabla J(\mathbf{w}) &= \tfrac{2}{L^2\sigma^2} \mathbf{U}_L (\mathbf{P}_L - \mathbf{Q}_L) \boldsymbol{\varepsilon}_L - \gamma_n \{\nabla^s f(\mathbf{w})\} \\ &= \tfrac{2}{L^2\sigma^2} \mathbf{U}_L \boldsymbol{\Phi}_L \boldsymbol{\varepsilon}_L - \gamma_n \{\nabla^s f(\mathbf{w})\} \end{aligned} \tag{17}$$

where

$$\begin{cases} [\mathbf{Q}]_{(i+1)(j+1)} = \lambda^{i+j} G_\sigma(e_i - e_j), i, j = 0, 1, \cdots, L-1, \\ [\mathbf{P}]_{(i+1)(j+1)} = \begin{cases} \sum\limits_{k=0}^{L-1} \lambda^{i+j} G_\sigma(e_i - e_k), i = j, \\ 0, i \neq j, \end{cases} \\ \boldsymbol{\Phi}_L = \mathbf{P}_L - \mathbf{Q}_L = \begin{bmatrix} \phi_0 & \varphi^T \\ \varphi & \lambda^2 \boldsymbol{\Phi}_{L-1} \end{bmatrix}, \\ [\varphi]_i = \lambda^i G_\sigma(e_i - e_0), j = 0, 1, \cdots, L-1 \\ \phi_0 = \sum\limits_{k=1}^{L-1} \lambda^k G_\sigma(e_0 - e_k). \end{cases} \tag{18}$$

Setting the gradient equal to zero, we can solve for

$$\mathbf{w} = \left(\mathbf{U}_L \boldsymbol{\Phi}_L \mathbf{U}_L^T\right)^{-1} \left[\mathbf{U}_L \boldsymbol{\Phi}_L \mathbf{D}_L - \frac{1}{2} L^2 \sigma^2 \gamma_n \{\nabla^s f(\mathbf{w})\}\right] \tag{19}$$

Let

$$\begin{cases} \mathbf{R}_L = \mathbf{U}_L \boldsymbol{\Phi}_L \mathbf{U}_L^T \\ \mathbf{r}_L = \mathbf{U}_L \boldsymbol{\Phi}_L \mathbf{D}_L \end{cases} \tag{20}$$

Then, (19) can be rewritten as follows:

$$\mathbf{w} = {\mathbf{R}_L}^{-1} \left[\mathbf{r}_L - \frac{1}{2} L^2 \sigma^2 \gamma_L \{\nabla^s f(\mathbf{w})\}\right] \tag{21}$$

where $\mathbf{R}_L$ is the weighted autocorrelation matrix of the input signal and can be rewritten as

$$\mathbf{R}_L = \lambda^2 \mathbf{U}_{L-1} \boldsymbol{\Phi}_{L-1} \mathbf{U}_{L-1}^T + \left[\mathbf{u}_0 \phi_0 \mathbf{u}_0^T + \mathbf{U}_{L-1} \varphi \mathbf{u}_0^T + \mathbf{u}_0 \varphi \mathbf{U}_{L-1}^T\right] \tag{22}$$

$\mathbf{r}_L$ is the weighted cross-correlation vector between the input signal and the desired response and can be rewritten as

$$\mathbf{r}_L = \lambda^2 \mathbf{U}_{L-1} \boldsymbol{\Phi}_{L-1} \mathbf{D}_{L-1} + \left[\mathbf{u}_0 \phi_0 d_0 + \mathbf{U}_{L-1} \varphi d_0 + \mathbf{u}_0 \varphi \mathbf{D}_{L-1}^T\right] \tag{23}$$

Under the conditions of Assumption (b) and Assumption (c), the following expectations have the following properties:

$$\begin{cases} E\left[\mathbf{U}_{L-1} \varphi \mathbf{u}_0^T + \mathbf{u}_0 \varphi \mathbf{U}_{L-1}^T\right] = 0 \\ E\left[\mathbf{U}_{L-1} \varphi d_0 + \mathbf{u}_0 \varphi \mathbf{D}_{L-1}^T\right] = 0 \end{cases} \tag{24}$$

So, (22) and (23) can be rewritten as follows:

$$\begin{aligned}\mathbf{R}_L &= \lambda^2 \mathbf{U}_{L-1}\mathbf{\Phi}_{L-1}\mathbf{U}_{L-1}^T + \mathbf{u}_0\phi_0\mathbf{u}_0^T \\ &= \lambda^2 \mathbf{R}_{L-1} + \mathbf{u}_0\phi_0\mathbf{u}_0^T\end{aligned} \tag{25}$$

$$\begin{aligned}\mathbf{r}_L &= \lambda^2 \mathbf{U}_{L-1}\mathbf{\Phi}_{L-1}\mathbf{D}_{L-1} + \mathbf{u}_0\phi_0 d_0 \\ &= \lambda^2 \mathbf{r}_{L-1} + \mathbf{u}_0\phi_0 d_0\end{aligned} \tag{26}$$

Define a new vector θ_L:

$$\theta_L = \mathbf{r}_L - \frac{1}{2}L^2\sigma^2\gamma_L\{\nabla^s f(\mathbf{w})\} \tag{27}$$

By substituting the update (26) into (27), assuming that γ_L and $\nabla^s f(\mathbf{w})$ change only slightly at each time point [6], the vector θ_L can also be recursively updated as follows:

$$\begin{aligned}\theta_L &= \lambda^2\mathbf{r}_{L-1} + \mathbf{u}_0\phi_0 d_0 - \tfrac{1}{2}L^2\sigma^2\gamma_L\{\nabla^s f(\mathbf{w})\} \\ &= \lambda^2\Big(\mathbf{r}_{L-1} - \tfrac{1}{2}L^2\sigma^2\gamma_{L-1}\{\nabla^s f(\mathbf{w})\}\Big) + \mathbf{u}_0\phi_0 d_0 \\ &-\tfrac{1}{2}L^2\sigma^2\gamma_L\{\nabla^s f(\mathbf{w})\} + \tfrac{1}{2}\lambda^2 L^2\sigma^2\gamma_{L-1}\{\nabla^s f(\mathbf{w})\} \\ &\approx \lambda^2\theta_{L-1} + \mathbf{u}_0\phi_0 d_0 - \tfrac{1}{2}\big(1-\lambda^2\big)L^2\sigma^2\gamma_{L-1}\{\nabla^s f(\mathbf{w})\}\end{aligned} \tag{28}$$

Then, (21) can be rewritten as

$$\mathbf{w} = \mathbf{P}_L\theta_L \tag{29}$$

where $\mathbf{P}_L = {\mathbf{R}_L}^{-1}$. By using the matrix inversion lemma [20], we can obtain the updated formula

$$\mathbf{P}_L = \lambda^{-2}\Big(\mathbf{P}_{L-1} - \mathbf{k}_L\mathbf{u}_o^T\mathbf{P}_{L-1}\Big) \tag{30}$$

The gain matrix $\mathbf{k}_L$ is given by

$$\mathbf{k}_L = \mathbf{P}_{L-1}\mathbf{u}_o\left(\frac{\lambda^2}{\phi_0} + \mathbf{u}_o^T\mathbf{P}_{L-1}\mathbf{u}_o\right)^{-1} \tag{31}$$

Rewriting (29) recursively, we can obtain the following updated formula:

$$\mathbf{w}_L = \mathbf{w}_{L-1} + \mathbf{k}_L e_0 - \frac{1}{2}L^2\sigma^2\Big(1-\lambda^2\Big)\gamma\mathbf{P}_{L-1}\{\nabla^s f(\mathbf{w})\} \tag{32}$$

3.2. Selection of the Regularization Factor

The use of regularization functions in the field of adaptive filtering is because they can provide some prior knowledge about the unknown system. Here, we assume that the prior knowledge is an upper bound on the regularization function, i.e., $f(w) \le \rho$, where ρ is a constant. Let $\hat{\mathbf{w}}(n)$ be the solution obtained from the cost function without regularization, and $\varepsilon = \mathbf{w} - \mathbf{w}_o$ and $\hat{\varepsilon} = \hat{\mathbf{w}} - \mathbf{w}_o$ represent the differences between $\mathbf{w}$ and $\hat{\mathbf{w}}$ with respect to the true weights. Substituting the difference expressions into (19), we obtain

$$\varepsilon(n) = \hat{\varepsilon}(n) - \frac{1}{2}L^2\sigma^2\mathbf{P}(n)\gamma(n)\{\nabla^s f[\mathbf{w}(n)]\} \tag{33}$$

The squared deviation of ε_n from D_n can be computed as follows:

$$\begin{aligned}D(n) &= \varepsilon^T(n)\varepsilon(n) = \|\varepsilon(n)\|_2^2 = \hat{D}(n) - L^2\sigma^2\mathbf{P}(n)\gamma(n)\{\nabla^s f[\mathbf{w}(n)]\}\hat{\varepsilon}(n) \\ &+\tfrac{1}{4}L^4\sigma^4\gamma^2(n)\|\mathbf{P}(n)\{\nabla^s f[\mathbf{w}(n)]\}\|_2^2\end{aligned} \tag{34}$$

where $\hat{D}(n)$ is the squared deviation from $\hat{\varepsilon}(n)$. One can obtain the following theorem from the (34) [6].

Theorem 1. *$D(n) \leq \hat{D}(n)$ if $\gamma(n) \leq [0, \max(\hat{\gamma}(n), 0)]$, where*

$$\hat{\gamma}(n) = 4\frac{\nabla^s f(\mathbf{w}(n))\mathbf{P}(n)\hat{\varepsilon}(n)}{L^2\sigma^2\|\mathbf{P}(n)\{\nabla^s f[\mathbf{w}(n)]\}\|_2^2} \tag{35}$$

Proof. From (34), when $\frac{1}{4}L^4\sigma^4\gamma^2(n)\|\mathbf{P}(n)\gamma(n)\{\nabla^s f[\mathbf{w}(n)]\}\|_2^2 - L^2\sigma^2\mathbf{P}(n)\gamma(n)\{\nabla^s f[\mathbf{w}(n)]\}\hat{\varepsilon}(n) \leq 0$, then we obtain $D(n) \leq \hat{D}(n)$. Consequently, the inequality can be solved as follows:

$$\begin{cases} 0 \leq \gamma(n) \leq 4\frac{\nabla^s f(\mathbf{w}_n)\mathbf{P}_n\hat{\varepsilon}_n}{L^2\sigma^2\|\mathbf{P}_n\{\nabla^s f[\mathbf{w}(n)]\}\|_2^2}, & \nabla^s f[\mathbf{w}(n)]\mathbf{P}(n)\hat{\varepsilon}(n) \geq 0 \\ \gamma(n) = 0, & \nabla^s f[\mathbf{w}(n)]\mathbf{P}(n)\hat{\varepsilon}(n) < 0 \end{cases} \tag{36}$$

From Theorem 1, it can be seen that if the method described in (36) is used to adaptively update the regularization factor, the performance of the CR-RMEE algorithm will be close to or even surpass the standard RMEE algorithm. However, since in practical use, we cannot obtain the true weight vector, $\hat{\gamma}(n)$ is also not easy to calculate directly. Inspired by the work in [6], and using the subgradient definition of a convex function, $\hat{\gamma}(n)$ can be approximated by the following expression when the input is a white sequence and n is sufficiently large:

$$\gamma(n) \approx \gamma\prime(n) = 4\frac{\frac{tr[\mathbf{P}(n)]}{N}\{f[\mathbf{w}(n)] - \rho\} + \nabla^s f[\mathbf{w}(n)]^T\mathbf{P}(n)\boldsymbol{\varepsilon}\prime(n)}{L^2\sigma^2\|\mathbf{P}(n)\{\nabla^s f[\mathbf{w}(n)]\}\|_2^2} \tag{37}$$

where $\boldsymbol{\varepsilon}'(n) = \hat{\mathbf{w}}(n) - \mathbf{w}(n)$, meaning the difference between the unregularized coefficients and the regularized coefficients, and $tr(\cdot)$ represents a tracking operator. According to (38), $\gamma(n)$ can be automatically updated to $\gamma(n) = \max[\gamma'(n), 0]$ in each iteration, and the computational complexity for each automatic update is only $O(N^2)$, which effectively saves time in the search for the optimal value. □

3.3. Convergence Analysis

Whether an algorithm converges can be determined by observing the behavior when it reaches a steady state characterized by $E\{\widetilde{\mathbf{w}}(n)\}$, where $\widetilde{\mathbf{w}}(n) = \mathbf{w}_o - \mathbf{w}(n)$. So, from Equation (32), it can be concluded that

$$\widetilde{\mathbf{w}}(n) = \widetilde{\mathbf{w}}(n-1) - \mathbf{k}(n)e(n) + \frac{1}{2}L^2\sigma^2\left(1-\lambda^2\right)\gamma(n-1)\mathbf{P}(n-1)\{\nabla^s f(\mathbf{w})\} \tag{38}$$

Substituting Equation (1) into it, we obtain

$$\begin{aligned} \widetilde{\mathbf{w}}(n) \quad &= \widetilde{\mathbf{w}}(n-1) - \mathbf{k}(n)\mathbf{u}(n)\widetilde{\mathbf{w}}(n-1) - \mathbf{k}(n)\mathbf{v}(n) \\ &+\tfrac{1}{2}L^2\sigma^2(1-\lambda^2)\gamma(n-1)\mathbf{P}(n-1)\{\nabla^s f(\mathbf{w})\} \end{aligned} \tag{39}$$

Substituting (30) and (31) into the previous expression yields

$$\begin{aligned} \widetilde{\mathbf{w}}(n) = \quad &\lambda^2\mathbf{P}(n)\mathbf{R}(n-1)\widetilde{\mathbf{w}}(n-1) - \mathbf{P}(n)\mathbf{u}(n)\phi(n)\mathbf{v}(n) \\ &+\tfrac{1}{2}L^2\sigma^2(1-\lambda^2)\gamma(n-1)\mathbf{P}(n-1)\{\nabla^s f(\mathbf{w})\} \end{aligned} \tag{40}$$

From (25), the steady-state average value of A can be obtained as follows:

$$E\{\mathbf{R}(n)\} = \lambda^2 E\{\mathbf{R}(n-1)\} + E\{\phi(n)\}E\left\{\mathbf{u}(n)\mathbf{u}^T(n)\right\} \tag{41}$$

Assuming the input signal $\mathbf{u}(n)$ is a wide-sense stationary process, combined with the assumptions mentioned earlier (a) and (41), we can obtain

$$E\{\mathbf{R}(n)\} = \lambda^2 E\{\mathbf{R}(n-1)\} + E\{\phi(n)\}\sigma_u^2\mathbf{I} \tag{42}$$

where $\mathbf{I}$ is the unit matrix of the same dimension as $\mathbf{R}(n)$. Summarizing the above expressions, we obtain

$$E\{\mathbf{R}(n)\} = \left(1-\lambda^2\right)^{-1} E\{\phi(n)\}\sigma_u^2\mathbf{I} \tag{43}$$

Although it is known from the above that $\mathbf{P}(n) = \mathbf{R}(n)^{-1}$, calculating the steady-state average of $\mathbf{P}(n)$ is still very challenging. Therefore, it is commonly approximated using the following expression [19].

$$E\{\mathbf{P}(n)\} \approx E^{-1}\left\{\mathbf{P}^{-1}(n)\right\} = \left(1-\lambda^2\right)E^{-1}\{\phi(n)\}\sigma_u^{-2}\mathbf{I} \tag{44}$$

From (37), it can be easily deduced that $\lim_{n\to\infty}\gamma(n) = 0$, and under the conditions of assumption (a), $E\{\mathbf{v}(n)\} = 0$ can also be obtained. Therefore, the steady-state value of $\widetilde{\mathbf{w}}(n)$ can be expressed as

$$E\{\widetilde{\mathbf{w}}(n)\} = \lambda^2 E\{\widetilde{\mathbf{w}}(n-1)\} \tag{45}$$

For $0 < \lambda < 1$, it follows that

$$\lim_{n\to\infty} E\{\widetilde{\mathbf{w}}(n)\} = 0 \tag{46}$$

Therefore, we can demonstrate that the CR-RMEE algorithm proposed by us possesses convergence.

4. Simulation Experiments

In order to compare the advantages of the proposed algorithm, our simulations were conducted separately in steady and non-steady environments, and all simulation results were obtained through 1000 independent Monte Carlo experiments. The simulations use normalized mean squared deviation (NMSD) as the performance metric to evaluate the algorithms. A smaller value of NMSD indicates better algorithm performance, and the stability of the resulting waveform is also an important indicator of algorithm stability. NMSD can be obtained using the following formula:

$$NMSD(n) = 10\log_{10}\left(\frac{\|\mathbf{w}(n)-\mathbf{w}_o\|^2}{\|\mathbf{w}_o\|^2}\right) \tag{47}$$

In the simulations, the unknown weight vector $\mathbf{w}_o$ is a 20×1 vector with S non-zero tap weights, and the sliding window width is 8. Unless otherwise stated, the input $\mathbf{u}$ is a white Gaussian random sequence, and the additive noise is impulsive with a mixed Gaussian distribution, $v(n) \sim 0.95N(0,0.01) + 0.05N(0,25)$.

For the selection of the function in the regularization term, we will introduce a sparsity-inducing convex penalty function. Since the l_0 norm is a non-convex function, we will use its approximate function instead as follows:

$$\|\mathbf{w}\|_0 \approx f^{\beta}(\mathbf{w}) = \sum_{k=0}^{N-1}\left(1-e^{-\beta|\mathbf{w}_k|}\right) \tag{48}$$

where β is a constant, and the approximate gradient of the above expression can be represented using the following formula:

$$\nabla^s f^{\beta}(\mathbf{w})_k \approx \begin{cases} \beta\mathrm{sgn}(w_k) - \beta^2 w_k, |w_k| \le \frac{1}{\beta} \\ 0, \quad elsewhere \end{cases} \tag{49}$$

Firstly, we present the performance of the CR-RMEE algorithm in a stationary sparse environment. For comparison, we also present the performance of the LMS algorithm, RLS algorithm, RMC algorithm, and the original RMEE algorithm. We set the unknown

system as a sparse system with $S = 1$. For the RMC algorithm, RMEE algorithm, CR-RMC algorithm, and our proposed CR-RMEE algorithm, the same $\sigma = 1$ is chosen. Moreover, the step size for the LMS algorithm is 0.01. As for the choice of λ, RLS, RMC, RMEE, CR-RMC, and CR-RMEE algorithms are, respectively, chosen as $\lambda = 0.992$, $\lambda = 0.942$, $\lambda^2 = 0.996$, $\lambda^2 = 0.996$, and $\lambda^2 = 0.996$. As we can see from Figure 1, it is evident that, with the initial convergence rates of various algorithms set to be similar, the NMSD of the LMS, RLS, and RMC algorithms is relatively large. Additionally, both the LMS and RMC algorithms exhibit significant fluctuations after convergence. Although both the RMEE and CR-RMC algorithms have very small NMSD values, the RMEE algorithm exhibits faster convergence, while the CR-RMC algorithm ultimately achieves a lower steady-state error. Compared to the LMS and RMC algorithms, our proposed CR-RMEE algorithm shows significantly reduced fluctuations within an acceptable range. Combining the advantages of the fast convergence of the RMEE algorithm and the low steady-state error of the CR-RMC algorithm, the CR-RMEE algorithm even achieves the lowest NMSD. This simulation experiment demonstrates that our proposed new algorithm exhibits excellent performance in a steady sparse environment.

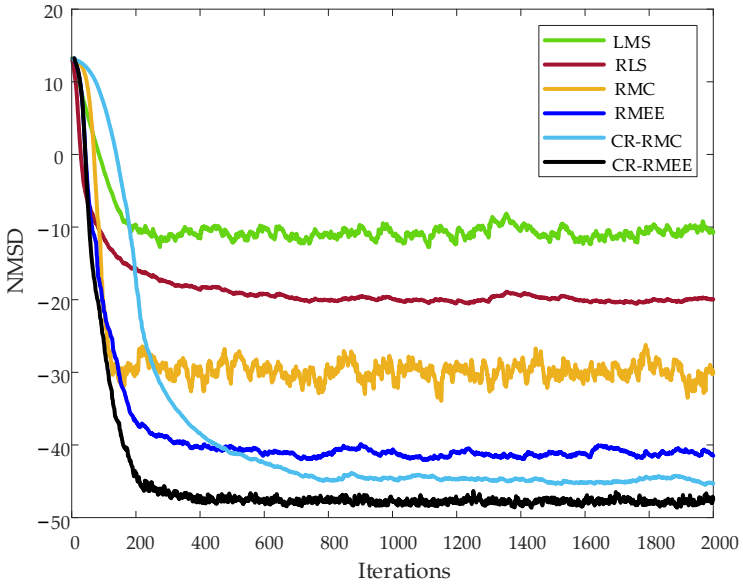


Figure 1. Transient NMSDs (dB) for different algorithms.

Secondly, to demonstrate the convergence and tracking performance of the CR-RMEE algorithm in a non-stationary sparse environment, we identify another three-stage sparse system. On the other hand, to compare the performance of each algorithm in different sparse environments, we assume the parameter vector of the unknown system as follows:

$$\mathbf{w}_o = \begin{cases} [0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0, \\ 0,\ 0,\ 0,\ 0,\ 0], n \leq 2000 \\ [1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1, \\ 0,\ 1,\ 0,\ 1,\ 0], 2000 < n \leq 4000 \\ [1, -1,\ 1, -1,\ 1, -1,\ 1, -1,\ 1, -1,\ 1, -1,\ 1, -1,\ 1, \\ -1,\ 1, -1,\ 1, -1], 4000 < n \end{cases} \tag{50}$$

From the above equation, it can be seen that as S increases from "1" to "10" and finally to "20", the sparsity of the system decreases over time. It is worth noting that due to the fact that the system is re-identified each time the algorithm detects a sudden change in the system, there will be a process of convergence again. And by changing the system parameters twice during the simulation, the subsequent convergence curves will also produce a three-step process. For LMS, RLS, RMS, and RMEE algorithms, the parameters

are kept the same as before, and for the CR-RMC and CR-RMEE algorithms, γ is set to 0.1. From Figure 2, it can be seen that the performance of each algorithm in each sparse phase follows the same trend as in Figure 1. Due to not considering the impact of sparse systems, the LMS algorithm, RLS algorithm, RMC algorithm, and RMEE algorithm are less affected by the sparsity of the environment. Similarly, both CR-RMEE and CR-RMC are influenced by sparsity. When the sparsity level is high, the CR-RMEE algorithm has a significant advantage compared to other algorithms. Although the advantage decreases slightly as the sparsity decreases, regardless of how sparse the system is, the normalized mean squared deviation (NMSD) of the CR-RMEE algorithm remains the lowest. The system undergo two parameter changes in total, and our proposed algorithm responded quickly and converged rapidly to a steady state, which indicates that the new algorithm also possesses excellent tracking performance. In summary, compared to the other algorithms, our algorithm exhibits the lowest steady-state error and excellent performance.

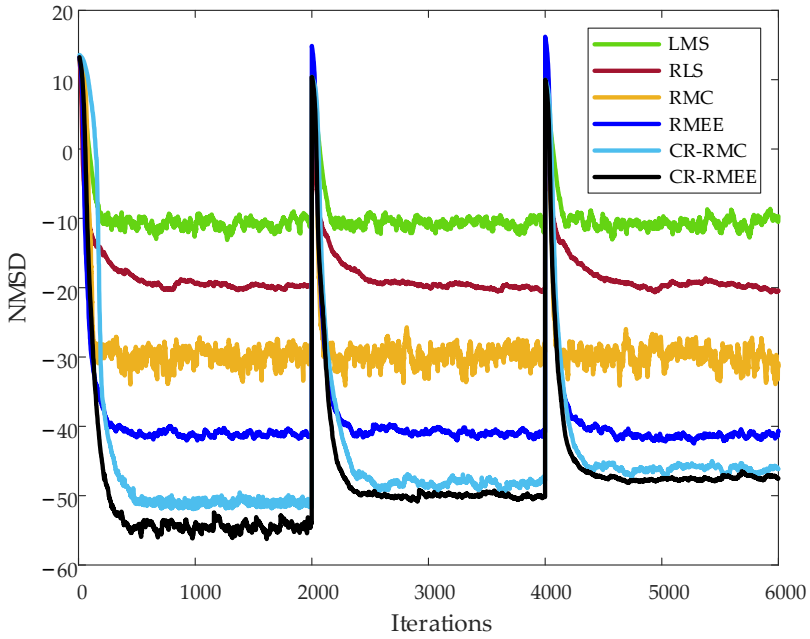


Figure 2. Transient NMSDs (dB) of different algorithms in a non-stationary environment.

Thirdly, we vary the parameter (σ, λ) for the CR-RMEE algorithm to test its impact on algorithm performance. All simulation results are the average of 200 independent runs, so we can only obtain an approximate value. The experimental results are shown in Table 1. To more intuitively illustrate the effect of different parameters on algorithm performance, we have selected a set of obvious results in graphical form, as shown in Figure 3. The parameters for CR-RMEE1, CR-RMEE2, CR-RMEE3, CR-RMEE4, and CR-RMEE5 are [$\sigma = 3$, $\lambda^2 = 0.942$], [$\sigma = 2$, $\lambda^2 = 0.970$], [$\sigma = 1$, $\lambda^2 = 0.980$], [$\sigma = 0.5$, $\lambda^2 = 0.992$], and [$\sigma = 0.2$, $\lambda^2 = 0.994$]. Combining Table 1 and Figure 3, it can be observed that the NMSD of the algorithm is negatively correlated with the σ value and positively correlated with the λ^2 value. As the steady-state error decreases, the convergence speed of the algorithm gradually slows down, and the level of post-convergence oscillation also gradually increases. This indicates that achieving a lower steady-state error often requires sacrificing the algorithm's convergence speed and stability. Therefore, in practical applications, it is necessary to adjust the parameters judiciously to achieve the best performance. Additionally, from Table 1, it can be seen that the steady-state error of the CR-RMEE algorithm in Figure 1 is approximately -49.6 dB when [$\sigma = 1$, $\lambda^2 = 0.996$].

Table 1. Transient NMSDs of CR-RMEE with different parameters after convergence.

	σ = 0.2	σ = 0.5	σ = 1	σ = 2	σ = 3
λ^2 = 0.942	−47.2	−38.2	−26.9	−14.5	−3.2
λ^2 = 0.970	−52.1	−44.1	−32.2	−20.8	−14.0
λ^2 = 0.980	−54.2	−46.9	−36.1	−23.8	−17.9
λ^2 = 0.992	−60.4	−54.8	−43.5	−31.1	−24.3
λ^2 = 0.994	−61.2	−56.6	−46.2	−33.4	−26.6
λ^2 = 0.996	−64.2	−60.3	−49.6	−36.7	−29.6

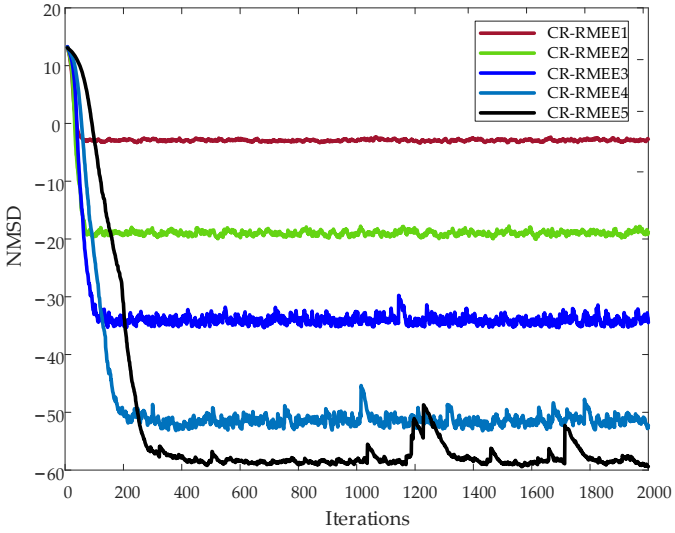


Figure 3. Transient NMSDs (dB) of CR-RMEE algorithm under different parameters (σ, λ).

Then, although the algorithms were tested separately in both steady and non-steady environments in the above experiments, demonstrating good tracking and convergence performance under certain conditions, the testing for algorithm robustness remains incomplete. Therefore, we continue to test the algorithm and comparative algorithms in four different noise environments. The four types of noise are Gaussian white noise with a distribution of $v(n) \sim N(0,0.25)$, pulse noise with a distribution of $v(n) \sim 0.95N(0,0.01) + 0.05N(0,1)$, pulse noise with a distribution of $v(n) \sim 0.95N(0,0.01) + 0.05N(0,9)$, and pulse noise with a distribution of $v(n) \sim 0.95N(0,0.01) + 0.05N(0,25)$, with an increasing intensity of pulse noise. These correspond, respectively, to Figure 4a–d. The selection of parameters for each algorithm aims to maintain optimal performance as much as possible. We set the unknown system as a sparse system, with the same choice of parameters for RMC algorithm, RMEE algorithm, CR-RMC algorithm, and the proposed CR-RMEE algorithm. The step size for the LMS algorithm is set to 0.01. The choices for λ are, respectively, $\lambda = 0.992$, $\lambda = 0.942$, $\lambda^2 = 0.996$, $\lambda^2 = 0.996$, and $\lambda^2 = 0.996$.

The results of testing under the four noise conditions can be seen from Figure 4. Firstly, as the intensity of noise pulses increases, the gap between different algorithms also becomes larger (convergence plots initially cluster together but then become dispersed). This indicates that different algorithms have varying degrees of sensitivity to noise, with RMC algorithm being the most affected. While classic algorithms such as LMS and RLS exhibit larger steady-state errors, they demonstrate strong stability. Among the algorithms tested under different noise conditions, the RMEE algorithm, CR-RMC algorithm, and our proposed CR-RMEE algorithm perform the best. They exhibit not only fast convergence rates in different environments but also low steady-state errors. It is noted that, among these three algorithms, our proposed new algorithm performs the best in all four scenarios, which strongly proves the robustness of the algorithm against noise.

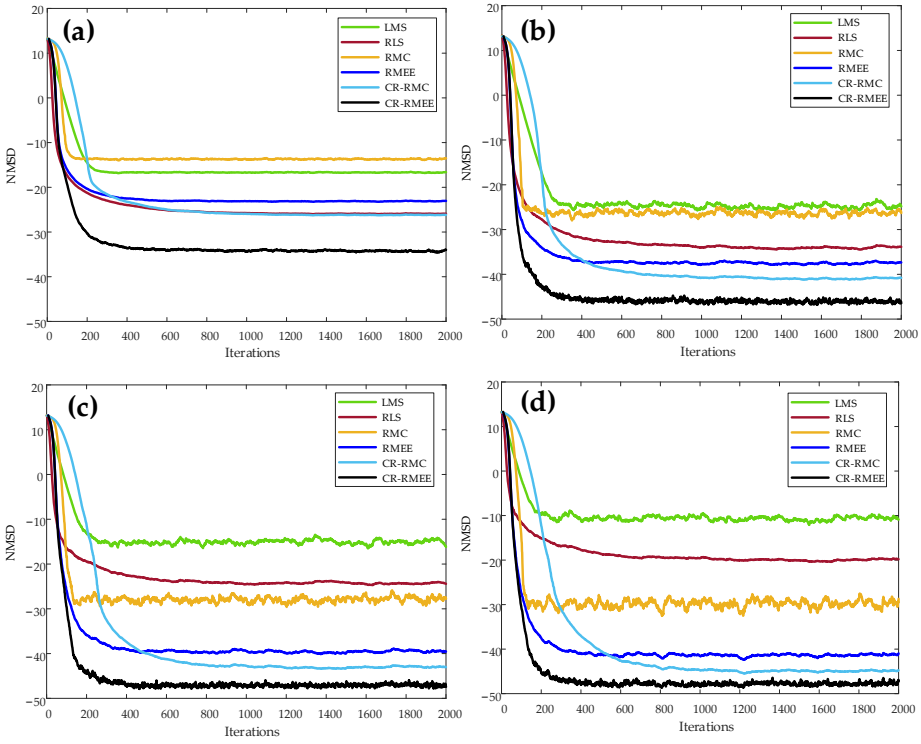


Figure 4. Transient NMSD (dB) of different algorithms in four different types of noise (From (**a**–**d**), the noise distributions are $v(n) \sim N(0,0.25)$, $v(n) \sim 0.95N(0,0.01) + 0.05N(0,1)$, $v(n) \sim 0.95N(0,0.01) + 0.05N(0,9)$, and $v(n) \sim 0.95N(0,0.01) + 0.05N(0,25)$).

Finally, to demonstrate the performance of the CR-RMEE algorithm in practical applications, we consider an echo cancellation application, whose basic structure is as shown in Figure 5. The generation principle of acoustic echo is mainly due to the sound from room A being transmitted to room B through channels such as wired or wireless, played through a speaker. After a series of acoustic reflections, the sound carries the inherent noise in room B and is picked up by the microphone, then transmitted back to room A. Acoustic echo can cause speakers in room A to hear their own speech shortly after speaking, significantly impacting the communication quality. In such cases, echo cancellation aims to eliminate these effects as much as possible. The mathematical model of echo cancellation is as follows: the far-end signal **u**(i) from room A, when transmitted to room B, generates the echo signal $r(i)$ after multiple reflections. Finally, the near-end signal, carrying the echo signal $r(i)$ and the noise $v(i)$ from room B, is combined to produce $d(i)$. The Acoustic Echo Canceller (AEC) module processes the far-end signal $\boldsymbol{u}(i)$ to obtain $y(i)$ (the echo signal). By taking the difference between d(i) and y(i), the error e(i) is obtained and passed to the adaptive filter (AEC module), leading to iterative updates of the filter coefficients. A smaller error indicates better echo cancellation effectiveness.

In this simulation, the input was changed to a real speech signal with a sampling rate of 8 kHz, as shown in Figure 6. The additive noise is impulsive with a mixed Gaussian distribution, $v(n) \sim 0.95N(0,0.01) + 0.05N(0,5)$. The parameters used for each algorithm were the same as in Figure 1. From Figure 7, it can be observed that, when the input is real speech, the convergence curves of various algorithms exhibit noticeable fluctuations, especially the LMS algorithm. Although the RMC and RMEE algorithms also have relatively small steady-state errors, the fluctuations are still evident. Our proposed CR-RMEE algorithm not only has smaller fluctuations but also the minimum steady-state error, indicating its superior performance.

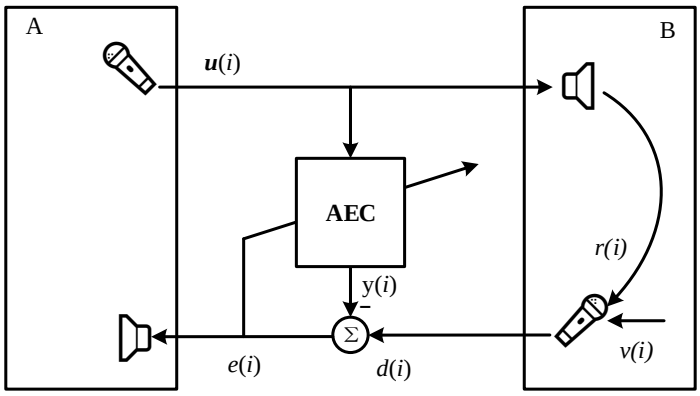


Figure 5. Principle diagram of echo cancellation.

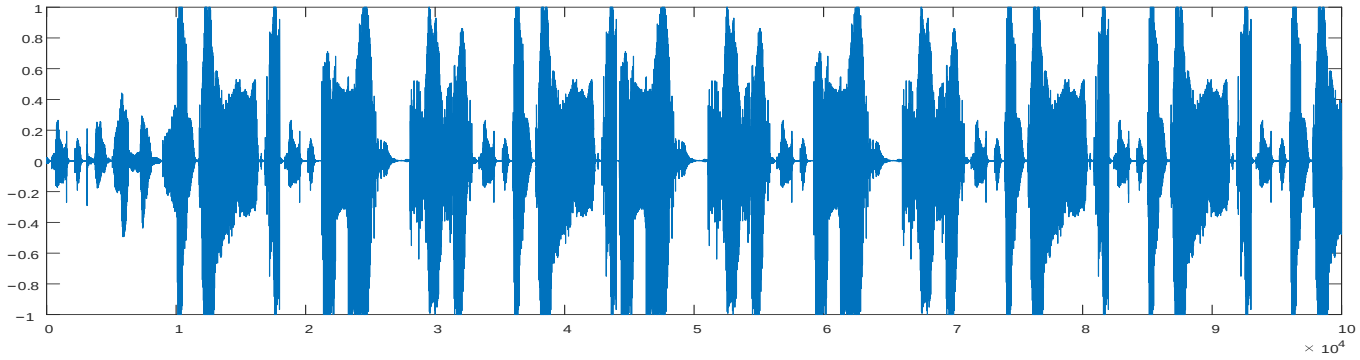


Figure 6. Speech signal.

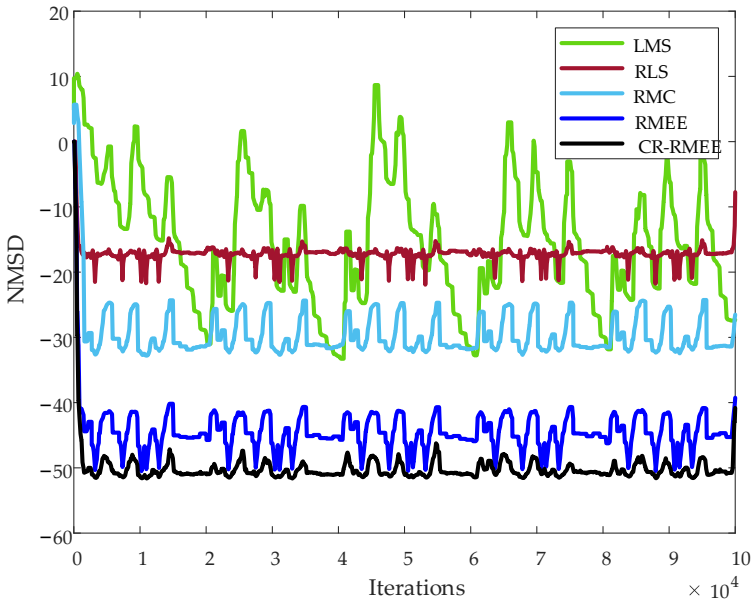


Figure 7. Transient NMSDs (dB) of different algorithms in Echo cancellation experiment.

5. Conclusions

This paper introduces a Convex Regularized Recursive Minimum Error Entropy algorithm (CR-RMEE) to address the challenges faced by traditional adaptive filtering algorithms in sparse environments, such as poor recognition capability and susceptibility to impulse noise. Building upon the RLS algorithm, the CR-RMEE algorithm incorporates a series of improvements tailored to pulse noise and sparse environments. By introducing the Minimum Error Entropy (MEE) criterion into the cost function, this algorithm enhances

resistance to impulse noise. Furthermore, regularization based on general convex functions is applied to the cost function, enabling the algorithm to leverage prior knowledge of sparsity, significantly improving its performance in sparse environments. Simulation results demonstrate that, in sparse system identification, the CR-RMEE algorithm outperforms the original RMEE algorithm. The new algorithm exhibits robustness in the presence of impulse noise and can adapt to system sparsity using prior knowledge. In future work, the application of the CR-RMEE algorithm may extend to complex-valued filtering and even nonlinear kernel filtering.

Author Contributions: Methodology, X.W. and S.O.; validation, Y.G. and X.W.; investigation, X.W. and S.O.; writing—original draft preparation, X.W., Y.G. and S.O.; writing—review and editing, X.W., S.O. and Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Shandong Provincial Natural Science Foundation under Grant ZR2022MF314.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank all reviewers for their helpful comments and suggestions on this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhao, H.; Gao, Y.; Zhu, Y. Robust Subband Adaptive Filter Algorithms-Based Mixture Correntropy and Application to Acoustic Echo Cancellation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2023**, *31*, 1223–1233. [CrossRef]
2. Pauline, S.H.; Samiappan, D.; Kumar, R.; Anand, A.; Kar, A. Variable Tap-Length Non-Parametric Variable Step-Size NLMS Adaptive Filtering Algorithm for Acoustic Echo Cancellation. *Appl. Acoust.* **2020**, *159*, 107074. [CrossRef]
3. Carbonelli, C.; Vedantam, S.; Mitra, U. Sparse Channel Estimation with Zero Tap Detection. *IEEE Trans. Wirel. Commun.* **2007**, *6*, 1743–1763. [CrossRef]
4. Gu, Y.; Jin, J.; Mei, S. L_0 Norm Constraint LMS Algorithm for Sparse System Identification. *IEEE Signal Process. Lett.* **2009**, *16*, 774–777. [CrossRef]
5. Chen, Y.; Gu, Y.; Hero, A.O. Sparse LMS for System Identification. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 3125–3128.
6. Eksioglu, E.M.; Tanc, A.K. RLS Algorithm with Convex Regularization. *IEEE Signal Process. Lett.* **2011**, *18*, 470–473. [CrossRef]
7. Chen, B.; Zhao, S.; Zhu, P.; Principe, J.C. Quantized Kernel Least Mean Square Algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 22–32. [CrossRef] [PubMed]
8. Chen, B.; Zhao, S.; Zhu, P.; Príncipe, J.C. Quantized Kernel Recursive Least Squares Algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 1484–1491. [CrossRef] [PubMed]
9. Chen, B.; Zhao, S.; Seth, S.; Principe, J.C. Online Efficient Learning with Quantized KLMS and L1 Regularization. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, 10–15 June 2012; pp. 1–6.
10. Wang, G.; Xue, R.; Wang, J. A Distributed Maximum Correntropy Kalman Filter. *Signal Process.* **2019**, *160*, 247–251. [CrossRef]
11. Heravi, A.R.; Hodtani, G.A. A New Information Theoretic Relation Between Minimum Error Entropy and Maximum Correntropy. *IEEE Signal Process. Lett.* **2018**, *25*, 921–925. [CrossRef]
12. Radmanesh, H.; Hajiabadi, M. Recursive Maximum Correntropy Learning Algorithm with Adaptive Kernel Size. *IEEE Trans. Circuits Syst. II Express Briefs* **2018**, *65*, 958–962. [CrossRef]
13. Wu, Z.; Shi, J.; Zhang, X.; Ma, W.; Chen, B. Kernel Recursive Maximum Correntropy. *Signal Process.* **2015**, *117*, 11–16. [CrossRef]
14. Chen, B.; Liu, X.; Zhao, H.; Principe, J.C. Maximum Correntropy Kalman Filter. *Automatica* **2017**, *76*, 70–77. [CrossRef]
15. Chen, B.; Zhu, Y.; Hu, J. Mean-Square Convergence Analysis of ADALINE Training with Minimum Error Entropy Criterion. *IEEE Trans. Neural Netw.* **2010**, *21*, 1168–1179. [CrossRef] [PubMed]
16. Shen, P.; Li, C. Minimum Total Error Entropy Method for Parameter Estimation. *IEEE Trans. Signal Process.* **2015**, *63*, 4079–4090. [CrossRef]
17. Chen, B.; Dang, L.; Gu, Y.; Zheng, N.; Príncipe, J.C. Minimum Error Entropy Kalman Filter. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 5819–5829. [CrossRef]
18. Li, Z.; Xing, L.; Chen, B. Adaptive Filtering with Quantized Minimum Error Entropy Criterion. *Signal Process.* **2020**, *172*, 107534. [CrossRef]
19. Ma, W.; Qu, H.; Gui, G.; Xu, L.; Zhao, J.; Chen, B. Maximum Correntropy Criterion Based Sparse Adaptive Filtering Algorithms for Robust Channel Estimation under Non-Gaussian Environments. *J. Frankl. Inst.* **2015**, *352*, 2708–2727. [CrossRef]

20. Wang, G.; Peng, B.; Feng, Z.; Yang, X.; Deng, J.; Wang, N. Adaptive Filtering Based on Recursive Minimum Error Entropy Criterion. *Signal Process.* **2021**, *179*, 107836. [CrossRef]
21. Zhang, X.; Li, K.; Wu, Z.; Fu, Y.; Zhao, H.; Chen, B. Convex Regularized Recursive Maximum Correntropy Algorithm. *Signal Process.* **2016**, *129*, 12–16. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.