Article

Low-Complexity Constrained Recursive Kernel Risk-Sensitive Loss Algorithm

Shunling Xiang *, Chunzhe Zhao, Zilin Gao and Dongfang Yan

School of Computer Science and Engineering, Chongqing Three Gorges University, Wanzhou, Chongqing 404120, China; zhaochunzhe@tsinghua.org.cn (C.Z.); zilingao@sanxiau.edu.cn (Z.G.); 20140013@sanxiau.edu.cn (D.Y.)
* Correspondence: 20160016@sanxiau.edu.cn

Abstract: The constrained recursive maximum correntropy criterion (CRMCC) combats the non-Gaussian noise effectively. However, the performance surface of maximum correntropy criterion (MCC) is highly non-convex, resulting in low accuracy. Inspired by the smooth kernel risk-sensitive loss (KRSL), a novel constrained recursive KRSL (CRKRSL) algorithm is proposed, which shows higher filtering accuracy and lower computational complexity than CRMCC. Meanwhile, a modified update strategy is developed to avoid the instability of CRKRSL in the early iterations. By using Isserlis's theorem to separate the complex symmetric matrix with fourth-moment variables, the mean square stability condition of CRKRSL is derived, and the simulation results validate its advantages.

Keywords: constrained adaptive filtering; kernel risk-sensitive loss; instability; mean square stability

1. Introduction

The constrained adaptive filters (CAFs) [1], where the weight is subject to linear constraints, have been widely studied in the field of adaptive signal processing. The original research of CAFs was derived from the antenna array processing, which employed the linearly-constrained minimum-variance (LCMV) criterion to estimation the direction of the antenna array [2]. CAFs have since been successfully applied to adaptive beamforming [3], system identification [4], channel equalization [5], and blind multiuser detection [6].

The simplest linearly-constrained adaptive filter, named constrained least mean-square (CLMS) [2], is developed from the LCMV criterion, and its mean square performance is analyzed based on a decomposable symmetric matrix in [7]. Due to stochastic gradient optimization, the CLMS has a simple structure with low computational complexity. However, its performance is highly influenced by the step size and correlated input. To improve the convergence speed, the constrained fast least-squares (CFLS) algorithm [8], linear-equality-constrained recursive least-squares (CRLS) algorithm [9] and its relaxed version are proposed at the expense of high computational complexity. Furthermore, the reduced-complexity constrained recursive least-squares algorithm based on the dichotomous coordinate descent (CRLS-DCD) iterations [10] and low-complexity constrained affine-projection (CAP) algorithm [11] with the data-selection method are proposed for reducing the computational complexity effectively. Other types of improved constrained filters [12–14] have been widely used, which make a trade-off between the computational complexity and filtering performance, i.e., convergence speed and filtering accuracy. All the mentioned constrained algorithms above are developed based on the mean square error (MSE) criterion [15] with good performances under Gaussian assumptions. However, with non-Gaussian cases, the filtering accuracy of these algorithms will decline sharply.

Therefore, the maximum correntropy criterion (MCC) [16,17], generalized MCC (GMCC) [18] and minimum error entropy (MEE) [19] criteria from information theoretic learning (ITL) [20] become alternative criteria, showing strong robustness to non-Gaussian signals. Adding the linear constraints on the weights of MCC and GMCC, the constrained

MCC (CMCC) [21] and constrained GMCC (CGMCC) [22] are developed by using stochastic gradient optimization. CMCC and CGMCC display good filtering performances in the presence of single-peak heavy-tailed noise. However, when coping with multi-peak noise, their performances will decline. Due to the symmetry of the errors, MEE counteracts the influence of multi-peak noise effectively. Adding the linear constraints into MEE, a gradient-based constrained MEE (CMEE) algorithm [23] with a sliding window is proposed at the expense of higher complexity but with better accuracy than CMCC and GMCC in the presence of multi-peak noise. Except for the ITL-based constrained filters, there also exist other criteria [24,25] to show good performance under a non-Gaussian environment. A constrained least mean M-estimation (CLMM) algorithm based on an improved M-estimation loss function has been proposed in [24]. Inspired by the boundedness of the gradient of the lncosh function, a constrained least lncosh adaptive filtering algorithm (CLLAF) has been developed in [25]. These constrained algorithms show good performances under different non-Gaussian noise. However, the gradient-based constrained adaptive filters need to make a trade-off between accuracy and convergence speed through an adjustable step size. Hence, the constrained recursive MCC (CRMCC) algorithm [26] is developed, which not only improves the accuracy but also accelerates the convergence.

Recently, an advanced ITL-based criterion, named kernel risk-sensitive loss (KRSL) [27], has been proposed by introducing a beneficial risk-sensitive parameter to regulate the shape of its performance surface. Compared with MCC, the KRSL is more "convex", which conduces better accuracy and faster convergence. Based on the KRSL, a gradient-based constrained mixture KRSL algorithm [28] is proposed, which indicates a higher filtering accuracy than CMCC.

In this paper, thanks to the advantage of KRSL criterion, a novel constrained recursive KRSL (CRKRSL) algorithm is proposed by using an average approximation method [29]. Thus, the proposed CRKRSL algorithm is converted to a variable step size gradient-based algorithm, which has a lower complexity than the traditional constrained recursive algorithms. Meanwhile, due to the instability of CRKRSL at the initial update stage, we use a gradient algorithm with a fixed step size to replace its initial update. Moreover, the convergence condition of mean square stability regarding iteration times is derived by decomposing a symmetric matrix with fourth-order variables. Simulation results indicate the advantages of CRKRSL on filtering accuracy and computational complexity.

The rest of the paper is organized as follows. The constrained KRSL loss and the CRKRSL algorithm are presented in Section 2. Stability analysis of CRKRSL is given in Section 3. Simulation results and discussion of CRKRSL are shown in Section 4. Finally, the conclusion is given in Section 5.

2. CRKRSL Algorithm

2.1. Notations

Throughout this paper, $\mathbb{R}$ denotes the real field, $\mathbb{R}^m$ denotes an m-dimensional real-valued vector space, $\mathbb{R}^{n \times m}$ denotes $n \times m$ matrix where its entries belong to $\mathbb{R}$; $(\cdot)^{\mathrm{T}}$ represents the transpose operation; $\mathbb{E}[\cdot]$ is the expectation operation; $\| \cdot \|$ denotes the Euclidean norm; and $O(\cdot)$ represents the computational complexity of algorithm.

2.2. KRSL Loss

As a nonlinear similarity measure of variable $\mathcal{X}, \mathcal{Y} \in \mathbb{R}$, the KRSL [27] is defined as

$$\begin{aligned} S(\mathcal{X}, \mathcal{Y}) &= \frac{1}{\gamma} \mathbb{E}\left[\exp\left(\frac{\gamma}{2} \|\varphi(\mathcal{X}) - \varphi(\mathcal{Y})\|^2 \right) \right] \\ &= \frac{1}{\gamma} \mathbb{E}\left[\exp\left(\gamma(1 - \kappa(\mathcal{X} - \mathcal{Y})) \right) \right], \end{aligned} \tag{1}$$

where $\gamma \in [0, +\infty)$ is a risk-sensitive parameter. $\varphi(\cdot)$ is the corresponding mapping variable in reproducing kernel Hilbert spaces (RKHS) [30], satisfying $\varphi^{\mathrm{T}}(\mathcal{X})\varphi(\mathcal{Y}) = \kappa(\mathcal{X} - \mathcal{Y})$ with Gaussian kernel

$$\kappa(\mathcal{X} - \mathcal{Y}) = \exp\left(-\frac{\|\mathcal{X} - \mathcal{Y}\|^2}{2\sigma^2}\right). \tag{2}$$

Generally, in terms of the unknown joint distribution between $\mathcal{X}$ and $\mathcal{Y}$, we adopt a finite sampled data pair $\{\tilde{x}_l, \tilde{y}_l\}_{l=1}^N$ to approximate the expected loss, which is given by

$$\hat{S}(\mathcal{X}, \mathcal{Y}) = \frac{1}{\gamma N}\sum_{l=1}^N \exp\left(\gamma(1 - \kappa(\tilde{x}_l - \tilde{y}_l))\right). \tag{3}$$

2.3. Constrained KRSL Loss

When applied to constrained adaptive filtering, the optimization problem with KRLS loss becomes

$$\min_{\mathbf{w}} \quad \frac{1}{\gamma}\sum_{l=1}^n \exp\left(\gamma(1 - \kappa(d_l - \mathbf{w}^{\mathrm{T}}\mathbf{u}_l))\right)$$
$$s.t. \quad \mathbf{C}^{\mathrm{T}}\mathbf{w} = \mathbf{f}, \tag{4}$$

where $\{\mathbf{u}_l, d_l\}_{l=1}^N \in \mathbb{R}^m \times \mathbb{R}$ are the input–output training data pair; $\mathbf{w} \in \mathbb{R}^m$ is the weight vector and $e_l = d_l - \mathbf{w}^{\mathrm{T}}\mathbf{u}_l$ is the corresponding error; $\mathbf{C} \in \mathbb{R}^{m \times q}$ and $\mathbf{f} \in \mathbb{R}^q$ are the constrained matrix and vector, respectively. By constructing the Lagrange function, the constrained problem is transformed to minimize the following constrained KRSL loss:

$$J_c(n) = \frac{\sigma^2}{\gamma}\sum_{l=1}^n \exp(\gamma(1 - \kappa(d_l - \mathbf{w}^{\mathrm{T}}\mathbf{u}_l))) + \boldsymbol{\theta}_n^{\mathrm{T}}(\mathbf{f} - \mathbf{C}^{\mathrm{T}}\mathbf{w}) \tag{5}$$

with $\boldsymbol{\theta}_n \in \mathbb{R}^q$ being the Lagrange multiplier.

2.4. Proposed CRKRSL Algorithm

Taking the gradient of $J_c(n)$ to zero at instant n, one has

$$\frac{\partial J_c(n)}{\partial \mathbf{w_n}} = \sum_{l=1}^n \phi(e_l)(d_l - \mathbf{w}_n^{\mathrm{T}}\mathbf{u}_l)\mathbf{u}_l + \mathbf{C}\boldsymbol{\theta}_n = \mathbf{0} \tag{6}$$

with $\phi(e_l) = \exp(\gamma(1 - \kappa(e_l)))\kappa(e_l)$.

Define

$$\mathbf{U}_n^{-1} = \sum_{l=1}^{n-1} \phi(e_l)\mathbf{u}_l\mathbf{u}_l^{\mathrm{T}} + \phi(e_n)\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}} \tag{7}$$

$$\mathbf{d}_n = \sum_{l=1}^{n-1} \phi(e_l)\mathbf{u}_l d_l + \phi(e_n)\mathbf{u}_n d_n. \tag{8}$$

Then, the constrained solution is derived as

$$\mathbf{w}_n = \mathbf{U}_n\mathbf{d}_n + \mathbf{U}_n\mathbf{C}\boldsymbol{\theta}_n \tag{9}$$

with $\boldsymbol{\theta}_n = (\mathbf{C}^{\mathrm{T}}\mathbf{U}_n\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^{\mathrm{T}}\mathbf{U}_n\mathbf{d}_n)$. By the matrix inversion lemma [31], Equation (7) is further rewritten by

$$\mathbf{U}_n = \mathbf{U}_{n-1} - \mathbf{g}_n\mathbf{u}_n^{\mathrm{T}}\mathbf{U}_{n-1}, \tag{10}$$

where the gain is given by

$$\mathbf{g}_n = \frac{\mathbf{U}_{n-1}\mathbf{u}_n}{\phi^{-1}(e_n) + \mathbf{u}_n^{\mathrm{T}}\mathbf{U}_{n-1}\mathbf{u}_n}. \tag{11}$$

Reorganizing Equation (11), we get another form of $\mathbf{g}_n$ by

$$\begin{aligned}
\mathbf{g}_n &= \phi(e_n)(\mathbf{U}_{n-1}\mathbf{u}_n - \mathbf{g}_n\mathbf{u}_n^{\mathrm{T}}\mathbf{U}_{n-1}\mathbf{u}_n) \\
&= \phi(e_n)\mathbf{U}_n\mathbf{u}_n.
\end{aligned} \tag{12}$$

To obtain a recursive solution, we expand Equation (9) as follow:

$$\begin{aligned}
\mathbf{w}_n &= \mathbf{U}_n(\mathbf{d}_{n-1} + \phi(e_n)\mathbf{u}_n d_n) + \mathbf{U}_n\mathbf{C}\boldsymbol{\theta}_n \\
&= (\mathbf{U}_{n-1} - \mathbf{g}_n\mathbf{u}_n^{\mathrm{T}}\mathbf{U}_{n-1})\mathbf{d}_{n-1} + \phi(e_n)\mathbf{U}_n\mathbf{u}_n d_n + \mathbf{U}_n\mathbf{C}\boldsymbol{\theta}_n \\
&= \mathbf{w}_{n-1} + \phi(e_n)\mathbf{U}_n\mathbf{u}_n d_n - \mathbf{g}_n\mathbf{u}_n^{\mathrm{T}}\mathbf{w}_{n-1} + \mathbf{U}_n\mathbf{C}\boldsymbol{\theta}_n \\
&= \mathbf{w}_{n-1} + \phi(e_n)e_n\mathbf{U}_n\mathbf{u}_n + \mathbf{U}_n\mathbf{C}\boldsymbol{\theta}_n,
\end{aligned} \tag{13}$$

where the corrected error e_n updated by the a priori weight $\mathbf{w}_{n-1}$ is obtained by

$$e_n = d_n - \mathbf{u}_n^{\mathrm{T}}\mathbf{w}_{n-1}. \tag{14}$$

Substituting Equation (12) into Equation (13), we get

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{g}_n e_n + \mathbf{U}_n\mathbf{C}\boldsymbol{\theta}_n \tag{15}$$

with

$$\boldsymbol{\theta}_n = (\mathbf{C}^{\mathrm{T}}\mathbf{U}_n\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^{\mathrm{T}}(\mathbf{w}_{n-1} + \mathbf{g}_n e_n)). \tag{16}$$

Therefore, combing Equations (10), (11), (14) and (15), we obtain the constrained recursive CRKRSL algorithm. The main drawback of Equation (15) is that the inverse matrix $(\mathbf{C}^{\mathrm{T}}\mathbf{U}_n\mathbf{C})^{-1}$ needs to be updated iteratively with complexity $O(q^3)$. To reduce the computational complexity of the CRKRSL algorithm, we consider the following linear model and give some assumptions. The linear model is described by

$$d_n = \mathbf{w}_*^{\mathrm{T}}\mathbf{u}_n + v_n, \tag{17}$$

where $\mathbf{w}_*$ is the model parameter and v_n is the noise at instant n. The assumptions are given as follows:

- A1 $\{\mathbf{u}_n\}$ is independent identically distributed (i.i.d), generated from a multivariate Gaussian distribution with covariance matrix $\mathbf{R} = \mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}}]$;
- A2 $\{v_n\}$ is zero mean, i.i.d, and independent with $\{\mathbf{u}_n\}$, satisfying $\delta^2 = \mathbb{E}[v_n^2]$;
- A3 the error e_n is uncorrelated with $\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}}$.

Inspired by the average approximation [29] and based on A1–A3, the correlation matrix $\mathbf{U}_n$ is approximated by

$$\begin{aligned}
\mathbf{U}_n &= \frac{1}{n}\left(\frac{1}{n}\sum_{l=1}^{n}\phi(e_l)\mathbf{u}_l\mathbf{u}_l^{\mathrm{T}}\right)^{-1} \\
&\approx \frac{1}{n}\mathbb{E}[\phi(e_n)]^{-1}\mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}}]^{-1} = \eta_n\mathbf{Z},
\end{aligned} \tag{18}$$

where $\mathbf{Z} = \mathbf{R}^{-1} = \mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}}]^{-1}$ and $\eta_n = (n\mathbb{E}[\phi(v_n)])^{-1}$ is approximated by the first-order Taylor expansion around the noise.

Furthermore, Equation (13) can be simplified as

$$\begin{aligned}
\mathbf{w}_n &= \mathbf{w}_{n-1} + \phi(e_n)e_n\mathbf{U}_n\mathbf{u}_n + \mathbf{U}_n\mathbf{C}\boldsymbol{\theta}_n \\
&= \mathbf{w}_{n-1} + \eta_n\phi(e_n)e_n\mathbf{Z}\mathbf{u}_n + \mathbf{Z}\mathbf{C}\hat{\boldsymbol{\theta}},
\end{aligned} \tag{19}$$

where $\hat{\boldsymbol{\theta}} = \boldsymbol{\Theta}(\mathbf{f} - \mathbf{C}^{\mathrm{T}}(\mathbf{w}_{n-1} + \eta_n \phi(e_n)e_n \mathbf{Z}\mathbf{u}_n))$ and the constrained inverse matrix is defined as $\boldsymbol{\Theta} = (\mathbf{C}^{\mathrm{T}}\mathbf{Z}\mathbf{C})^{-1}$.

Therefore, $\mathbf{w}_n$ is further expressed as

$$\mathbf{w}_n = \mathbf{Q}(\mathbf{w}_{n-1} + \eta_n \phi(e_n)e_n \mathbf{Z}\mathbf{u}_n) + \mathbf{p} \tag{20}$$

with $\mathbf{Q} = \mathbf{I} - \mathbf{Z}\mathbf{C}\boldsymbol{\Theta}\mathbf{C}^{\mathrm{T}}$ and $\mathbf{p} = \mathbf{Z}\mathbf{C}\boldsymbol{\Theta}\mathbf{f}$.

Based on the approximation, the recursive CRKRSL algorithm has been changed to a gradient one with variable step size η_n and transformed input $\mathbf{Z}\mathbf{u}_n$.

Remark 1. *The term $\phi(e_n)$ in Equation (20) has a significant impact on the stability of the CRKRSL algorithm under non-Gaussian noise since CRKRSL can suppress large outliers ($e_n \to \infty$) with a small $\phi(e_n)$. Figure 1 shows the relation between $\phi(e_n)$ and the error e_n. It is clear to see that $\phi(e_n)$ in CRKRSL is bigger than $\kappa(e_n)$ in CRMCC, when the error is small. (Note that $\phi(e_n) = \kappa(e_n)$ if $\gamma = 0$). Moreover, when $\gamma = 0$, the CRKRSL degenerates to an efficient CRMCC [26] algorithm and $\phi(e_n)$ reaches the maximum at $e_n = 0$. When $\gamma > 0$, $\phi(e_n)$ reaches the maximum at local points around $e_n = 0$ with a larger increment, resulting in a faster convergence speed and higher accuracy than CRMCC.*
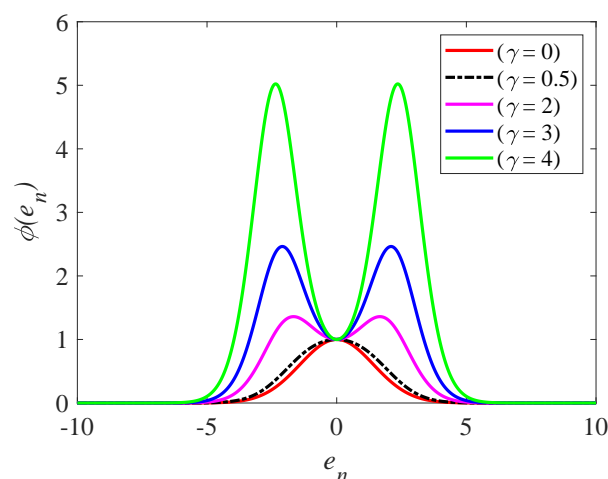

Figure 1. The relation between $\phi(e_n)$ and the error e_n ($\sigma = \sqrt{2}$).

However, Equation (20) is not stable at the initial update phase, since the variable step size η_n is large with a small instant n. Especially, when the kernel width of KRSL is small, the step size η_n even exceeds the convergence range, leading to the attenuation of filtering performance. To overcome this unfavorable factor, we introduce a gradient strategy with a fixed step size μ to replace the update of Equation (20) in the initial L iterations, which is described by

$$\mathbf{w}_n = \hat{\mathbf{Q}}(\mathbf{w}_{n-1} + \mu \phi(e_n)e_n \mathbf{u}_n) + \hat{\mathbf{p}} \tag{21}$$

with $\hat{\mathbf{Q}} = \mathbf{I} - \mathbf{C}(\mathbf{C}^{\mathrm{T}}\mathbf{C})^{-1}\mathbf{C}^{\mathrm{T}}$ and $\hat{\mathbf{p}} = \mathbf{C}(\mathbf{C}^{\mathrm{T}}\mathbf{C})^{-1}\mathbf{f}$.

Finally, the CRKRSL is summarized in Algorithm 1.

Algorithm 1: The CRKRSL Algorithm.

Input:

Data pair $\{\mathbf{u}_n, d_n\} \in \mathbb{R}^m \times \mathbb{R}$, $n = 1, 2, \cdots$.

Initialization:

Choose step-size μ; kernel width σ; risk-sensitive γ; initial iterative length L;
training size N_{tr};
initial weight $\mathbf{w}_0 = \mathbf{0}$.

  $e_n = d_n - \mathbf{w}_{n-1}^\mathrm{T}\mathbf{u}_n$

  for $n = 1 : L$

    $\mathbf{w}_n = \hat{\mathbf{Q}}(\mathbf{w}_{n-1} + \mu\phi(e_n)e_n\mathbf{u}_n) + \hat{\mathbf{p}}$

  end

  for $n = (L+1) : N_{tr}$

    $\mathbf{w}_n = \mathbf{Q}(\mathbf{w}_{n-1} + \eta_n\phi(e_n)e_n\mathbf{Z}\mathbf{u}_n) + \mathbf{p}$

  end

Remark 2. *In Equation* (20), *the constant inverse matrix* $\mathbf{\Theta} = (\mathbf{C}^\mathrm{T}\mathbf{Z}\mathbf{C})^{-1}$ *needs to be calculated only once before the update. On the contrary,* $(\mathbf{C}^\mathrm{T}\mathbf{U}_n\mathbf{C})^{-1}$ *in Equation* (15) *needs to be updated iteratively. Meanwhile, the update of matrix* $\mathbf{U}_n$ *by Equation* (10) *is avoided by using Equation* (20). *Therefore, the proposed CRKRSL with Equation* (20) *has a lower computational complexity than one with Equation* (15).

3. Stability Analysis

To obtain the stability condition of mean square weight error of CRKRSL, we define the weight error and model error as

$$\tilde{\mathbf{w}}_n = \mathbf{w}_n - \mathbf{w}_o \tag{22}$$

$$\mathbf{w}_\Delta = \mathbf{w}_* - \mathbf{w}_o \tag{23}$$

with the optimal weight

$$
\begin{aligned}
\mathbf{w}_o &= \lim_{n\to\infty}(\mathbf{U}_n\mathbf{d}_n + \mathbf{U}_n\mathbf{C}\boldsymbol{\theta}_n)\\
&= \lim_{n\to\infty}\mathbf{U}_n\mathbf{d}_n + \lim_{n\to\infty}\mathbf{U}_n\mathbf{C}(\mathbf{C}^\mathrm{T}\mathbf{U}_n\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^\mathrm{T}\mathbf{U}_n\mathbf{d}_n)\\
&= \lim_{n\to\infty}\Big[\frac{1}{n}\sum_{l=1}^{n}\phi(e_l)\mathbf{u}_l\mathbf{u}_l^\mathrm{T}\Big]^{-1}\Big[\frac{1}{n}\sum_{l=1}^{n}\phi(e_l)\mathbf{u}_l d_l\Big]\\
&\quad + \lim_{n\to\infty}\Big[\frac{1}{n}\sum_{l=1}^{n}\phi(e_l)\mathbf{u}_l\mathbf{u}_l^\mathrm{T}\Big]^{-1}\mathbf{C}\Big(\mathbf{C}^\mathrm{T}\Big[\frac{1}{n}\sum_{l=1}^{n}\phi(e_l)\mathbf{u}_l\mathbf{u}_l^\mathrm{T}\Big]^{-1}\mathbf{C}\Big)^{-1}(\mathbf{f} - \mathbf{C}^\mathrm{T}\mathbf{U}_n\mathbf{d}_n)\\
&= \mathbf{U}\mathbf{d} + \mathbf{U}\mathbf{C}(\mathbf{C}^\mathrm{T}\mathbf{U}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^\mathrm{T}\mathbf{U}\mathbf{d}),
\end{aligned}
\tag{24}
$$

where the robust correlation matrix and vector are defined as $\mathbf{U} = \lim_{n\to\infty}[\frac{1}{n}\sum_{l=1}^{n}\phi(e_l)\mathbf{u}_l\mathbf{u}_l^\mathrm{T}]^{-1} = \mathbb{E}[\phi(e_n)\mathbf{u}_n\mathbf{u}_n^\mathrm{T}]^{-1}$ and $\mathbf{d} = \lim_{n\to\infty}[\frac{1}{n}\sum_{l=1}^{n}\phi(e_l)\mathbf{u}_l d_l] = \mathbb{E}[\phi(e_n)\mathbf{u}_n d_n]$.

Subtracting $\mathbf{w}_o$ from both sides of Equation (20), we obtain

$$
\begin{aligned}
\tilde{\mathbf{w}}_n &= \mathbf{Q}(\mathbf{w}_{n-1} + \eta_n\phi(e_n)\mathbf{Z}\mathbf{u}_n d_n - \eta_n\phi(e_n)\mathbf{Z}\mathbf{u}_n\mathbf{u}_n^\mathrm{T}\mathbf{w}_{n-1}) + \mathbf{p} - \mathbf{w}_o\\
&= \mathbf{Q}(\mathbf{w}_{n-1} + \eta_n\phi(e_n)\mathbf{Z}\mathbf{u}_n(\mathbf{u}_n^\mathrm{T}\mathbf{w}_* + v_n) - \eta_n\phi(e_n)\mathbf{Z}\mathbf{u}_n\mathbf{u}_n^\mathrm{T}\mathbf{w}_{n-1}) + \mathbf{p} - \mathbf{w}_o\\
&= \mathbf{Q}(\mathbf{w}_{n-1} + \eta_n\phi(e_n)\mathbf{Z}\mathbf{u}_n(\mathbf{u}_n^\mathrm{T}(\mathbf{w}_\Delta + \mathbf{w}_o) + v_n) - \eta_n\phi(e_n)\mathbf{Z}\mathbf{u}_n\mathbf{u}_n^\mathrm{T}\mathbf{w}_{n-1}) + \mathbf{p} - \mathbf{w}_o\\
&= \mathbf{Q}(\mathbf{w}_{n-1} - \eta_n\phi(e_n)\mathbf{Z}\mathbf{u}_n\mathbf{u}_n^\mathrm{T}\tilde{\mathbf{w}}_{n-1}) + \eta_n\phi(e_n)\mathbf{Q}\mathbf{Z}\mathbf{u}_n(\mathbf{u}_n^\mathrm{T}\mathbf{w}_\Delta + v_n) + \mathbf{p} - \mathbf{w}_o\\
&= \mathbf{Q}(\mathbf{I} - \eta_n\phi(e_n)\mathbf{Z}\mathbf{u}_n\mathbf{u}_n^\mathrm{T})\tilde{\mathbf{w}}_{n-1} + \eta_n\phi(e_n)\mathbf{Q}\mathbf{Z}\mathbf{u}_n(\mathbf{u}_n^\mathrm{T}\mathbf{w}_\Delta + v_n) + \mathbf{Q}\mathbf{w}_o - \mathbf{w}_o + \mathbf{p}.
\end{aligned}
\tag{25}
$$

Since $\mathbf{Q}$ is an idempotent matrix, we have $\mathbf{Qw}_o - \mathbf{w}_o + \mathbf{p} = \mathbf{0}$ and $\mathbf{Q\tilde{w}}_n = \mathbf{\tilde{w}}_n$. Then, we get

$$
\begin{aligned}
\mathbf{\tilde{w}}_n &= \mathbf{Q}(\mathbf{I} - \eta_n \phi(e_n)\mathbf{Zu}_n\mathbf{u}_n^{\mathsf{T}})\mathbf{\tilde{w}}_{n-1} + \eta_n \phi(e_n)\mathbf{QZu}_n(\mathbf{u}_n^{\mathsf{T}}\mathbf{w}_\Delta + v_n) \\
&= (\mathbf{I} - \eta_n \phi(e_n)\mathbf{QZu}_n\mathbf{u}_n^{\mathsf{T}})\mathbf{\tilde{w}}_{n-1} + \eta_n \phi(e_n)\mathbf{QZu}_n(\mathbf{u}_n^{\mathsf{T}}\mathbf{w}_\Delta + v_n).
\end{aligned}
\tag{26}
$$

We take the expectation operator on both sides of the square norm, since the noise v_n is independent with $\mathbf{u}_n$ and the input sequence $\{\mathbf{u}_n\}$ is i.i.d under assumptions A1–A2. We further get the useful results that the a priori error weight $\mathbf{\tilde{w}}_{n-1}$ is independent with $\mathbf{u}_n$ and v_n based on the independence assumptions [32]. Hence, the cross terms are equal to zeros. Then, we obtain

$$
\begin{aligned}
\mathbb{E}[\|\mathbf{\tilde{w}}_n\|^2] &= \mathbb{E}[\|\mathbf{\tilde{w}}_{n-1}\|_{\mathbf{F}_n}^2] + \eta_n^2 \delta^2 \mathbb{E}[\phi^2(e_n)]\mathbb{E}[\mathbf{u}_n^{\mathsf{T}}\mathbf{ZQZu}_n] \\
&\quad + \eta_n^2 \mathbb{E}[\phi^2(e_n)]\mathbf{w}_\Delta^{\mathsf{T}}\mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathsf{T}}\mathbf{ZQZu}_n\mathbf{u}_n^{\mathsf{T}}]\mathbf{w}_\Delta
\end{aligned}
\tag{27}
$$

with

$$
\begin{aligned}
\mathbf{F}_n &= \mathbb{E}[(\mathbf{I} - \eta_n \phi(e_n)\mathbf{QZu}_n\mathbf{u}_n^{\mathsf{T}})^{\mathsf{T}}(\mathbf{I} - \eta_n \phi(e_n)\mathbf{QZu}_n\mathbf{u}_n^{\mathsf{T}})] \\
&= \mathbf{I} - 2\eta_n \mathbb{E}[\phi(e_n)]\mathbb{E}[\mathbf{QZu}_n\mathbf{u}_n^{\mathsf{T}}] + \eta_n^2 \mathbb{E}[\phi^2(e_n)]\mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathsf{T}}\mathbf{ZQZu}_n\mathbf{u}_n^{\mathsf{T}}].
\end{aligned}
\tag{28}
$$

According to Isserlis's theorem [33], the symmetric matrix with fourth-moment Gaussian variables can be separated by

$$
\begin{aligned}
\mathbf{V} &= \mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathsf{T}}\mathbf{ZQZu}_n\mathbf{u}_n^{\mathsf{T}}] \\
&= \mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathsf{T}}\mathbf{Z}]\mathbb{E}[\mathbf{QZu}_n\mathbf{u}_n^{\mathsf{T}}] + \mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathsf{T}}\mathbf{ZQ}]\mathbb{E}[\mathbf{Zu}_n\mathbf{u}_n^{\mathsf{T}}] + \mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathsf{T}}]\mathbb{E}[\mathbf{u}_n^{\mathsf{T}}\mathbf{ZQZu}_n] \\
&= 2\mathbf{Q} + \mathrm{tr}\{\mathbf{QZ}\}\mathbf{R},
\end{aligned}
\tag{29}
$$

where $\mathrm{tr}\{\cdot\}$ denotes the trace operator and $\mathbf{R} = \mathbf{Z}^{-1} = \mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathsf{T}}]$ is a positive define correlation matrix.

Therefore, Equation (27) can be simplified as

$$
\begin{aligned}
\mathbb{E}[\|\mathbf{\tilde{w}}_n\|^2] &= \mathbb{E}[\|\mathbf{\tilde{w}}_{n-1}\|_{\mathbf{F}_n}^2] + \eta_n^2 \delta^2 \mathbb{E}[\phi^2(e_n)]\mathrm{tr}\{\mathbf{QZ}\} + \eta_n^2 \mathbb{E}[\phi^2(e_n)]\mathbf{w}_\Delta^{\mathsf{T}}(2\mathbf{Q} + \mathrm{tr}\{\mathbf{QZ}\}\mathbf{R})\mathbf{w}_\Delta \\
&= \mathbb{E}[\|\mathbf{\tilde{w}}_{n-1}\|_{\mathbf{F}_n}^2] + \eta_n^2 \mathbb{E}[\phi^2(e_n)]\mathrm{tr}\{\mathbf{QZ}\}(\mathbf{w}_\Delta^{\mathsf{T}}\mathbf{Rw}_\Delta + \delta^2)
\end{aligned}
\tag{30}
$$

with a simplified

$$
\begin{aligned}
\mathbf{F}_n &= \mathbf{I} - 2\eta_n \mathbb{E}[\phi(e_n)]\mathbf{Q} + \eta_n^2 \mathbb{E}[\phi^2(e_n)]\mathbf{V} \\
&= \mathbf{I} - 2\eta_n \mathbb{E}[\phi(e_n)]\mathbf{Q} + \eta_n^2 \mathbb{E}[\phi^2(e_n)](2\mathbf{Q} + \mathrm{tr}\{\mathbf{QZ}\}\mathbf{R}).
\end{aligned}
\tag{31}
$$

Define q_k and r_k, $k \in \{1, 2, \cdots, m\}$, being the kth eigenvalue of matrix $\mathbf{Q}$ and $\mathbf{R}$. To ensure the square stability, the eigenvalues of $\mathbf{F}_n$ should satisfy the following condition:

$$
\begin{aligned}
&|1 - 2\eta_n \mathbb{E}[\phi(e_n)]q_k + \eta_n^2 \mathbb{E}[\phi^2(e_n)](2q_k + \mathrm{tr}\{\mathbf{QZ}\}r_k)| < 1, \\
&\text{for} \quad k = 1, 2, \cdots, m.
\end{aligned}
\tag{32}
$$

Then, the convergence condition about the step size can be expressed as

$$
\eta_n < \min_i \left(\frac{2\mathbb{E}[\phi(e_n)]q_k}{\mathbb{E}[\phi^2(e_n)](2q_k + \mathrm{tr}\{\mathbf{QZ}\}r_k)} \right).
\tag{33}
$$

Since the step size $\eta_n = (n\mathbb{E}[\phi(v_n)])^{-1}$ is related to the iteration n, finally, we obtain the weight square error stability condition about the iteration n by

$$
n > \max_i \left(\frac{\mathbb{E}[\phi^2(e_n)](2q_k + \mathrm{tr}\{\mathbf{QZ}\}r_k)}{2\mathbb{E}[\phi(v_n)]\mathbb{E}[\phi(e_n)]q_k} \right),
\tag{34}
$$

where the nonlinear terms $\mathbb{E}[\phi(e_n)]$ and $\mathbb{E}[\phi^2(e_n)]$ can be approximated by the Taylor expansion.

Remark 3. *Inequality (34) implies that the iteration n should be sufficiently large to guarantee the convergence. When n is small, CRKRSL with Equation (20) cannot satisfy (34), resulting in fluctuations at the initial update stage. Therefore, it is reasonable to use Equation (21) to replace Equation (20) to improve the convergence speed and filtering accuracy.*

4. Results and Discussion

In this section, we will show the advantages of the CRKRSL algorithm on the filtering accuracy and computational complexity for both low-dimensional and high-dimensional inputs. The noise model, data selection and algorithms comparison are given as follows, respectively.

Noise model: We first consider a pure Gaussian noise to test the filtering accuracy, i.e., Gaussian noise with $v_n \sim \mathrm{N}(0, \delta^2)$, where $\mathrm{N}(\bar{\mu}, \bar{\delta}^2)$ denotes the Gaussian distribution with mean $\bar{\mu}$ and variance $\bar{\delta}^2$. Then, a mixed noise is considered to test the robustness of CRKRSL. The mixed noise model under a probability process is denoted as

$$v_n = b(n)v_1(n) + (1 - b(n))v_2(n), \tag{35}$$

where $b(n) \in \{0, 1\}$ satisfies the binary distribution with probability $P\{b(n) = 0\} = 0.1$ and $P\{b(n) = 1\} = 0.9$. $v_1(n)$ with high probability generates the ordinary noise by a Gaussian distribution $\mathrm{N}(0, \delta_1^2)$ and $v_2(n)$ with low probability generates a few impulsive noise. Two types of $v_2(n)$ are considered: (a) Gaussian noise with large variance, i.e., $v_2(n) \sim \mathrm{N}(0, 100)$; (b) α-stable noise [34] with parameter function $\mathcal{F}(0.8, 0, 0.1, 0)$.

Data selection: The training inputs sampled from a Gaussian distribution with zero mean and variance matrix $\mathbf{R}$ and 5000 samples are chosen for the simulation. The parameters $\mathbf{C}$, $\mathbf{f}$, $\mathbf{R}$ are configured as the same as in [7]. Note that there exist two sets of data with different input dimensions in [7]. The underlying dimension is either $m = 7$ or $m = 31$. The simulated mean square deviation (MSD) is define as $\mathrm{MSD(dB)} = 10\log_{10}(\|\tilde{\mathbf{w}}_n\|^2)$ and the steady-state MSD is defined as the mean of last 1000 samples. The obtained results are averaged over 500 Monte Carlo trials. The results were run on MATLAB version R2020b on a Windows 10 operating system, configured with an Intel(R) Core(TM) i7-8700 CPU 3.20 GHz and RAM 16 GB.

Compared algorithms: The constrained algorithms, including the CLMS [7], CMCC [21], CLLAF [25] CRLS [9], and CRMCC [26], are chosen to be compared with the proposed CRKRSL. For fair comparison, the kernel widths σ of CMCC, CRMCC, and CRKRSL are set by the same value; the regularization terms of CRLS and CRMCC are set as 0.001; the initial iterative length of CRKRSL is set as the same the input dimension, i.e., $L = m$.

4.1. Low-Dimensional Input

In this part, the input dimension m and constrained dimension q are set as $m = 7$ and $q = 3$, respectively. The noise considered here satisfies the following conditions, i.e., Gaussian noise $v(n) \sim \mathrm{N}(0, 0.1)$ and mixed noise $v_1(n) \sim \mathrm{N}(0, 0.1)$. To reflect the influence of risk-sensitive parameter γ on the MSD, the relations between γ and steady-state MSD are shown in Figures 2 and 3 under different noise models. From Figure 2, one can see that CRKRSL has a stable steady-state MSD under Gaussian noise. Therefore, γ has little influence on the performance of CRKRSL. From Figure 3, it is observed that the CRKRSL is sensitive to γ under mixed noise with model (a) and achieves the lowest steady-state MSD around $\gamma = 2.5$. Therefore, we choose $\gamma = 2.5$ for the following algorithm comparison in Figures 4 and 5. Note that ρ denotes the shape parameter in the CLLAF algorithm. All necessary parameters are given in the figures.
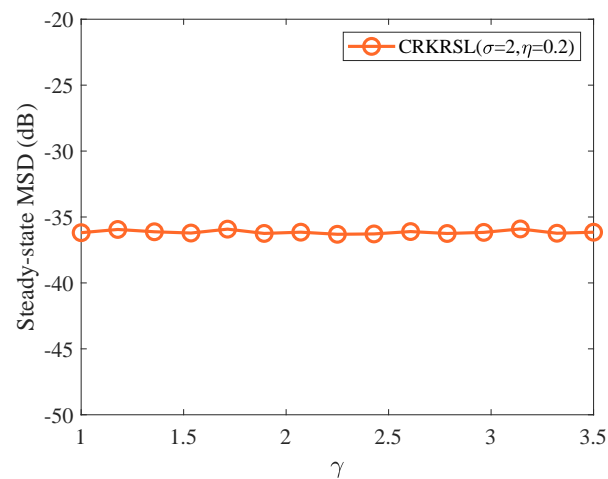
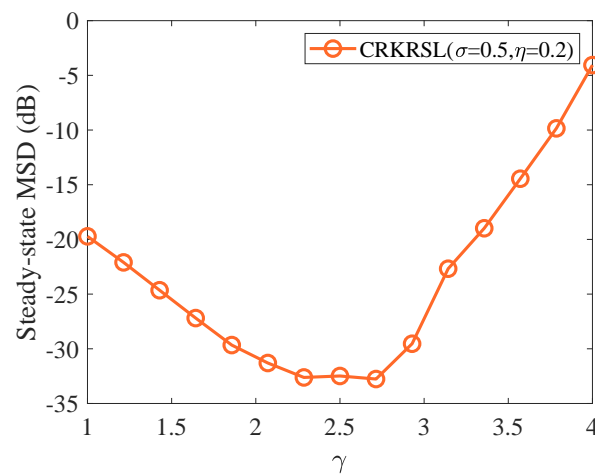**Figure 2.** Steady-state MSD versus risk-sensitive parameter γ under Gaussian noise ($m = 7, q = 3$).

Figure 3. Steady-state MSD versus risk-sensitive parameter γ under mixed noise with model (a) ($m = 7, q = 3$).
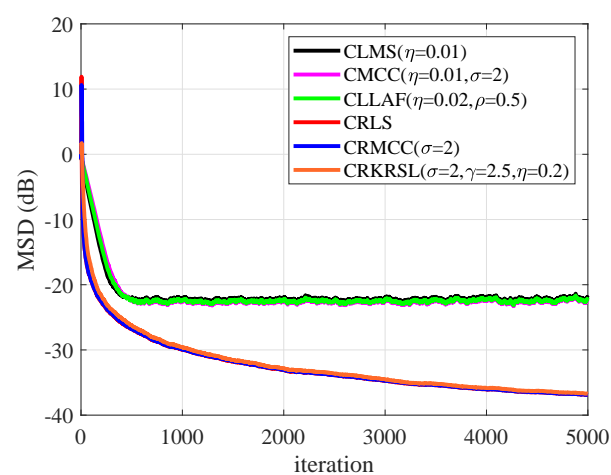

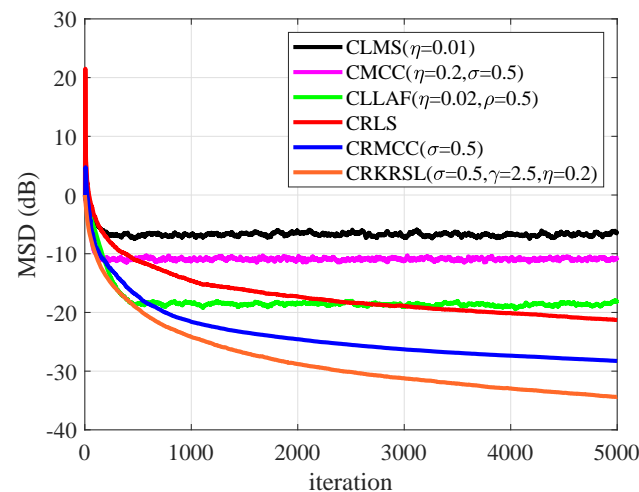**Figure 4.** MSDs of compared algorithms under Gaussian noise ($m = 7, q = 3$).

Figure 5. MSDs of compared algorithms under mixed noise with model (a) ($m = 7, q = 3$).

From Figure 4, one can see that CRKRSL, CRMCC and CRLS coincide and have almost the same MSDs under Gaussian noise. This implies that the CRKRSL deals with the Gaussian noise well by choosing a large kernel width. From Figure 5, it is observed that CRKRSL has the best performance among all constrained algorithms since the risk-sensitive parameter γ can avoid fluctuations caused by a small kernel width. Moreover, CRKRSL is more stable than CRMCC at the initial stage, potentially leading to a better filtering performance. In Table 1, we further compare the consumed time at each iteration and steady-state MSDs of each algorithm under mixed noise with model (a). The consumed time of CRKRSL is far less than that of CRMCC and CRLS.

Table 1. Consumed time and steady-state MSD of CLMS, CMCC, CLLAF, CRLS, CRMCC and CRKRSL under mixed noise with model (a) based on low-dimensional input.

Algorithm	Consumed Time (s)	Steady-State MSD (dB)
CLMS	0.0024	-6.70
CMCC	0.0028	-10.86
CLLAF	0.0025	-18.58
CRLS	0.0324	-21.23
CRMCC	0.0327	-27.75
CRKRSL	0.0038	-34.34

To show the advantage of CRKRSL on computational complexity, Table 2 lists the compared results of all mentioned algorithms at each iteration. One can see that CRKRSL has a lower computational complexity than CRLS and CRMCC by avoiding calculating the inverse matrix. Although the inverse matrix $\mathbf{U}_n$ of CRMCC is not required to be calculated, the inverse matrix $(\mathbf{C}^{\mathrm{T}}\mathbf{U}_n\mathbf{C})^{-1}$ is still needed to be calculated at each iteration, resulting in a high computational complexity with a large q.

To test the performance of CRKRSL under mixed noise with model (b), Figure 6 gives the MSD results of all mentioned algorithms. One can see from Figure 6 that CRKRSL, CRMCC, CMCC and CLLAF show strong robustness to outliers, and CRKRSL has the lowest MSD, whereas the CLMS and CRLS are not stable due to being sensitive to α-stable noise.

Table 2. Computational complexity of CLMS, CMCC, CLLAF, CRLS, CRMCC and CRKRSL at each iteration.

Algorithms	Computational Complexity
CLMS	$O(m^2)$
CMCC	$O(m^2)$
CLLAF	$O(m^2)$
CRLS	$O(m^3 + q^3)$
CRMCC	$O(m^2 + q^3)$
CRKRSL	$O(m^2)$
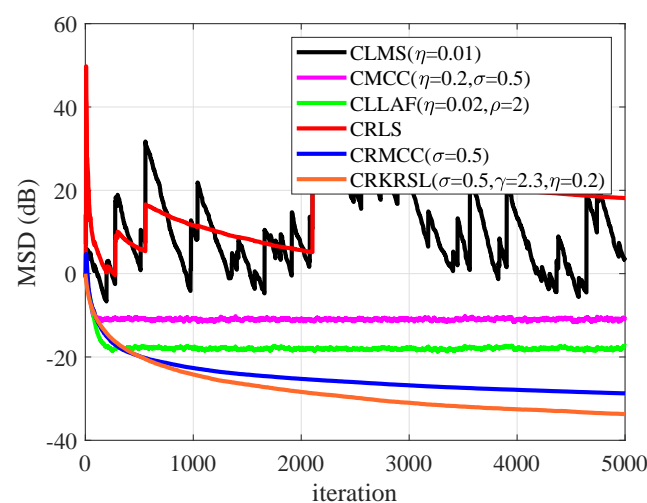
Figure 6. MSDs of compared algorithms under mixed noise with model (b) ($m = 7, q = 3$).

4.2. High-Dimensional Input

In this part, the input dimension m and constrained dimension q are set as $m = 31$ and $q = 1$, respectively. We only consider the mixed noise, since the Gaussian noise influences the performance of CRKRSL little by selecting a large kernel width. The mixed noise satisfies $v_1(n) \sim N(0,1)$.

Figures 7 and 8 show the MSDs of different algorithms under mixed noise with model (a) and model (b), respectively. It is clear to see that CRKRSL shows the best performance out of all the compared algorithms both under mixed noise with model (a) and model (b). The initial iterative length L influences the convergence speed, significantly. Therefore, the initial iterative length should be not smaller than the input dimension. In Table 3, we further compare the consumed time at each iteration and steady-state MSDs of each algorithm under mixed noise with model (a). One can see that the consumed time of CRKRSL is far less than that of CRMCC. Moreover, CRKRSL has the lowest steady-state MSD value.
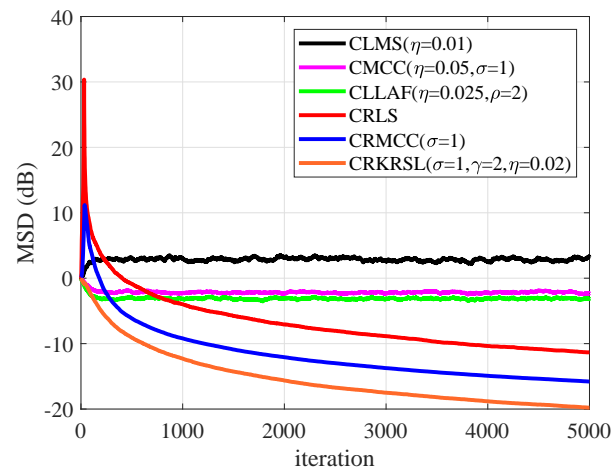
Figure 7. MSDs of compared algorithms under mixed noise with model (a) ($m = 31, q = 1$).
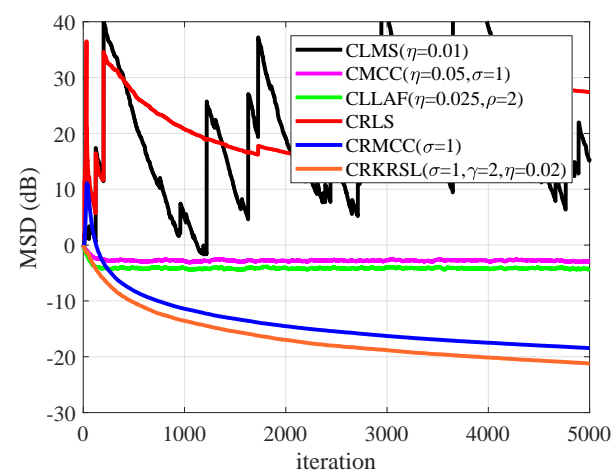

Figure 8. MSDs of compared algorithms under mixed noise with model (b) ($m = 31, q = 1$).

Table 3. Consumed time and steady-state MSD of CLMS, CMCC, CLLAF, CRLS, CRMCC and CRKRSL under mixed noise with model (a) based on high-dimensional input.

Algorithm	Consumed Time (s)	Steady-State MSD (dB)
CLMS	0.0035	2.98
CMCC	0.0034	−2.21
CLLAF	0.0034	−3.13
CRLS	0.0112	−11.02
CRMCC	0.0114	−15.45
CRKRSL	0.0063	−19.72

5. Conclusions

By introducing the linear constraints to the kernel risk-sensitive loss (KRSL), a low-complexity constrained recursive KRSL (CRKRSL) algorithm is presented with the help of average approximation. Since the risk-sensitive parameter is able to control the smoothness of the performance surface, CRKRSL achieves higher accuracy than some existing constrained recursive algorithms. Due to the inaccuracy of the average approximation with a few inputs, a fixed step-size gradient method is adopted to avoid the instability of CRKRSL at the initial update stage. Moreover, mean square analysis indicates that iteration times influence the stability of CRKRSL significantly and simulation results confirm the advantages of CRKRSL. The effectiveness of CRKRSL highly relies on the average approximation method, which has limits when coping with nonstationary signals. In the

future, we focus on finding a novel approximation method to process both stationary and nonstationary signals and further improve the computational efficiency and accuracy of constrained recursive algorithms.

References

1. De Campos, M.L.R.; Werner, S.; Apolinário, J.A. Constrained adaptive filters. In *Adaptive Antenna Arrays: Trends and Applications*; Springer: New York, NY, USA, 2004; pp. 46–64.
2. Frost, O. An algorithm for linearly constrained adaptive array processing. *Proc. IEEE* **1972**, *60*, 926–935. [CrossRef]
3. Li, J.; Stoica, P. *Robust Adaptive Beamforming*; John Wiley & Sons: New York, NY, USA, 2005.
4. Diniz, P.S. *Adaptive Filtering: Algorithms and Practical Implementation*; Springer: Cham, Switzerland, 1997.
5. Wu, Q.; Li, Y.; Xue, W. A kernel recursive maximum versoria-like criterion algorithm for nonlinear channel equalization. *Symmetry* **2019**, *11*, 1067. [CrossRef]
6. Verdu, S. *Multiuser Detection*; Cambridge University Press: Cambridge, UK, 1998.
7. Arablouei, R.; Doğançay, K.; Werner, S. On the mean-square performance of the constrained LMS algorithm. *Signal Process.* **2015**, *117*, 192–197. [CrossRef]
8. Resende, L.S.; Romano, J.M.T.; Bellanger, M.G. A fast least-squares algorithm for linearly constrained adaptive filtering. *IEEE Trans. Signal Process.* **1996**, *44*, 1168–1174. [CrossRef]
9. Arablouei, R.; Doğançay, K. Performance analysis of linear-equality-constrained least-squares estimation. *IEEE Trans. Signal Process.* **2015**, *63*, 3762–3769. [CrossRef]
10. Arablouei, R.; Doğançay, K. Reduced-complexity constrained recursive least-squares adaptive filtering algorithm. *IEEE Trans. Signal Process.* **2012**, *60*, 6687–6692. [CrossRef]
11. Werner, S.; Apolinário, J.A.; de Campos, M.L.R.; Diniz, P. Low-complexity constrained affine-projection algorithms. *IEEE Trans. Signal Process.* **2005**, *53*, 4545–4555. [CrossRef]
12. Apolinário, J.A., Jr.; de Campos, M.L.R.; Bernal, O.C.P. The constrained conjugate gradient algorithm. *IEEE Signal Process. Lett.* **2000**, *12*, 351–354. [CrossRef]
13. De Campos, M.L.R.; Werner, S.; Apolinário, J.A. Constrained adaptation algorithms employing Householder transformation. *IEEE Trans. Signal Process.* **2002**, *9*, 2187–2195. [CrossRef]
14. Arablouei, R.; Doğançay, K. Linearly-constrained recursive total least-squares algorithm. *IEEE Signal Process. Lett.* **2012**, *12*, 821–824. [CrossRef]
15. Haykin, S. *Adaptive Filter Theory*; Prentice-Hall: Upper Saddle River, NJ, USA, 2002.
16. Liu, W.; Pokharel, P.P.; Príncipe, J.C. Correntropy: Properties and applications in non-Gaussian signal processing. *IEEE Trans. Signal Process.* **2007**, *55*, 5286–5298. [CrossRef]
17. Li, Y.; Wang, Y.; Sun, L. A proportionate normalized maximum correntropy criterion algorithm with correntropy induced metric constraint for identifying sparse systems. *Symmetry* **2018**, *10*, 683. [CrossRef]
18. Chen, B.; Xing, L.; Zhao, H.; Zheng, N.; Príncipe, J.C. Generalized correntropy for robust adaptive filtering. *IEEE Trans. Signal Process.* **2016**, *64*, 3376–3387. [CrossRef]
19. Erdogmus, D.; Príncipe, J.C. An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems. *IEEE Trans. Signal Process.* **2002**, *50*, 1780–1786. [CrossRef]
20. Príncipe, J.C. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*; Springer: New York, NY, USA, 2010.
21. Peng, S.; Chen, B.; Sun, L.; Ser, W.; Lin, Z. Constrained maximum correntropy adaptive filtering. *Signal Process.* **2017**, *140*, 116–126. [CrossRef]
22. Bhattacharjee, S.S.; Shaikh, M.A.; Kumar, K.; George, N.V. Robust constrained generalized correntropy and maximum versoria criterion adaptive filters. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 3002–3006. [CrossRef]

23. Peng, S.; Ser, W.; Chen, B.; Sun, L.; Lin, Z. Robust constrained adaptive filtering under minimum error entropy criterion. *IEEE Trans. Circuits Syst. II Express Briefs* **2018**, *65*, 1119–1123. [CrossRef]
24. Wang, Z.; Zhao, H.; Zeng, X. Constrained least mean M-estimation adaptive filtering algorithm. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 1507–1511. [CrossRef]
25. Liang, T.; Li, Y.; Zakharov, Y.V.; Xue, W.; Qi, J. Constrained least lncosh adaptive filtering algorithm. *Signal Process.* **2021**, *183*, 108044. [CrossRef]
26. Qian, G.; Ning, X.; Wang, S. Recursive constrained maximum correntropy criterion algorithm for adaptive filtering. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 2229–2233 [CrossRef]
27. Chen, B.; Xing, L.; Xu, B.; Zhao, H.; Zheng, N.; Príncipe, J.C. Kernel risk-sensitive loss: Definition, properties and application to robust adaptive filtering. *IEEE Trans. Signal Process.* **2017**, *65*, 2888–2901. [CrossRef]
28. Qian, G.; Dong, F.; Wang, S. Robust constrained minimum mixture kernel risk-sensitive loss algorithm for adaptive filtering. *Digit. Signal Process.* **2020**, *107*, 102859. [CrossRef]
29. Qian, G.; Wang, S.; Wang, L.; Duan, S. Convergence analysis of a fixed point algorithm under maximum complex correntropy criterion. *IEEE Signal Process. Lett.* **2018**, *25*, 1830–1834. [CrossRef]
30. Liu, W.; Príncipe, J.C.; Haykin, S. *Kernel Adaptive Filtering: A Comprehensive Introduction*; Wiley: Hoboken, NJ, USA, 2011.
31. Zhang, X.-D. *Matrix Analysis and Applications*; Cambridge University Press: Cambridge, UK, 2017.
32. Sayed, A.H. *Adaptive Filters*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
33. Isserlis, L. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika* **1918**, *12*, 134–139. [CrossRef]
34. Zhang, Q.; Feng, W.; Iu, H.H.C.; Wang, S. Adaptive filters with robust augmented space linear model: A weighted *k*-NN method. *IEEE Trans. Signal Process.* **2021**, *69*, 6448–6461. [CrossRef]