Article

A Kernel Recursive Maximum Versoria-Like Criterion Algorithm for Nonlinear Channel Equalization

Qishuai Wu ¹, Yingsong Li ^{1,2,*} and Wei Xue ¹

¹ College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

² Key Laboratory of Microwave Remote Sensing, National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China

* Correspondence: liyingsong@ieee.org; Tel.: +86-451-8251-9808

Received: 1 August 2019; Accepted: 20 August 2019; Published: 21 August 2019

Abstract: In this paper, a kernel recursive maximum Versoria-like criterion (KRMVLC) algorithm has been constructed, derived, and analyzed within the framework of nonlinear adaptive filtering (AF), which considers the benefits of logarithmic second-order errors and the symmetry maximum-Versoria criterion (MVC) lying in reproducing the kernel Hilbert space (RKHS). In the devised KRMVLC, the Versoria approach aims to resist the impulse noise. The proposed KRMVLC algorithm was carefully derived for taking the nonlinear channel equalization (NCE) under different non-Gaussian interferences. The achieved results verify that the KRMVLC is robust against non-Gaussian interferences and performs better than those of the popular kernel AF algorithms, like the kernel least-mean-square (KLMS), kernel least-mixed-mean-square (KLMMN), and Kernel maximum Versoria criterion (KMVC).

Keywords: kernel recursive maximum Versoria-like criterion; maximum Versoria criterion; nonlinear channel equalization

1. Introduction

Nonlinear adaptive filters, like the Kernel methods [1,2], are widely used in adaptive filtering (AF) to promote nonlinear implementations in the field of signal processing [3,4], and plenty of linear filters have been recasted in high-dimensional space, such as the well-known reproducing-kernel-Hilbert space (RKHS), to give robust nonlinear extension [1]. In recent years, a great number of classical kernel AFs (KAF) have been proposed for non-linear filters [1,2,5], including the kernel recursive least-square (KRLS) [6] and kernel least mean-square (KLMS) [2]. Variations of the early reported kernel algorithms have also been expanded to improve the behaviors of classical KAFs [7–13]. However, KLMS and KRLS algorithms perform poorly under non-Gaussian interferences, since they only consider the second-order error signal statistics [14,15].

Finding a cost function of an AF algorithm which can capture beyond second-order error statistics makes it very useful and powerful for AF under various non-Gaussian noise interferences [16–30]. The least-mean with p -order error for realizing a powerful method called the LMP algorithm [16,17] uses a high-ordered error to create a cost function to form new AF, which has better behavior than the least mean square (LMS) algorithm for non-Gaussian interferences. However, sometimes it still shows slow convergence, like the least mean-fourth (LMF) for the parameter $p = 4$, since its convergence characteristics might be sensitive to the adaptive weights that are in proximity to the optimal solution from the wiener method [18]. To solve the above problems, a kind of KAF algorithm has been developed by using two different norms to construct a cost function, such as the kernel least mean-mixed-norm (KLMMN) algorithm [19–22]. Comparing with the KLMS and kernel LMF (KLMF), the KLMMN

algorithm combines the advantages of these algorithms, which have fast speed with respect to the speed of convergence, and reduces the estimated bias with the help of the mean square error (MSE) evaluation. Then, the kernel methods have been introduced into the maximum Versoria criterion (MVC) [23], as well as the recursive generalized mixed-norm (RGMN) to create the kerneled MVC (KMVC) [24] and kerneled RGMN (KRGMN) [22] for the nonlinear filter under non-Gaussian noise interferences.

The main purpose of this work was to develop a KAF algorithm for dealing with the equalization of nonlinear channels that is also denoted nonlinear channel equalization (NCE). Logarithmic second-order errors (LSOE) and the symmetry maximum Versoria-criterion (MVC) have been considered to form a desired novel cost function, while the gradient optimization has been adopted to seek for its solution to develop a more powerful KAF, called the kernel recursive maximum Versoria-like criterion (KRMVLC) algorithm. The KRMVLC algorithm has been derived in the field of the KAF, while its evaluated performance has been investigated and discussed for implementing NCE. The achieved numerical results demonstrate that the KRMVLC achieves better behaviors compared with the KLME, KLMS, KMVC, KLMMN, KRGMN, and KRLS algorithms when using the convergence and MSE criteria to discuss the evaluation performance over NCE.

The main contributions of the proposed KRMVLC are the following three aspects: (1) it uses the LSOE and MVC to design a new function; (2) it has been derived in the field of the KAF; and (3) it achieves the best performance. The rest of this paper is constructed as follows: In Section 2, the proposed KRMVLC algorithm is introduced briefly. In Section 3, the performance of the KRMVLC algorithm is illustrated using the simulation experiments. Finally, in Section 4, the conclusion is given.

2. The KRMVLC Algorithm

The well-known kernel methods gave rise to the idea of converting the input data to a higher dimensional featured space via the nonlinear-mapping method under $\varphi: \mathbf{U} \rightarrow \mathbf{F}$. Herein, $\mathbf{U}$ denotes the input space, and $\mathbf{F}$ is implemented in RKHS. The training data is written as $\{\mathbf{u}(i), d(i)\}_{i=1}^N$, where the training data has N elements, $d(i)$ stands for the signal we expected, and $\mathbf{u}(i)$ is the input of the considered system. In the kernel methods, the input $\mathbf{u}(i)$ is transformed as $\varphi(\mathbf{u}(i))$ in RKHS. According to Mercer's theorem [1], the general kernel function is expressed as:

$$\begin{aligned}\kappa(\mathbf{a}, \mathbf{a}') &= \varphi^T(\mathbf{a}) \varphi(\mathbf{a}') \\ \kappa(\mathbf{a}, \mathbf{a}') &= \exp\left(-\frac{\|\mathbf{a} - \mathbf{a}'\|}{\sigma^2}\right)\end{aligned}\quad (1)$$

Here, σ denotes the kernel width that is used in the kernel function given in Equation (1). In this paper, the KRMVLC algorithm is derived, analyzed, and discussed in the field of the kernel and mixture AF algorithms. An exponential weighting method is used herein for finding out the solution of the new cost function that constructs the KRMVLC method since it can resist the impulse noise, and this new cost function is:

$$\begin{aligned}J_{\text{KR}}(\Omega) &= \sum_{j=1}^i \gamma^{i-j} \left\{ \frac{\omega}{2} \log \left(1 + \frac{|d(j) - \Omega^T \varphi(j)|^2}{2\xi^2} \right) \right. \\ &\quad \left. + \frac{(1-\omega)}{2} E \left[\frac{1}{1 + \tau |d(j) - \Omega^T \varphi(j)|^2} \right] \right\} + \frac{1}{2} \gamma^i \lambda \|\Omega\|^2.\end{aligned}\quad (2)$$

The expectation operator is written as $E[\cdot]$. τ is the Versoria-shaped parameter, and ξ is a constant, while the forgetting factor γ is used for gradually strengthening the weights. ω and λ are the mixed and regularization factors, respectively. Herein, the regularization term is implemented by using a norm constraint to ensure the existence of the inverse autocorrelation matrix. The gradient operation of Equation (2) is

$$\frac{\partial J_{KR}(\boldsymbol{\Omega})}{\partial \boldsymbol{\Omega}} = - \sum_{j=1}^i \gamma^{i-j} \boldsymbol{\varphi}(j) z(j) d(j) + \left(\sum_{j=1}^i \gamma^{i-j} \boldsymbol{\varphi}(j) z(j) \boldsymbol{\varphi}^T(j) + \gamma^i \lambda \right) \boldsymbol{\Omega}, \quad (3)$$

where $z(j)$ is written as

$$z(j) = \left\{ \frac{\omega}{|d(j) - \boldsymbol{\Omega}^T \boldsymbol{\varphi}(j)|^2 + 2\zeta^2} + \frac{(1-\omega)\tau}{\left(1 + \tau|d(j) - \boldsymbol{\Omega}^T \boldsymbol{\varphi}(j)|^2\right)^2} \right\}. \quad (4)$$

Let Equation (3) be zero, and consider $\boldsymbol{\Omega}$

$$\boldsymbol{\Omega} = \left(\sum_{j=1}^i \gamma^{i-j} \boldsymbol{\varphi}(j) z(j) \boldsymbol{\varphi}^T(j) + \gamma^i \lambda \right)^{-1} \sum_{j=1}^i \gamma^{i-j} \boldsymbol{\varphi}(j) z(j) d(j). \quad (5)$$

By considering

$$\mathbf{d}(i) = [d(1), d(2), d(3), \dots, d(i-1), d(i)]^T, \quad (6)$$

$$\boldsymbol{\Phi}(i) = [\boldsymbol{\varphi}(1), \boldsymbol{\varphi}(2), \boldsymbol{\varphi}(3), \dots, \boldsymbol{\varphi}(i-1), \boldsymbol{\varphi}(i)], \quad (7)$$

$$\begin{aligned} \boldsymbol{\Psi}(i) = \text{diag} & \left[\gamma^{i-1} \left\{ \frac{\omega}{|d(1) - \boldsymbol{\Omega}^T \boldsymbol{\varphi}(1)|^2 + 2\zeta^2} + \frac{(1-\omega)\tau}{\left(1 + \tau|d(1) - \boldsymbol{\Omega}^T \boldsymbol{\varphi}(1)|^2\right)^2} \right\}, \right. \\ & \gamma^{i-2} \left\{ \frac{\omega}{|d(2) - \boldsymbol{\Omega}^T \boldsymbol{\varphi}(2)|^2 + 2\zeta^2} + \frac{(1-\omega)\tau}{\left(1 + \tau|d(2) - \boldsymbol{\Omega}^T \boldsymbol{\varphi}(2)|^2\right)^2} \right\}, \dots, \\ & \left. \left\{ \frac{\omega}{|d(i) - \boldsymbol{\Omega}^T \boldsymbol{\varphi}(i)|^2 + 2\zeta^2} + \frac{(1-\omega)\tau}{\left(1 + \tau|d(i) - \boldsymbol{\Omega}^T \boldsymbol{\varphi}(i)|^2\right)^2} \right\} \right]. \end{aligned} \quad (8)$$

The matrix form of Equation (5) is

$$\boldsymbol{\Omega}(i) = \left(\boldsymbol{\Phi}(i) \boldsymbol{\Psi}(i) \boldsymbol{\Phi}^T(i) + \gamma^i \lambda \mathbf{I} \right)^{-1} \boldsymbol{\Phi}(i) \boldsymbol{\Psi}(i) \mathbf{d}(i). \quad (9)$$

The matrix inverse lemma is introduced into Equation (9), which is

$$(\mathbf{B} + \mathbf{DEF})^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1} \mathbf{D} (\mathbf{E}^{-1} + \mathbf{FB}^{-1} \mathbf{D})^{-1} \mathbf{FB}^{-1}, \quad (10)$$

and considering

$$\gamma^i \lambda \mathbf{I} \rightarrow \mathbf{B}, \boldsymbol{\Phi}(i) \rightarrow \mathbf{D}, \boldsymbol{\Psi}(i) \rightarrow \mathbf{E}, \boldsymbol{\Phi}^T(i) \rightarrow \mathbf{F}, \quad (11)$$

then, Equation (9) is modified as

$$\left(\boldsymbol{\Phi}(i) \boldsymbol{\Psi}(i) \boldsymbol{\Phi}^T(i) + \gamma^i \lambda \mathbf{I} \right)^{-1} \boldsymbol{\Phi}(i) \boldsymbol{\Psi}(i) = \boldsymbol{\Phi}(i) \left(\boldsymbol{\Phi}^T(i) \boldsymbol{\Phi}(i) + \gamma^i \lambda \boldsymbol{\Psi}^{-1}(i) \right)^{-1}. \quad (12)$$

$\boldsymbol{\Omega}(i)$ can also be considered as

$$\boldsymbol{\Omega}(i) = \boldsymbol{\Phi}(i) \left(\boldsymbol{\Phi}^T(i) \boldsymbol{\Phi}(i) + \gamma^i \lambda \boldsymbol{\Psi}^{-1}(i) \right)^{-1} \mathbf{d}(i), \quad (13)$$

Then, $\Omega(i)$ is analyzed by considering the input data linear combination

$$\Omega(i) = \Phi(i) \mathbf{a}(i) \quad (14)$$

with $\mathbf{a}(i)$ of

$$\mathbf{a}(i) = \left(\Phi^T(i) \Phi(i) + \gamma^i \lambda \Psi^{-1}(i) \right)^{-1} \mathbf{d}(i). \quad (15)$$

Defining $\mathbf{Q}(i)$ as

$$\mathbf{Q}(i) = \left(\Phi^T(i) \Phi(i) + \gamma^i \lambda \Psi^{-1}(i) \right)^{-1}, \quad (16)$$

where $\Phi(i) = \left\{ \Phi(i-1), \varphi(i) \right\}$, and then, $\mathbf{Q}(i)$ is obtained

$$\mathbf{Q}(i) = \begin{bmatrix} \Phi^T(i-1) \Phi(i-1) + \gamma^i \lambda \Psi^{-1}(i-1) & \Phi^T(i-1) \varphi(i) \\ \varphi^T(i) \Phi(i-1) & \left\{ \frac{\omega}{|d(i) - \Omega^T \varphi(i)|^2 + 2\zeta^2} + \frac{(1-\omega)\tau}{(1 + \tau|d(i) - \Omega^T \varphi(i)|^2)^2} \right\}^{-1} \\ & \times \gamma^i \lambda + \varphi^T(i) \varphi(i) \end{bmatrix}^{-1}. \quad (17)$$

Then, defining $c(i)$

$$c(i) = \left\{ \frac{\omega}{|d(i) - \Omega^T \varphi(i)|^2 + 2\zeta^2} + \frac{(1-\omega)\tau}{(1 + \tau|d(i) - \Omega^T \varphi(i)|^2)^2} \right\}^{-1} \quad (18)$$

Then, $\mathbf{Q}(i)$ is modified to be

$$\mathbf{Q}(i) = \begin{bmatrix} \Phi^T(i-1) \Phi(i-1) + \gamma^i \lambda \Psi^{-1}(i-1) & \Phi^T(i-1) \varphi(i) \\ \varphi^T(i) \Phi(i-1) & \varphi^T(i) \varphi(i) + \gamma^i \lambda c(i) \end{bmatrix}^{-1}. \quad (19)$$

Based on the derivation above, we have

$$\mathbf{Q}^{-1}(i) = \begin{bmatrix} \mathbf{Q}^{-1}(i-1) & \mathbf{b}(i) \\ \mathbf{b}^T(i) & \varphi^T(i) \varphi(i) + \gamma^i \lambda c(i) \end{bmatrix}, \quad (20)$$

where $\mathbf{b}(i) = \Phi(i-1)^T \varphi(i)$, while the block-matrix inversion operation is presented as

$$\begin{bmatrix} \mathbf{B} & \mathbf{D} \\ \mathbf{E} & \mathbf{F} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{B} - \mathbf{D}\mathbf{F}^{-1}\mathbf{E})^{-1} & -\mathbf{B}^{-1}\mathbf{D}(\mathbf{F} - \mathbf{E}\mathbf{B}^{-1}\mathbf{D})^{-1} \\ -\mathbf{F}^{-1}\mathbf{E}(\mathbf{B} - \mathbf{D}\mathbf{F}^{-1}\mathbf{E})^{-1} & (\mathbf{F} - \mathbf{E}\mathbf{B}^{-1}\mathbf{D})^{-1} \end{bmatrix}. \quad (21)$$

Utilizing the block matrix inversion operation, Equation (19) can be rewritten as

$$\mathbf{Q}(i) = \varepsilon^{-1}(i) \begin{bmatrix} \mathbf{Q}(i-1) \varepsilon(i) + \mathbf{f}(i) \mathbf{f}^T(i) & -\mathbf{f}(i) \\ -\mathbf{f}^T(i) & 1 \end{bmatrix}, \quad (22)$$

where $\mathbf{f}(i) = \mathbf{Q}(i-1) \mathbf{b}(i)$, and $\varepsilon(i) = \gamma^i \lambda c(i) + \boldsymbol{\varphi}^T(i) \boldsymbol{\varphi}(i) - \mathbf{f}^T(i) \mathbf{b}(i)$. Therefore, we can get $\mathbf{a}(i)$ given by

$$\begin{aligned} \mathbf{a}(i) &= \mathbf{Q}(i) \mathbf{d}(i) \\ &= \begin{bmatrix} \mathbf{Q}(i-1) + \mathbf{f}(i) \mathbf{f}^T(i) \varepsilon^{-1}(i) & -\mathbf{f}(i) \varepsilon^{-1}(i) \\ -\mathbf{f}^T(i) \varepsilon^{-1}(i) & \varepsilon^{-1}(i) \end{bmatrix} \begin{bmatrix} \mathbf{d}(i-1) \\ d(i) \end{bmatrix}, \\ &= \begin{bmatrix} \mathbf{a}(i-1) - \mathbf{f}(i) \varepsilon^{-1}(i) e(i) \\ \varepsilon^{-1}(i) e(i) \end{bmatrix}. \end{aligned} \quad (23)$$

In Equation (23), $e(i) = d(i) - \mathbf{b}^T(i) \mathbf{a}(i-1)$. At last, the KRMVLC is devised and presented in Algorithm 1 in the form of a pseudo-code.

Algorithm 1 KRMVLC.

Require: $\tau, \xi, \gamma, \omega, \lambda$

Ensure: $\mathbf{a}(i)$

```

1:  $\mathbf{Q}(1) = (\gamma\lambda + \kappa(\mathbf{u}(1), \mathbf{u}(1)))^{-1}$ ,
2:  $\mathbf{a}(1) = \mathbf{Q}(1) d(1)$ ,
3: for  $i > 1$  do
4:    $\mathbf{b}(i) = [\kappa(\mathbf{u}(1), \mathbf{u}(1)), \dots, \kappa(\mathbf{u}(i), \mathbf{u}(i-1))]^T$ 
5:    $\mathbf{f}(i) = \mathbf{Q}(i-1) \mathbf{b}(i)$ 
6:    $c(i) = \left\{ \frac{\omega}{|d(i) - \Omega^T \boldsymbol{\varphi}(i)|^2 + 2\xi^2} + \frac{(1-\omega)\tau}{(1+\tau|d(i) - \Omega^T \boldsymbol{\varphi}(i)|^2)^2} \right\}^{-1}$ 
7:    $\varepsilon(i) = \gamma^i \lambda c(i) + \boldsymbol{\varphi}^T(i) \boldsymbol{\varphi}(i) - \mathbf{f}^T(i) \mathbf{b}(i)$ 
8:    $\mathbf{Q}(i) = \varepsilon(i)^{-1} \begin{bmatrix} \mathbf{Q}(i-1) \varepsilon(i) + \mathbf{f}(i) \mathbf{f}^T(i) - \mathbf{f}(i) \\ -\mathbf{f}^T(i) & 1 \end{bmatrix}$ 
9:    $e(i) = d(i) - \mathbf{b}^T(i) \mathbf{a}(i-1) = d(i) - \Omega^T \boldsymbol{\varphi}(i)$ 
10:   $\mathbf{a}(i) = \begin{bmatrix} \mathbf{a}(i-1) - \mathbf{f}(i) \varepsilon^{-1}(i) e(i) \\ \varepsilon^{-1}(i) e(i) \end{bmatrix}$ 
11: end for
```

3. Simulation Result

In this section, the NCE is considered to discuss the behavior of the devised KRMVLC under different noise-interference environments. The nonlinear channel equalization is presented in Figure 1. In the channel equalization system, a binary signal $s(i)$ is input to the nonlinear channel, and an output signal $r(i)$ to which noise has been added is obtained at the output of the nonlinear channel, and the signal affected by channel distortion is used as the input signal of the nonlinear adaptive filter. The purpose of channel equalization is to construct an “inverse” filter to reproduce the original input signal with the lowest possible error probability, so when the MSE reaches the minimum value, the nonlinear adaptive filter can be used to represent the inverse model of a nonlinear channel. The basic nonlinear channel model is presented in Figure 2, and the description of this model is given: $\{s(1), s(2), \dots, s(L)\}$ where the binary value is considered as the input of the utilized nonlinear channel model. At the receiving side, the received signal is the presence of the additive noise $n(i)$, while the observation is $\{r(1), r(2), r(3), \dots, r(L-1), r(L)\}$. If it is regarded as a simple regression

problem, the samples can be written as $\{([r(i), \dots, r(i+l)], s(i-D))\}$. In this paper, l was the embedded time-length, while D was considered as the equilibrium lag time. $D = 2$ and $l = 3$ are used in the following experiments. The input of the nonlinear channel is given by $x(i) = s(i) + 0.5s(i-1)$, while the output of this channel is expressed as $r(i) = x(i) - 0.9x(i)^2 + n(i)$, and $n(i)$ denotes the mixed noise using $n_1(i)$ and $n_2(i)$ [31]. The devised KRMVLC was investigated using Monte-Carlo simulations, and its MSE was compared with KLME, KLMS, KRGMN, KLMMN, KMVC, and KRLS. Three different mixed noises were chosen to analyze the performance of the KRMVLC algorithm.

(1) Noise $n_1(i)$ with the Bernoulli distribution, a power of 0.5, and noise $n_2(i)$ with Gaussian distribution and power of 0.5 are utilized in the constructed Experiment-1.

(2) Noise $n_1(i)$ with the Laplace distribution, a power of 0.5, and noise $n_2(i)$ with Gaussian distribution and power of 0.5 are considered in the constructed Experiment-2.

(3) Noise $n_1(i)$ with a uniform distribution, power of 0.5, and noise $n_2(i)$ with Gaussian distribution and power of 0.5 are utilized in the constructed Experiment-3.

The achieved results with 50 independent Monte-Carlo runs are presented to discuss the proposed KRMVLC. In these simulations, the noise power is 0.1, and some parameters are listed in Table 1, where τ was set to be 0.01, and ζ was set to be 0.01, where the parameters were settled to achieve almost the same initial convergence. The convergence of the algorithms is presented in Figure 3. These figures show that the KRMVLC converges the fastest and has lower MSE, which is because the MVC can resist impulse noises. Then, only the Laplace distribution noise was considered to discuss the performance of the KRMVLC, given in Figure 2. Herein, the power of the Laplace distribution noise is 0.5 in Experiment-4. The KRMVLC algorithm still converges faster and has the lowest MSE compared to the others with respect to algorithms. Thus, the proposed KRMVLC is amazing in the MSE and convergence speed for handling the non-Gaussian noises.

Next, the key parameters, λ and τ , are discussed regarding the performance effects on the devised KRMVLC algorithm. When a parameter is tuned, other parameters are fixed. λ was chosen from 0.15 to 0.9 with an interval of 0.15 to show the effect of the devised KRMVLC algorithm. In this case, other parameters are the same as the above simulations. The noise model in Experiment-1 was used herein. The achieved MSE are illustrated in Figure 4. From Figure 4, we see that the KRMVLC achieves the best MSE for $\lambda = 0.15$, meaning that λ has important effects on the estimation error behavior for the developed KRMVLC algorithm.

Then, the performance of the KRMVLC algorithms with different Versoria shape parameter is verified, where τ is set to be (1, 0.25, 0.0625, 0.01, 0.0025). In this experiment, other conditions are the same as in Experiment-1. The results obtained from the simulation are provided in Figure 5, which shows that the KRMVLC has the best behavior when $\tau = 1$.

Finally, the tracking ability for the derived KRMVLC was tested by considering an abrupt changing channel that was changed when 500 iterations had been done to analyze the tracking behavior. In this case, the measurement noise was considered to be the Laplace and Uniform distribution noises used above. After 500 iterations, the used channel was modified to $r(i) = 0.9x^2(i-1) + n(i) - x(i)$. The simulation results demonstrated in this paper were averaged from 50 independent Monte Carlo runs. The performance is illustrated in Figure 6. Results from the experiment show that the behavior of the KRMVLC is still superior to other algorithms, with a small change at the 500th iteration.

Figure 1. Nonlinear channel equalization.

Figure 2. General nonlinear channel model.

Table 1. Parameters for algorithms.

Algorithm	α	ω	μ	λ	γ
KLMS	-	0	0.02	-	-
KLMF	-	0	0.01	-	-
KLMMN	-	0.25	0.0225	-	-
KMVC	-	0	0.02	-	-
KRLS	1	0.25	-	0.45	1
KRGMN	1	0.25	-	0.45	1
KRMVLC	1	0.25	-	0.45	1

Experiment-1 Bernoulli-Gaussian noise

Experiment-2 Laplace-Gaussian noise

Experiment-3 Uniform-Gaussian noise

Experiment-4 Laplace noise

Figure 3. Convergence analysis of the KRMVLC for various mixed noises.

Figure 4. MSE performance against λ . Figure 5. MSE performance against τ .

Figure 6. Learning performance for the KRMVLC with an abrupt change.

4. Conclusions

In this paper, the kernel recursive maximum Versoria-like criterion (KRMVLC) has been developed and analyzed for estimating nonlinear channels in a non-Gaussian interference environment. The KRMVLC algorithm was invented via taking the considering the logarithmic second-order error and the MVC in RKHS. Simulations show that the KRMVLC achieved excellent estimation behavior when using the convergence and MSE as criteria. The KRMVLC algorithm provides much more benefits in various environments compared with the other aforementioned algorithms, and can combat abrupt channel changes. The results also indicate that the devised KRMVLC provides a powerful solution for handling nonlinear and non-Gaussian signals. In the future, we will discuss and analyze the KRMVLC in complex kernel adaptive filter areas.

Author Contributions: Q.W. wrote the first version of this paper and coded for the proposed algorithm. Y.L. proposed the algorithms and give the analysis of the idea, he is also the projection administration. W.X. gave the investigation of the algorithm and he got the funding. All the authors wrote and gave the discussion of this paper together, and they have read and approved the final version of this manuscript.

Funding: This paper is supported by the National Key Research and Development Program of China (2016YFE0111100), Key Research and Development Program of Heilongjiang (GX17A016), the Science and Technology innovative Talents Foundation of Harbin (2016RAXXJ044), the Natural Science Foundation of Beijing (4182077), China Postdoctoral Science Foundation (2017M620918, 2019T120134), the Fundamental Research Funds for the Central Universities (HEUCFG201829, 2072019CFG0801), and Opening Fund of Acoustics Science and Technology Laboratory (SSKF2016001).

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

References

1. Liu, W.; Príncipe, J.C.; Haykin, S. *Kernel Adaptive Filtering: A Comprehensive Introduction*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
2. Liu, W.; Pokharel, P.P.; Príncipe, J.C. The kernel least mean square algorithm. *IEEE Trans. Signal Process.* **2018**, *56*, 543–554. [[CrossRef](#)]
3. Li, Y.; Wang, Y.; Jiang, T. Sparse-aware set-membership NLMS algorithms and their application for sparse channel estimation and echo cancelation. *AEU Int. J. Electron. Commun.* **2016**, *70*, 895–902. [[CrossRef](#)]
4. Shi, W.; Li, Y.; Zhao, L.; Liu, X. Controllable sparse antenna array for adaptive beamforming. *IEEE Access* **2019**, *7*, 6412–6423. . [[CrossRef](#)]
5. Lu, L.; Zhao, H.; Chen, B. Time series prediction using kernel adaptive filter with least mean absolute third loss function. *Nonlinear Dyn.* **2017**, *90*, 999–1013. [[CrossRef](#)]
6. Engel, Y.; Mannor, S.; Meir, R. The kernel recursive least squares algorithm. *IEEE Trans. Signal Process.* **2004**, *52*, 2275–2285. [[CrossRef](#)]
7. Chen, B.; Zhao, S.; Zhu, P.; Príncipe, 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. Liu, W.; Park, I.; Príncipe, J.C. An information theoretic approach of designing sparse kernel adaptive filters. *IEEE Trans. Neural Netw.* **2009**, *20*, 1950–1961. [[CrossRef](#)]
10. Liu, W.; Park, I.; Wang, Y.; Príncipe, J.C. Extended kernel recursive least squares algorithm. *IEEE Trans. Signal Process.* **2009**, *57*, 3081–3814.
11. Wang, S.; Zheng, Y.; Ling, C. Regularized kernel least mean square algorithm with multiple-delay feedback. *IEEE Signal Process. Lett.* **2016**, *23*, 98–101. [[CrossRef](#)]
12. Zhao, J.; Liao, X.; Wang, S.; Tse, C.K. Kernel least mean square with single feedback. *IEEE Signal Process. Lett.* **2015**, *22*, 953–957. [[CrossRef](#)]
13. Chen, B.; Liang, J.; Zheng, N.; Príncipe, J.C. Kernel least mean square with adaptive kernel size. *Neurocomputing* **2016**, *191*, 95–106. [[CrossRef](#)]
14. Shi, L.; Lin, Y.; Xie, X. Combination of affine projection sign algorithms for robust adaptive filtering in non-gaussian impulsive interference. *Electron. Lett.* **2014**, *50*, 466–467. [[CrossRef](#)]

15. Pelekanakis, K.; Chitre, M. Adaptive sparse channel estimation under symmetric α -stable noise. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 3183–3195. [[CrossRef](#)]
16. Pei, S.C.; Tseng, C.C. Least mean p -power error criterion for adaptive FIR filter. *IEEE J. Sel. Areas Commun.* **1994**, *12*, 1540–1547.
17. Li, Y.; Wang, Y.; Jiang, T. Norm-adaption penalized least mean square/fourth algorithm for sparse channel estimation. *Signal Process.* **2016**, *128*, 243–251. [[CrossRef](#)]
18. Tanrikulu, O.; Chambers, J.A. Convergence and steady-state properties of the least-mean mixed-norm (LMMN) adaptive algorithm. *IEE Proc. Vis. Image Signal Process.* **1994**, *143*, 137–142. [[CrossRef](#)]
19. Chambers, J.A.; Tanrikulu, O.; Constantinides, A.G. Least mean mixed-norm adaptive filtering. *Electron. Lett.* **1994**, *30*, 1574–1575. [[CrossRef](#)]
20. Miao, Q.Y.; Li, C.G. Kernel least-mean mixed-norm algorithm. In Proceedings of the International Conference on Automatic Control and Artificial Intelligence (ACAI), Xiamen, China, 3–5 March 2012; pp. 1285–1288.
21. Luo, X.; Deng, J.; Liu, J.; Li, A.; Wang, W.; Zhao, W. A novel entropy optimized kernel least-mean mixed-norm algorithm. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 1716–1722.
22. Ma, W.; Qiu, X.; Duan, J.; Li, Y.; Chen, B. Kernel recursive generalized mixed norm algorithm. *J. Frankl. Inst.* **2018**, *355*, 1596–1613. [[CrossRef](#)]
23. Huang, F.; Zhang, J.; Zhang, S. Maximum versoria criterion-based robust adaptive filtering algorithm. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2017**, *355*, 1596–1613. [[CrossRef](#)]
24. Jain, S.; Mitra, R.; Bhatia, V. Kernel adaptive filtering based on maximum versoria criterion. In Proceedings of the 2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Indore, India, 16–19 December 2018.
25. Li, Y.; Jiang, Z.; Shi, W.; Han, X.; Chen, B. Blocked maximum correntropy criterion algorithm for cluster-sparse system identifications. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2019**. [[CrossRef](#)]
26. Shi, W.; Li, Y.; Wang, Y. Noise-free maximum correntropy criterion algorithm in non-Gaussian environment. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2019**. [[CrossRef](#)]
27. Li, Y.; Jiang, Z.; Osman, O.; Han, X.; Yin, J. Mixed norm constrained sparse APA algorithm for satellite and network echo channel estimation. *IEEE Access* **2018**, *6*, 65901–65908. [[CrossRef](#)]
28. Lerga, J.; Sucic, V.; Sersic, D. Performance analysis of the LPA-RICI denoising method. In Proceedings of the 2009 6th International Symposium on Image and Signal Processing and Analysis, Salzburg, Austria, 16–18 September 2009.
29. Lerga, J.; Grbac, E.; Sucic, V. An ICI based algorithm for fast denoising of video signals. *Automatika* **2014**, *55*, 351–358. [[CrossRef](#)]
30. Segon, G.; Lerga, J.; Sucic, V. Improved LPA-ICI-based estimators embedded in a signal denoising virtual instrument. *Signal Image Video Process.* **2017**, *11*, 211–218. [[CrossRef](#)]
31. Filipovic, V.Z. Consistency of the robust recursive Hammerstein model identification algorithm. *J. Frankl. Inst.* **2015**, *352*, 1932–1945. [[CrossRef](#)]