symmetry

MDPI

Article

A Novel Second-OrderSine-Cost-Function-Derived Kernel Adaptive Algorithm for Non-Linear System Identification

Liping Sun †, Hongju Zhou † and Hongwei Zhou *

College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China

* Correspondence: easyid@163.com

† These authors contributed equally to this work.

Abstract: A novel kernel recursive second-order sine adaptive (KRSOSA) algorithm was devised for identifying non-linear systems, which was constructed using a symmetry squared sine function to develop a novel kernel loss function and recursive scheme. In the proposed KRSOSA algorithm, the squared sine function provides resistance to impulsive noise due to the sine operation, which was well-derived and investigated in the framework of kernel adaptive filtering (KAF). The behavior of the proposed KRSOSA algorithm was investigated and analyzed using computer simulations, which provided good performance for identifying non-linear systems under impulsive noises.

Keywords: kernel adaptive filter; second-order sine cost function; non-Gaussian noise environments; non-linear system identifications

check for updates

Citation: Sun, L.; Zhou, H.; Zhou, H. A Novel Second-OrderSine-Cost-Function-Derived Kernel Adaptive Algorithm for Non-Linear System Identification. *Symmetry* **2023**, *15*, 827. https://doi.org/10.3390/sym15040827

Academic Editor: José Carlos R. Alcantud

Received: 28 February 2023
Revised: 25 March 2023
Accepted: 27 March 2023
Published: 29 March 2023

1. Introduction

Adaptive algorithms (AAs) are popular and useful for various signal processing, including linear and non-linear channel equalizers, DOA estimation, beamforming, noise reduction [1–5] , and so on. For these targets, many simple and effective algorithms have been presented, well-modeled, derived, analyzed, investigated, and discussed for system identifications, beamforming, DOA estimations, and channel estimations [1–16], which include the least mean square (LMS) [5–10], maximum correntropy criterion (MCC) [13–15], least mean fourth (LMF) [11,12], and so on. Although these algorithms achieve good performance to meet different applications in practical engineering, they might work poorly when there is a non-linear system with impulsive noise. Thus, it is necessary to develop non-linear AAs.

The kernel adaptive algorithm (KAA) has been developed for several years based on the least-mean-squares scheme and has been used for non-linear channel equalization [17–23] and tree pest prediction. The kernel adaptive filter (KAF) has had much more attention paid to it, which is always presented for non-linear predictions [24–26]. As we know, the KAF is famous for its online learning ability derived in the reproducing kernel Hilbert space (RKHS) [17–27]. Recently, many KAF algorithms have been presented for non-linear signal processing [28,29], including the kernel-driven least mean squares (KLMS) [21], kernel-driven least mean fourth (KLMF) [27], and kernel-driven recursive least squares (KRLS) [28] based on the symmetry squared error function. Subsequently, several variants of these methods have improved the performance and expanded the applications. Though these mentioned kernel-driven algorithms can achieve good performance for estimating non-linear systems, they still face performance degradation when the system is working in non-Gaussian environments, as the statistic second-order error scheme in these algorithms cannot provide resistance to pulse interference.

To improve the performance of these kernel-driven algorithms for non-Gaussian environment applications, the sign function and different error-criteria-promoted kernel adaptive filter (KAF) methods have been reported, including the high-order error scheme [27], correntropy error scheme [29], and mixture error scheme [30]. Considering the KAF theory, the kernel-driven least mean mixed norm (KLMMN) [17,18] and kernel-driven least mean fourth (KLMF) [27] have been presented. Compared with the KLMS, these algorithms reduce the estimation error and accelerate the convergence. Additionally, this theory has also been integrated into the correntropy error scheme to obtain the kernel-driven maximum correntropy criterion (KMCC) [29], and the kernel-driven recursive generalization mixed norm (KRGMN) [31] has also been presented for non-linear system identification when the system comes acrossnon-Gaussian noises.

This paper presents a novel kernel recursive second-order sine adaptive (KRSOSA) algorithm for non-linear system identification, where the KRSOSA algorithm is implemented using a squared sine to construct a new kernel cost function. In the KRSOSA algorithm, the squared sine function provides resistance to impulsive noise, and it is well-derived and investigated within the frame of the KAF. The behavior of the proposed KRSOSA was studied and analyzed using computer simulations, and it provided better performance for identifying non-linear systems under impulsive noise compared with popular kernel-driven algorithms, including the KLMS, KLMF, KLMMN, KMCC, and KRGMN. The contribution of the KRSOSA is a new cost function constructed by the squared sine and logarithmic functions, which provides a new iteration equation within the KAA.

2. The KRSOSA Algorithm

The mentioned kernel-like methods and their variant give the idea to convert the input data to a higher-dimensional featurespace via a non-linear mapping technique that is always described using $\varphi : \mathbf{U} \to \mathbf{F}$ [18–27]. For the KAF methods, $\mathbf{U}$ is the input and $\mathbf{F}$ is the operating space using the RKHS. In this way, the training data such as tree pest data can be described as $\{\mathbf{u}(m), d(m)\}_{m=1}^{N}$, where N is the number of elements, $d(m)$ is the expected signal in the estimation system, and $\mathbf{u}(m)$ is the system input. For the KAF methods, $\mathbf{u}(m)$ is transformed into $\varphi(\mathbf{u}(m))$ [23]. On the basis of Mercer's theorem [18], a general kernel-driven function is

$$\kappa\left(\mathbf{u}, \mathbf{u}'\right) = \varphi^T(\mathbf{u})\varphi(\mathbf{u}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{u}'\|}{\sigma^2}\right) . \tag{1}$$

In this case, σ describes the kernel width used in the kernel-driven functions given in Equation (1). From the aforementioned fundamentals, the KRSOSA was derived, analyzed, and discussed within the frame of KAFs. A logarithmic and a sine-weighted second-order method are illustrated to obtain a solution of the new combined cost function (CCF), which was used to create the KRSOSA method, and the CCF is

$$\begin{aligned} J(\Omega) = \sum_{n=1}^{m} \gamma^{m-n} \Bigg\{ \frac{\omega}{2} \log\left(1 + \frac{\left|d(n) - \Omega^T \varphi(m)\right|^2}{2\xi^2}\right) \\ + \frac{(1-\omega)}{2} \times 4\sin^2\left(\frac{\left|d(n) - \Omega^T \varphi(n)\right|}{2c}\right) \Bigg\} + \frac{1}{2}\gamma^m \lambda \|\Omega\|^2. \end{aligned} \tag{2}$$

$E[\cdot]$ is the expectation operation, while ξ is a constant, c is a small constant of a second-order sine function, and γ, a forgetting factor, strengthens the weights. λ and ω are the regularization and mixture parameters, respectively. For regularization, the implementation is realized via a norm constraint to make the inverse auto-correlation matrix. Then, we obtain the gradient of Equation (2):

$$\frac{\partial J(\Omega)}{\partial \Omega} = -\sum_{n=1}^{m} \gamma^{m-n} \varphi(n) z(n) d(n) + \left(\sum_{n=1}^{m} \gamma^{m-n} \varphi(n) z(n) \varphi^T(n) + \gamma^m \lambda\right)\Omega \ , \tag{3}$$

where $z(n)$ is written as

$$z(n)=\left\{\frac{\omega}{\left|d(n)-\Omega^{T}\varphi(n)\right|^{2}+2\xi^{2}}+\frac{(1-\omega)\sin\left(\frac{\left|d(n)-\Omega^{T}\varphi(n)\right|}{c}\right)}{c\left|d(n)-\Omega^{T}\varphi(n)\right|}\right\}. \tag{4}$$

Let Equation (3) be zero, and consider Ω:

$$\Omega=\left(\sum_{n=1}^{m}\gamma^{m-n}\varphi(n)z(n)\varphi^{T}(n)+\gamma^{m}\lambda\right)^{-1}\sum_{n=1}^{m}\gamma^{m-n}\varphi(n)z(n)d(n). \tag{5}$$

By considering

$$\mathbf{d}(m)=\begin{bmatrix} d(1), & \ldots & d(m-1), & d(m) \end{bmatrix}^{T}, \tag{6}$$

$$\boldsymbol{\Phi}(m)=\begin{bmatrix} \boldsymbol{\varphi}(1), & \ldots & \boldsymbol{\varphi}(m-1), & \boldsymbol{\varphi}(m) \end{bmatrix}, \tag{7}$$

$$\begin{aligned}
&\Psi(m)=\mathrm{diag}[\gamma^{m-1}\left\{\frac{\omega}{\left|d(1)-\boldsymbol{\Omega}^{T}\boldsymbol{\varphi}(1)\right|^{2}+2\xi^{2}}+\frac{(1-\omega)\sin\left(\frac{\left|d(1)-\boldsymbol{\Omega}^{T}\boldsymbol{\varphi}(1)\right|}{c}\right)}{c\left|d(1)-\boldsymbol{\Omega}^{T}\boldsymbol{\varphi}(1)\right|}\right\},\\
&\gamma^{m-2}\left\{\frac{\omega}{\left|d(2)-\boldsymbol{\Omega}^{T}\boldsymbol{\varphi}(2)\right|^{2}+2\xi^{2}}+\frac{(1-\omega)\sin\left(\frac{\left|d(2)-\boldsymbol{\Omega}^{T}\boldsymbol{\varphi}(2)\right|}{c}\right)}{c\left|d(2)-\boldsymbol{\Omega}^{T}\boldsymbol{\varphi}(2)\right|}\right\},\ldots,\\
&\left\{\frac{\omega}{\left|d(m)-\boldsymbol{\Omega}^{T}\boldsymbol{\varphi}(m)\right|^{2}+2\xi^{2}}+\frac{(1-\omega)\sin\left(\frac{\left|d(m)-\boldsymbol{\Omega}^{T}\boldsymbol{\varphi}(m)\right|}{c}\right)}{c\left|d(m)-\boldsymbol{\Omega}^{T}\boldsymbol{\varphi}(m)\right|}\right\}\Bigg],
\end{aligned} \tag{8}$$

Then Equation (5) is rewritten in matrix form:

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

We consider Equation (9) using the matrix inverse lemma to obtain

$$(\mathbf{A}+\mathbf{BCD})^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}, \tag{10}$$

and use

$$\gamma^{m}\lambda\mathbf{I}\rightarrow\mathbf{A},\boldsymbol{\Phi}(m)\rightarrow\mathbf{B},\boldsymbol{\Psi}(m)\rightarrow\mathbf{C},\boldsymbol{\Phi}^{T}(m)\rightarrow\mathbf{D}, \tag{11}$$

to obtain

$$\begin{aligned}
&\left(\boldsymbol{\Phi}(m)\boldsymbol{\Psi}(m)\boldsymbol{\Phi}^{T}(m)+\gamma^{m}\lambda\mathbf{I}\right)^{-1}\boldsymbol{\Phi}(m)\boldsymbol{\Psi}(m)\\
&=\boldsymbol{\Phi}(m)\left(\boldsymbol{\Phi}^{T}(m)\boldsymbol{\Phi}(m)+\gamma^{m}\lambda\boldsymbol{\Psi}^{-1}(m)\right)^{-1}.
\end{aligned} \tag{12}$$

$\boldsymbol{\Omega}(m)$ is modified to

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

Then, $\boldsymbol{\Omega}(m)$ is discussed using the linear combination of the input data:

$$\boldsymbol{\Omega}(m)=\boldsymbol{\Phi}(m)\mathbf{a}(m) \tag{14}$$

where $\mathbf{a}(m)$ is

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

Define $\mathbf{Q}(m)$ as

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

where $\mathbf{\Phi}(m) = \{\ \mathbf{\Phi}(m-1),\quad \boldsymbol{\varphi}(m)\ \}$, and after that, we have $\mathbf{Q}(m)$:

$$\mathbf{Q}(m) = \\ \begin{bmatrix} \begin{matrix} \mathbf{\Phi}^T(m-1)\mathbf{\Phi}(m-1) \\ +\gamma^m\lambda\mathbf{\Psi}^{-1}(m-1) \end{matrix} & \mathbf{\Phi}^T(m-1)\boldsymbol{\varphi}(m) \\ \boldsymbol{\varphi}^T(m)\mathbf{\Phi}(m-1) & \begin{matrix} \left\{\dfrac{\omega}{\left|d(m) - \mathbf{\Omega}^T\boldsymbol{\varphi}(m)\right|^2 + 2\xi^2} + \dfrac{(1-\omega)\sin\left(\frac{\left|d(m)-\mathbf{\Omega}^T\boldsymbol{\varphi}(m)\right|}{c}\right)}{c\left|d(m) - \mathbf{\Omega}^T\boldsymbol{\varphi}(m)\right|}\right\}^{-1} \\ \times\gamma^m\lambda + \boldsymbol{\varphi}^T(m)\boldsymbol{\varphi}(m) \end{matrix} \end{bmatrix}^{-1}. \tag{17}$$

Defining $\delta(m)$,

$$\delta(m) = \left\{\frac{\omega}{\left|d(m) - \mathbf{\Omega}^T\boldsymbol{\varphi}(m)\right|^2 + 2\xi^2} + \frac{(1-\omega)\sin\left(\frac{\left|d(m)-\mathbf{\Omega}^T\boldsymbol{\varphi}(m)\right|}{c}\right)}{c\left|d(m) - \mathbf{\Omega}^T\boldsymbol{\varphi}(m)\right|}\right\}^{-1} \tag{18}$$

Next, $\mathbf{Q}(m)$ is

$$\mathbf{Q}(m) = \begin{bmatrix} \begin{matrix} \mathbf{\Phi}^T(m-1)\mathbf{\Phi}(m-1) \\ +\gamma^m\lambda\mathbf{\Psi}(m-1)^{-1} \end{matrix} & \mathbf{\Phi}^T(m-1)\boldsymbol{\varphi}(m) \\ \boldsymbol{\varphi}^T(m)\mathbf{\Phi}(m-1) & \begin{matrix} \boldsymbol{\varphi}^T(m)\boldsymbol{\varphi}(m) \\ +\gamma^m\lambda\delta(m) \end{matrix} \end{bmatrix}^{-1}. \tag{19}$$

From the aforementioned analysis, we obtain

$$\mathbf{Q}^{-1}(m) = \begin{bmatrix} \mathbf{Q}^{-1}(m-1) & \mathbf{b}(m) \\ \mathbf{b}^T(m) & \boldsymbol{\varphi}^T(m)\boldsymbol{\varphi}(m) + \gamma^m\lambda\delta(m) \end{bmatrix} \tag{20}$$

with $\mathbf{b}(m) = \mathbf{\Phi}^T(m-1)\boldsymbol{\varphi}(m)$. In (21), the block matrix inversion operation is

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \\ \begin{bmatrix} \left(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}\right)^{-1} & -\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}\left(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}\right)^{-1} & \left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1} \end{bmatrix}. \tag{21}$$

Utilizing the block matrix inversion operation [1–10], Equation (19) is

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

with $\mathbf{f}(m) = \mathbf{Q}(m-1)\mathbf{b}(m)$. In (22), $\varepsilon(m) = \gamma^m\lambda\delta(m) + \boldsymbol{\varphi}^T(m)\boldsymbol{\varphi}(m) - \mathbf{f}^T(m)\mathbf{b}(m)$. Thus, $\mathbf{a}(m)$ is

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

In Equation (23), $e(m) = d(m) - \mathbf{b}^T(m)\mathbf{a}(m-1)$.

3. Simulation Results

Simulation examples were set up to verify the superiority of the devised KRSOSA algorithm over a non-linear channel estimation (NCE) when the communication environments are interfered with by different impulsive noises. A non-linear channel was modeled with a combination of a memory-less non-linear (MLNL) system and a linear filter. Here, a Gaussian kernel function, which is presented in Equation (3), was used to model the NCE channel, and the constructed NCE channel example is described in Figure 1. In Figure 1, $\{s(1), s(2), \ldots, s(L)\}$ is a binary signal, which was used as the inputof the NCE channel. After the signal has passed though a linear system using a signal transform $W(z) = 1 - 0.5z^{-1}$, we obtain an MLNL signal $x(m)$. At the receiver, we have $\{r(1), r(2), \ldots, r(L)\}$, which is obtained in the presence of noise $n(m)$, and its samples are $\{([r(m), r(m+1), \ldots, r(m+l)], s(m-D))\}$ with time embedded length l and equilibrium lag time D. $l = 3$ and $D = 2$ were used in all the discussions. The NCE channel was constructed using the input and the output. In this paper, $x(m) = s(m) + 0.5s(m-1)$ and $r(m) = x(m) - 0.9x(m)^2 + n(m)$. To obtain complex noise, $n(m)$ was mixed using a combination of $n_1(m)$ and $n_2(m)$ [16]. The created KRSOSA was analyzed by the use of Monte Carlo simulations, and its behaviors were compared with the KLMMN, KLMF, KLMS, and KRGMN. In the investigations, the related simulation parameters for the aforementioned algorithms were $\mu_{\text{KLMS}} = 0.125$, $\mu_{\text{KLMF}} = 0.01$, $\mu_{\text{KLMMN}} = 0.025$, $\mu_{\text{KMCC}} = 0.0125$, $\mu_{\text{KLMS}} = 0.125$, and $\omega_{\text{KLMMN}} = \omega_{\text{KRGMN}} = 0.25$, where the parameter for the corresponding algorithms is identified by subscripts, and $\sigma = 1$ was used in all the experiments. The rest of the parameters for the corresponding algorithms are not mentioned here as they are the same as those in [18,23,26–28].

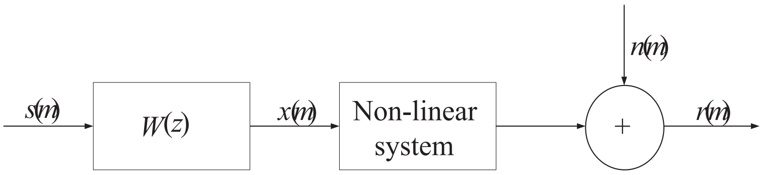

Figure 1. Non-linear channel used in the experiments.

Firstly, three examples were created to analyze the convergence of the devised KRSOSA algorithm by considering various mixture noises to model the impulsive noises, which are given as follows. (1) A Bernoulli distribution noise $n_1(m)$ and a Gaussian noise $n_2(m)$ were mixed with powers of 0.45 and 0.08 to model the mixture impulsive noise for the first example. (2) A Bernoulli distribution noise $n_1(m)$ and a Laplace noise $n_2(m)$ were mixed with powers of 0.45 to model the mixture impulsive noise for the second example. (3) A Bernoulli distribution noise $n_1(m)$ and a uniform noise $n_2(m)$ were mixed with powers of 0.45 and 1 to model the mixture impulsive noise for the third example.

In all the simulations, $\delta = 1$ and the total mixture noise power was 0.1. We used 1000 iterations to analyze the KRSOSA algorithm's behaviors, and 100 independent trialswere used to carry out a point. The convergence results for the constructed KRSOSA algorithm are illustrated in Figures 2–4 compared with the KLMMN, KLMF, KLMS, and KRGMN. From the convergence results, the KRSOSA algorithm converged the fastest under various noises. In Figure 2, the proposed KRSOSA algorithm converges faster than the latestKRGMN with the same MSE. From Figures 3 and 4, we can see that the proposed KRSOSA not only obtained a faster convergence, but also had a smaller MSE. On the other hand, the MSE behavior of the KRSOSA algorithm was the best compared with the KLMMN, KLMF, KLMS, and KRGMN since the KRSOSA algorithm combines the squared sine and logarithmic errors in the construction of the CCF to resist the impulsive noise. Finally, single non-Gaussian distribution noises to investigate the KRSOSA algorithm's performance are analyzed in Figure 5, where we only considered the Laplace distributed noise, whose power is 0.45 with parameter $\beta = 0.55$. The KRSOSA algorithm still converged the fastest and possessed the lowest MSE. Thus, we reached the conclusion that the KRSOSA algorithm is robust and has the best behaviors.

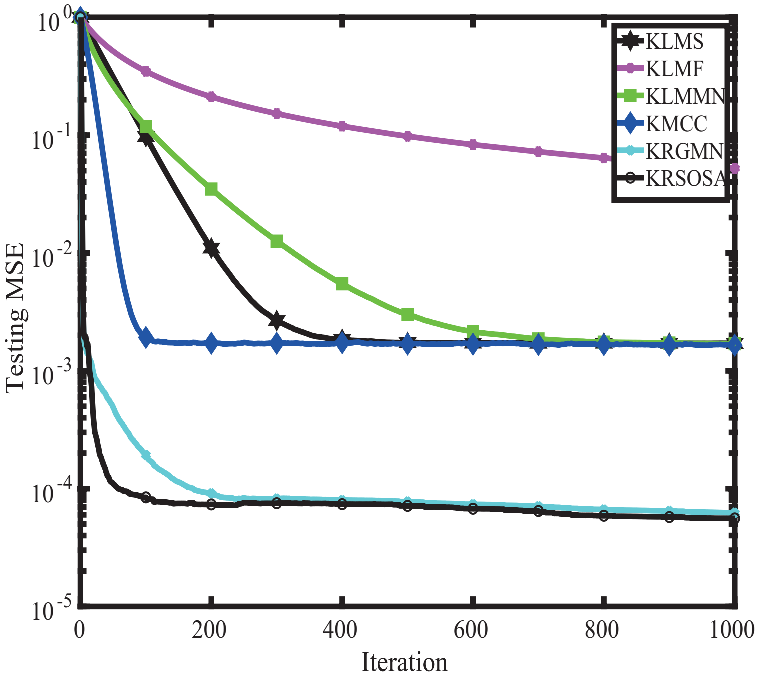

Figure 2. Bernoulli-Gaussian noise effects on the KRSOSA algorithm.

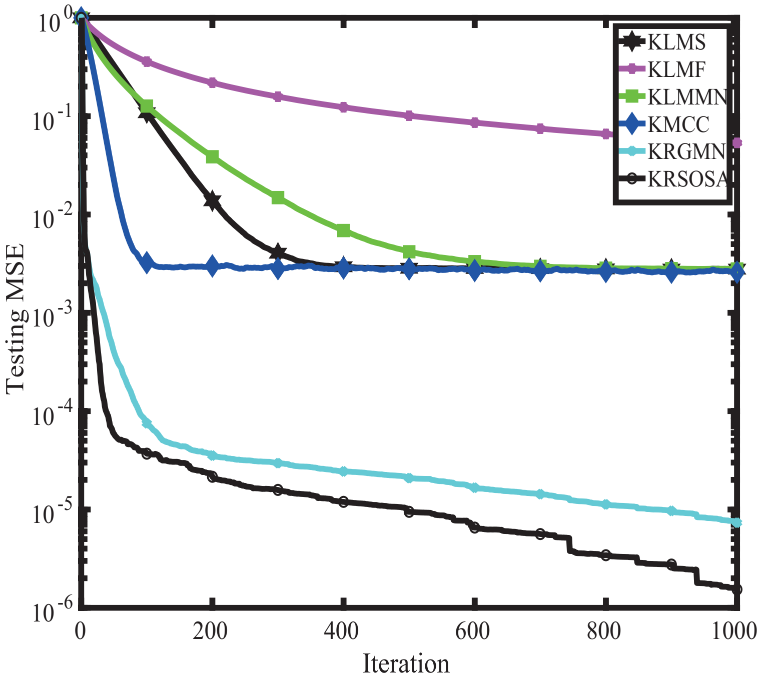

Figure 3. Bernoulli-Laplace noise effects on the KRSOSA algorithm.

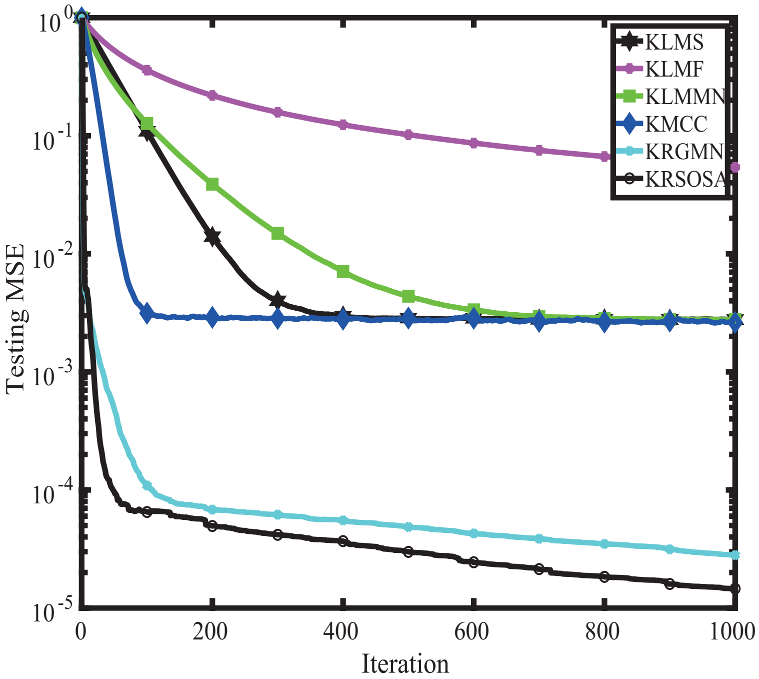

Figure 4. Bernoulli-uniform noise effects on the KRSOSA algorithm.

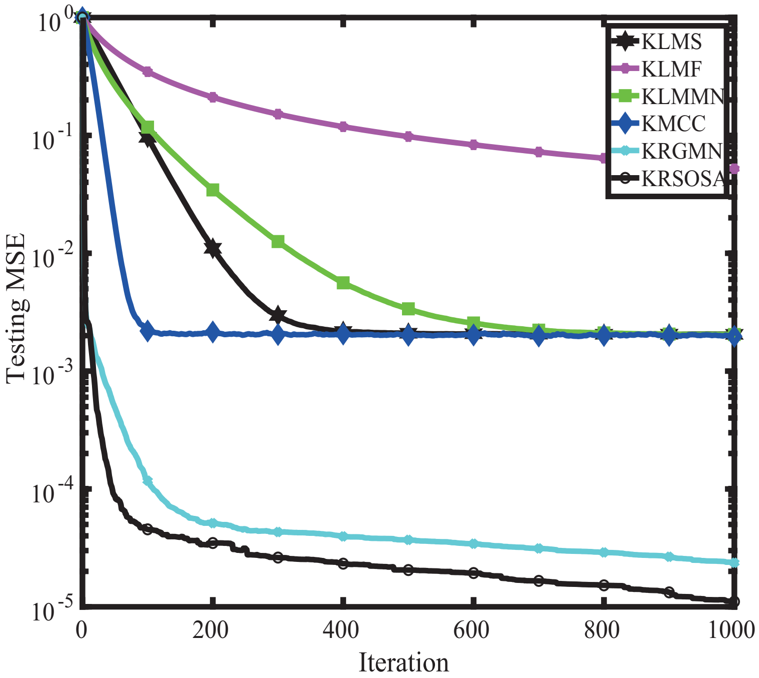

Figure 5. Only Laplace noise effects on the KRSOSA algorithm.

The effect of the parameters on the KRSOSA algorithm were studied with the simulation parameters listed in the former examples. One parameter was changed, while the others were left unchanged. Various regularization parameters λ, set as (0.45, 0.55, 0.6, 0.75, 0.85, and 0.95), of the KRSOSA algorithm were analyzed to discuss the convergence, while $\omega = 0.25$, $\gamma = 1$, $\alpha = 1$, and $\beta = 0.45$ for the mentioned algorithms were set to achieve the same MSE level. Here, the noise was the same as that in Figure 4, and the results are illustrated in Figure 6, which shows that the KRSOSA algorithm gave the smallest MSE for $\lambda = 0.55$, indicating that λ is vital to control the estimation behavior of the KRSOSA algorithm.

Then, different weights were investigated to analyze the effects on the MSE of the KRSOSA algorithm, where ω was selected from (0.15, 0.35, 0.55, 0.75, 0.95), and the system noise was the same as that in Figure 4. From the MSE behaviors in Figure 7, we found that ω can also affect the estimation error of the KRSOSA algorithm. When $\omega = 0.95$, the KRSOSA algorithm had the smallest MSE.

Next, different parameters c, namely $c = \{0.5, 1, 1.5, 2, 2.5, 3\}$, can also control the performance of the KRSOSA algorithm since c can change the shape of the CCF; hence, it changes the update equation of the KRSOSA algorithm. From the results in Figure 8, we observed that the KRSOSA algorithm had the smallest MSE for $c = 0.5$ since the squared sine function suppresses the magnitude of the large outliers. From Figures 6–8, we can see that the performance of the proposed KRSOSA algorithm was affected by the different parameters. Thus, in practical applications, we can adjust the mentioned parameters in Figures 6–8 to control the performance of the KRSOSA algorithm. In Figure 9, we can see that the proposed KRSOSA algorithm can track the abrupt changes of the system, which also shows its robustness.

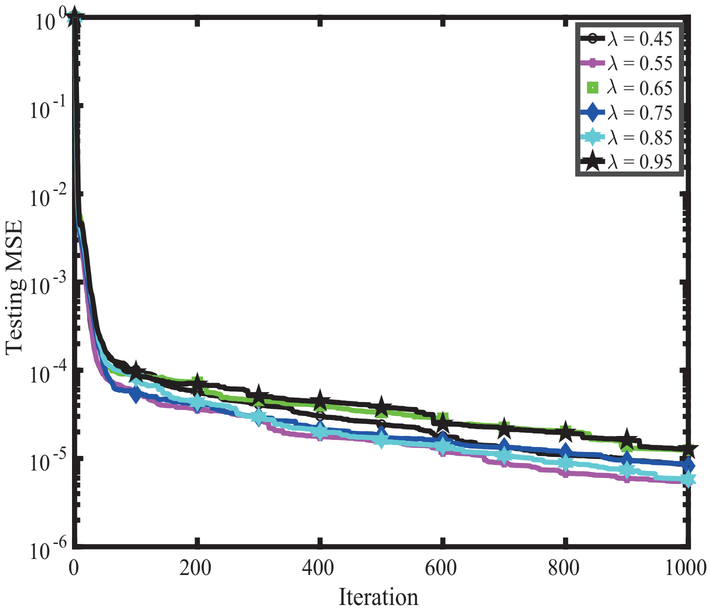

Figure 6. Different λ effects on the KRSOSA algorithm.

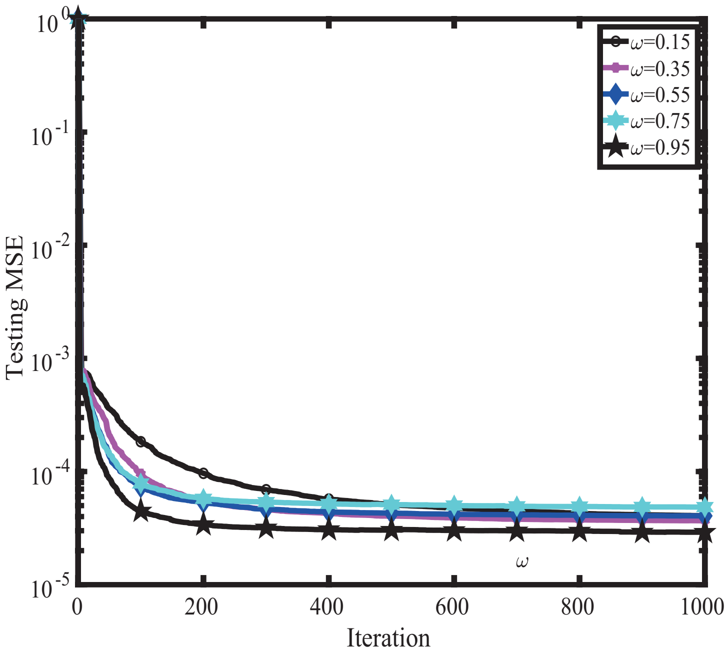

Figure 7. Different Ω effects on the KRSOSA algorithm.

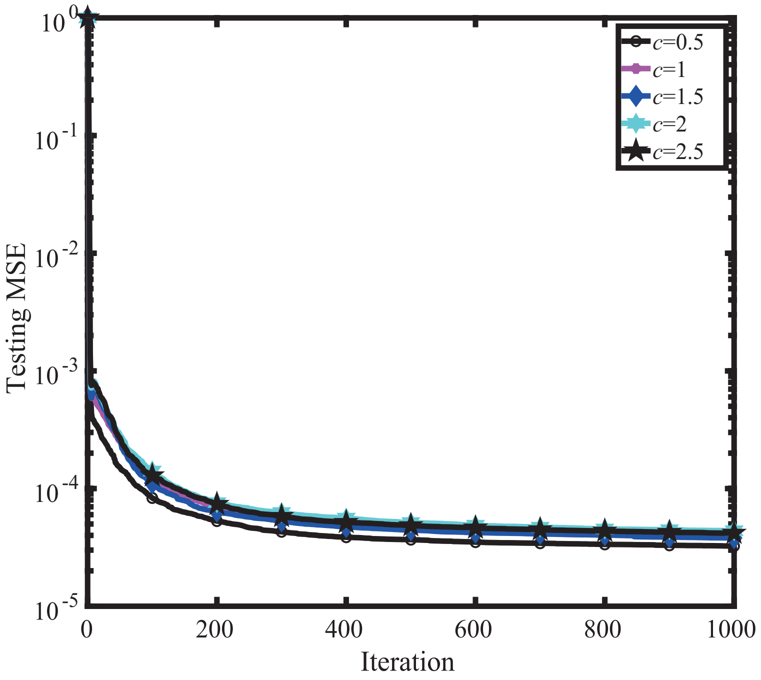

Figure 8. Different *c* effects on the KRSOSA algorithm.

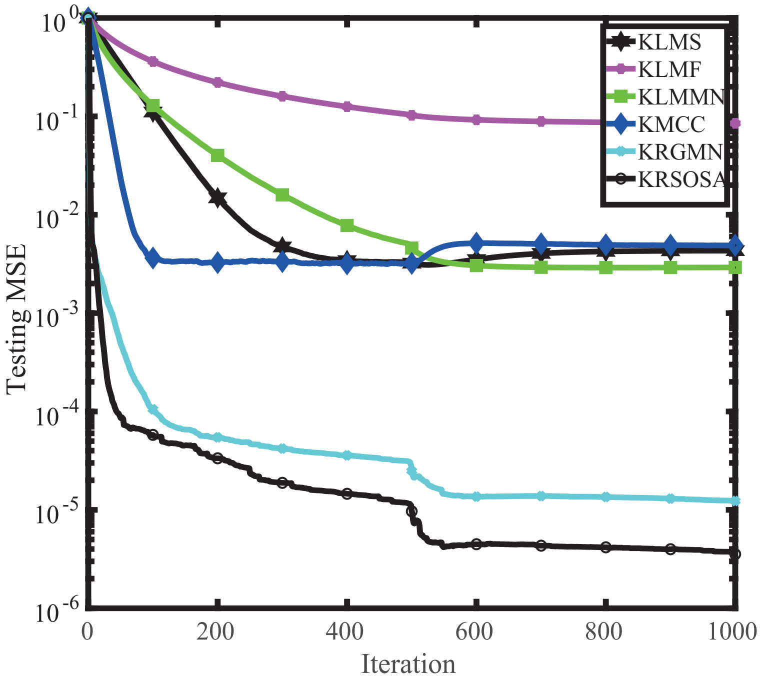

Figure 9. Abrupt change testing for the KRSOSA algorithm.

Finally, the tracking of the KRSOSA algorithm was investigated, where we used Laplace uniform mixture noise, and $r(i)$ was the same as the examples above. After 500 iterations, $r(i)$ changed to $r(i) = -x(i) + 0.9x^2(i-1) + n(i)$, and the results are given in Figure 9. It was found that the KRSOSA algorithm is still superior to the KLMMN, KLMF, KLMS, and KRGMN. In the future, the annual tree pest data will be collected, for which we can use the proposed KRSOSA algorithm to predict the tree pests' trends. Furthermore, we can also use the proposed KRSOSA algorithm to predict annual sunspots, as the methods in [18]. Since the parameters for the proposed KRSOSA algorithm still need adjusting, we will develop a joint parameter optimization for the KRSOSA algorithm.

4. Conclusions

A kernel recursive second-order sine adaptive (KRSOSA) algorithm was proposed and verified for non-linear channel equalization under various noises. The KRSOSA algorithm was realized by using a combination of the squared sine and logarithmic errors to construct a new cost function. The performance of the KRSOSA algorithm was compared with the recent kernel adaptive filtering algorithms, and the results confirmed that the KRSOSA algorithm achieved the best behaviors with respect to the convergence and MSE. In the future, we will develop a joint parameter optimization for the KRSOSA algorithm since there are three parameters that should be well selected.

Author Contributions: Conceptualization, H.Z. (Hongju Zhou); methodology, L.S.; software, L.S.; validation, H.Z. (Hongju Zhou); formal analysis, H.Z. (Hongju Zhou); investigation, L.S.; resources, H.Z. (Hongju Zhou); data curation, L.S.; writing—original draft preparation, L.S. and H.Z. (Hongwei Zhou); writing—review and editing, H.Z. (Hongwei Zhou); visualization, L.S.; supervision, H.Z. (Hongwei Zhou); project administration, H.Z. (Hongwei Zhou); funding acquisition, H.Z. (Hongwei Zhou). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program, Grant No. 2022YFD1401000, the Fundamental Research Funds for the Central Universities, Grant Number 2572022DP04, and the Forestry Science and Technology Promotion Demonstration Project of the Central Government, Grant Number Hei[2022]TG21.

Data Availability Statement: The data will be provided upon request.

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

Abbreviations

The following abbreviations are used in this manuscript:

KRSOSA	kernel recursive second-order sine adaptive
KAF	kernel adaptive filtering
KAA	kernel adaptive algorithm
RKHS	reproducing kernel Hilbert space
KLMS	kernel-driven least mean squares
KLMF	kernel-driven least mean fourth
KRLS	kernel-driven recursive least squares
KLMMN	kernel-driven least mean mixed norm
KMCC	kernel-driven maximum correntropy criterion
KRGMN	kernel-driven recursive generalization mixed norm
NCE	non-linear channel estimation
MSE	mean squared error

References

1. Gui, G.; Adachi, F. Improved least mean square algorithm with application to adaptive sparse channel estimation. *EURASIP J. Wirel. Commun. Netw.* **2013**, *2013*, 204.
2. Guo, K.; Guo, L.; Li, Y.; Zhang, L.; Dai, Z.; Yin, J. Efficient DOA Estimation Based on Variable Least Lncosh Algorithm under Impulsive Noise interferences. *Digit. Signal Process.* **2022**, *122*, 103383.
3. Shi, W.; Li, Y. A p-norm-like constraint LMS algorithm for sparse adaptive beamforming. *Appl. Comput. Electromagn. Soc. J.* **2019**, *34*, 1797–1803.
4. Steinbuch, K.; Widrow, B. A Critical Comparison of Two Kinds of Adaptive Classification Networks. *IEEE Trans. Electron. Comput.* **1965**, *14*, 737–740. [CrossRef]
5. Geladi, P.; Kowalski, B.R. Partial least-squares regression: A tutorial. *Anal. Chim. Acta* **1986**, *185*, 1–17.
6. Diniz, P.S.R. *Adaptive Filtering Algorithms and Practical Implementation*, 4th ed.; Springer: New York, NY, USA, 2013.
7. Widrow, B.; Stearns, S.D. *Adaptive Signal Processing*; Prentice Hall: Hoboken, NJ, USA, 1985.
8. Chen, Y.; Gu, Y.; Hero, A.O. Sparse LMS for system identification. In Proceedings of the IEEE International Conference on Acoustic Speech and Signal Processing, (ICASSP'09), Taipei, Taiwan, 19–24 April 2009; pp. 3125–3128.
9. Li, Y.; Wang, Y.; Jiang, T. Norm-adaption penalized least mean square/fourth algorithm for sparse channel estimation. *Signal Process.* **2016**, *128*, 243–251.

10. Li, Y.; Wang, Y.; Jiang, T. Sparse-aware set-membership NLMS algorithms and their application for sparse channel estimation and echo cancelation. *AEÜ-Int. J. Electron. Commun.* **2016**, *70*, 895–902.
11. Walach, E.; Widrow, B. The least mean fourth (LMF) adaptive algorithm and its family. *IEEE Trans. Inf. Theory* **1984**, *30*, 275–283. [CrossRef]
12. Narasimhan, S.V. Invited paper Application of the least mean fourth (LMF) adaptive algorithm to signals associated with gaussian noise. *Int. J. Electron.* **1987**, *62*, 895–913. [CrossRef]
13. Ma, W.; Duan, J.; Li, Y.; Chen, B. Proportionate adaptive filtering algorithms based on mixed square/fourth error criterion with unbiasedness criterion for sparse system identification. *Int. J. Adapt. Control Signal Process.* **2018**, *32*, 1644–1654.
14. 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]
15. Li, Y.; Jiang, Z.; Shi, W.; Han, X.; Chen, B.D. Blocked maximum correntropy criterion algorithm for cluster-sparse system identification. *IEEE Trans. Circuits Syst. II Express Briefs* **2019**, *66*, 1915–1919. [CrossRef]
16. Shi, W.; Li, Y.; Chen, B. A separable maximum correntropy adaptive algorithm. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 2797–2801. [CrossRef]
17. Wu, Q.; Li, Y.; Jiang, Z.; Zhang, Y. A Novel Hybrid Kernel Adaptive Filtering Algorithm for Nonlinear Channel Equalization. *IEEE Access* **2019**, *7*, 62107–62114. [CrossRef]
18. Liu, W.; Príncipe, J.C.; Haykin, S. *Kernel Adaptive Filtering: A Comprehensive Introduction*; Wiley: Hoboken, NJ, USA, 2010.
19. Wu, Q.; Li, Y.; Xue, W. A quantized adaptive algorithm based on the q-Rényi kernel function. *Digit. Signal Process.* **2022**, *120*, 103255.
20. Wu, Q.; Li, Y.; Zakharov, Y.; Xue, W. Quantized Kernel Least Lncosh Algorithm. *Signal Process.* **2021**, *189*, 108255. [CrossRef]
21. Liu, W.; Pokharel, P.P.; Príncipe, J.C. The kernel least-mean-square algorithm. *IEEE Trans. Signal Process.* **2008**, *56*, 543–554. [CrossRef]
22. Wu, Q.; Li, Y.; Xue, W. A Parallel Kernelized Data-Reusing Maximum Correntropy Algorithm. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 2792–2796. [CrossRef]
23. Wu, Q.; Li, Y.; Zakharov, Y.; Xue, W.; Shi, W. A kernel affine projection-like algorithm in reproducing kernel hilbert space. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 2249–2253. [CrossRef]
24. Engel, Y.; Mannor, S.; Meir, R. The kernel recursive least-squares algorithm. *IEEE Trans. Signal Process.* **2004**, *52*, 2275–2285. [CrossRef]
25. 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]
26. 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]
27. Ma, W.; Duan, J.; Man, W.; Zhao, H.; Chen, B. Robust Kernel Adaptive Filters Based on Mean p-power Error for Noisy Chaotic Time Series Prediction. *Eng. Appl. Artif. Intell.* **2017**, *58*, 101–110. [CrossRef]
28. Liu, W.; Park, I.; Wang, Y.; Príncipe, J.C. Extended kernel recursive least squares algorithm. *IEEE Trans. Signal Process.* **2009**, *57*, 3801–3814.
29. Zhao, S.; Chen, B.; Príncipe, J.C. Kernel adaptive filtering with maximum correntropy criterion. In Proceedings of the the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 2012–2017.
30. 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.
31. Ma, W.; Qiu, X.; Duan, J.; Li, Y.; Chen, B. Kernel recursive generalized mixed norm algorithm. *J. Frankl. Inst.* **2018**, *355*, 1596–1613. [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.