An Efficient Filtering Algorithm against Impulse Noise in Communication Systems

Yijie Tang[1], Guobing Qian[1], Wenqi Wu[1], Ying-Ren Chien[2]*

[1] *College of Electronic and Information Engineering, Southwest University, China*
[2] *Department of Electrical Engineering, National Ilan University, Taiwan*
tangyijie2001@163.com, guobingqian@swu.edu.cn, lifewhowwq@163.com, yrchien@niu.edu.tw

Abstract

The kernel adaptive filter (KAF), which processes data in the reproducing kernel Hilbert space (RKHS), can improve the performance of conventional adaptive filters in nonlinear systems. However, the presence of impulse noise can seriously degrade the performance of KAF. In this paper, we propose a kernel modified-sign least-mean-square algorithm (KMSLMS) to mitigate the impact of impulse noise in communication systems. Moreover, we apply the nearest-instance-centroid estimation (NICE) algorithm to reduce the computational complexity of our KMSLMS algorithm, called the NICE-KMSLMS algorithm. Finally, computer simulations were used to evaluate the effectiveness of our proposed method. Compared with the conventional kernel least-mean-square algorithm (KLMS), our proposed method can improve the testing mean-squared error (MSE) by 2.32 dB and 7.39 dB for the nonlinear channel equalization and Mackey-Glass chaotic time series prediction problems, respectively. Furthermore, the testing MSE degradation caused by combining the NICE algorithm with our KMSLMS algorithm is negligible but can save about 55% computational cost in terms of the required mean size.

Keywords: Impulse noise, Kernel least-mean-square (KLMS) algorithm, Nearest-instance-centroid estimation (NICE), Nonlinear system

1 Introduction

Various filtering algorithms have been widely used in communication systems [1-3], but conventional linear filtering algorithms can only work in linear systems, and communication systems in practical problems are often nonlinear. Many nonlinear adaptive filtering algorithms, such as the adaptive filter based on the genetic algorithm [4], the digital noise reduction scheme [5], the Volterra model based on the Volterra series [6], and the communication systems based on deep learning [7], were proposed to solve the issue. However, these nonlinear filtering models have problems, such as high complexity, unstable convergence, and difficulty determining the proper order. Therefore, how to design a stable and reliable nonlinear adaptive filtering algorithm has practical significance.

Recently, the kernel adaptive filter (KAF) has been proposed, which filters data in reproducing kernel Hilbert spaces with a linear structure but implements nonlinear filtering for the input space to deal with nonlinear filtering problems. The main idea of the kernel method (KM) [8] is to convert the computation of the inner product in high-dimensional feature space to the calculation of kernel function in input space. Moreover, KM has been successfully introduced into machine learning and nonlinear estimation, proving KM's superior performance in dealing with nonlinear problems. However, two drawbacks need to be solved for the KAF-based algorithms. Firstly, the computational complexity of the KAF algorithms is too high. The complicated structure of KAF increases linearly with the amount of input data and thus seriously restricts its practical value. Many sparsification techniques have been proposed to reduce the necessary computational costs, such as the novelty criterion (NC) [9], the data-selective approach [10], and the K-means clustering method [11]. Although these sparsification techniques can effectively reduce computational complexity, these methods may significantly decrease the convergence accuracy. In this case, the nearest-instance-centroid estimation (NICE) has been proposed [12] to deal with this issue. NICE uses the predefined distance threshold to determine the number of clusters, so the final sparse filtering network can effectively reduce the computational complexity of the accounting method. However, an inappropriate clustering formation could be done in the presence of impulse noise. Secondly, the presence of impulse noise can significantly degrade the performance of the KAF. Moreover, impulse noise is far more destructive than other noise in high-quality digital communication [13]. Thus, it is necessary to design a KAF against impulse noise; little literature has dealt with this problem to the best of the author's knowledge. Applying the sign function to error signals to mitigate the impact of the impulse noise has been widely used [14-15]. However, the sign function may cause a slow convergence rate. In this paper, we propose using a modified sign (MS) function to deal with the impulse noise's impact and apply it to the kernel least-mean-square algorithm (hereinafter referred to as the KMSLMS algorithm). Moreover, we combine the concept of the NICE method with our KMLMS (hereinafter referred to as the NICE-KMSLMS algorithm) to reduce the computational complexity of the KMSLMS algorithm. The effectiveness of our proposed

*Corresponding Author: Ying-Ren Chien; E-mail: yrchien@niu.edu.tw
DOI: 10.53106/160792642023032402014

method was evaluated by nonlinear channel equalization and chaotic time serious prediction problem.

The remainder of this paper is organized as follows. Section 2 describes the system model. Section 3 outlines the proposed KMSLMS algorithm and the NICE-KMSLMS algorithm. Section 4 presents two numerical simulation results to confirm the effectiveness of our proposed method. Finally, conclusions are drawn in Section 5.

2 System Models

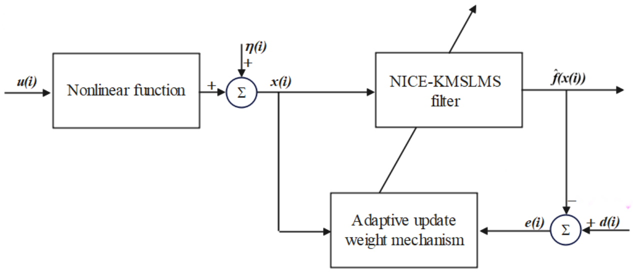


Figure 1. System model of a nonlinear adaptive filtering problem

Figure 1 illustrates the system model of a nonlinear adaptive filtering problem. The input of the adaptive filter ($x(i)$) can be expressed as follow:

$$x(i) = g\left(u(i)\right) + \eta(i)\,, \tag{1}$$

where i is the time index; $u(i)$denotes the input of a nonlinear function $g(\cdot)$; the output of $g(u(i))$ is corrupted by an additive impulse noise $\eta(i)$, which can be modeled as follows:

$$\eta(i) = \zeta(i) \cdot \rho(i)\,, \tag{2}$$

where $\zeta(i)$ is a Bernoulli random variable with the success probability p_c, i.e., $p[\zeta(i) = 1] = p_c$ and $p[\zeta(i) = 0] = 1 - p_c$; $\rho(i)$ is a continuously uniformly distributed random variable with range $[-L, L]$.

The adaptive update weight aims at adjusting our proposed NICE-KMSLMS filter so that the error signal $e(i)$, can be minimized in the minimum mean square error (MMSE) sense. The error signal can be expressed as follow:

$$e(i) = d(i) - \hat{f}\left(\boldsymbol{x}(i)\right), \tag{3}$$

where $d(i)$ denotes the desired signal and $\hat{f}\left(\boldsymbol{x}(i)\right)$ is the output of the adaptive filter.

3 Proposed Method

3.1 KMSLMS Algorithm

Inspired by the conventional error-sign LMS algorithm, we devise a modified sign function to mitigate the impact of impulse noise for the KLMS algorithm. The proposed modified sign function $MS(\cdot)$ of $e(i)$ can be expressed as follows:

$$MS(e(i)) = \begin{cases} \gamma \operatorname{sgn}[e(i)] & ,\text{if } |e(i)| \geq \beta \\ \delta e(i) + (1-\delta)\operatorname{sgn}[e(i)] & ,\text{otherwise} \end{cases} \tag{4}$$

where $\operatorname{sgn}(\cdot)$ denotes the sign function. Note that the parameter γ can balance the resulting convergence rate and accuracy when the value of $|e(i)|$ is greater than a predetermined threshold β; on the other hand, we combine $e(i)$ with $\operatorname{sgn}[e(i)]$ via linear interpolation; the combination parameter δ is a positive value (less than one). The function curve of the proposed MS function is depicted in Figure 2.

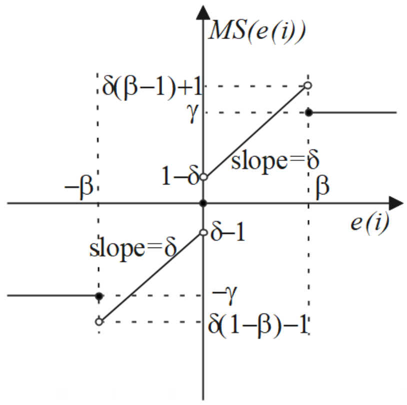


Figure 2. The proposed modified sign (MS) function curve

Next, we evolve the KMSLMS algorithm by replacing the $e(i)$ with $MS(e(i))$ as follows:

$$\boldsymbol{w}(i) = \boldsymbol{w}(i-1) + \mu MS(e(i))\varphi\left(\boldsymbol{x}(i)\right), \tag{5}$$

where $\boldsymbol{w}(i)$ denotes the estimate of the weight vector in feature space induced by the kernel mapping $\varphi(\cdot)$; μ is the step size. By iteratively replacing i with $i - 1$, We can further rewrite Eq. (5) as follows:

$$\begin{aligned} \boldsymbol{w}(i) &= \left[\boldsymbol{w}(i-2) + \mu MS\left(e(i-1)\right)\varphi(\boldsymbol{x}(i-1))\right] \\ &\quad + \mu MS(e(i))\varphi(\boldsymbol{x}(i)) \\ &= \boldsymbol{w}(0) + \mu \sum_{j=1}^{i} MS(e(j))\varphi(\boldsymbol{x}(j)) \\ &= \mu \sum_{j=1}^{i} MS(e(j))\varphi(\boldsymbol{x}(j)). \end{aligned} \tag{6}$$

Note that the last equality holds if we assume the initial weight vector $\boldsymbol{w}(0) = \mathbf{0}$. At the end of the training phase, the weight estimation at time i, the output of the system to a new input $\boldsymbol{x}'$ can be expressed as follows:

$$\begin{aligned} \boldsymbol{w}^T(i)\varphi(\boldsymbol{x}') &= \mu \sum_{j=1}^{i} MS(e(j))\left[\varphi^T(\boldsymbol{x}(j))\varphi(\boldsymbol{x}')\right] \\ &= \mu \sum_{j=1}^{i} MS(e(j))\kappa\left(\boldsymbol{x}(j), \boldsymbol{x}'\right). \end{aligned} \tag{7}$$

Assumed that a Gaussian kernel is adopted as the kernel function, the proposed KMSLMS algorithm can be expressed as follows:

$$\hat{f}(x(i)) = \sum_{j=1}^{i-1} \alpha_j (i-1) \exp(-A \|x(i) - x(j)\|^2), \tag{8}$$

where $\hat{f}(\cdot)$ denotes the estimation of the nonlinear mapping; $\alpha_j(i-1)$ is the j -th component of the vector $\alpha(i-1)$; A is a kernel parameter, which is used to adjust the Gaussian kernel size. Note that the new component of $\alpha(i)$ is the coefficient $\mu MS(e(i))$ with the error signal $e(i) = d(i) - \hat{f}(\boldsymbol{x}(i))$.

3.2 NICE-KMSLMS Algorithm

From Eq. (8), we can observe that one of the main drawbacks of the kernel adaptive filtering algorithm is the high computational complexity. The size of the structure $\alpha(i)$ grows for each new sample. The nearest-instance-centroid-estimation (NICE) algorithm introduces the concept of online clustering into the process of selecting dictionaries to reduce the computational complexity involved in Eq. (8). We denote a set of a cluster as $\boldsymbol{C} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, ..., \boldsymbol{C}_{|C|}\}$, where $|\boldsymbol{C}|$ represents the number of clusters in the set. The number of elements within the j -th cluster is denoted as Z_j. The distance between the input $\boldsymbol{x}(i)$ and the nearest cluster is as follows:

$$d_{\min} = \min_{1<j<|\boldsymbol{C}|} \left\| \boldsymbol{x}(i) - \bar{\boldsymbol{c}}_j \right\|^2, \tag{9}$$

where $\bar{\boldsymbol{c}}_j$ represent the centroid of the j -th cluster. Let the index of the nearest cluster for $\boldsymbol{x}(i)$ be j^*, which can be expressed as follows:

$$j^* = \arg\min_{1<j<|\boldsymbol{C}|} \left\| \boldsymbol{x}(i) - \bar{\boldsymbol{c}}_j \right\|^2. \tag{10}$$

If $d_{\min} < d_c$, the new input $\boldsymbol{x}(i)$ is joined its nearest $\boldsymbol{C}_{j^*}$. Note that the value of the threshold d_c can be chosen from the range $(0, 3/\sqrt{2\mathrm{A}}]$. In this case, the centroid of the j^* -th cluster $(\bar{\boldsymbol{c}}_{\mathbf{j}^*}(i))$ is updated as follows:

$$\bar{\boldsymbol{c}}_{\mathbf{j}^*}(i) = \frac{\bar{\boldsymbol{c}}_{\mathbf{j}^*}(i) \times \mathrm{Z}_{\mathrm{j}^*} + \boldsymbol{x}(i)}{\mathrm{Z}_{\mathrm{j}^*} + 1}. \tag{11}$$

Note that the size of the j^* -th cluster will be added by one after the new input $\boldsymbol{x}(i)$ joined. On the other hand, a new cluster will be formed if $d_{\min} > d_c$ holds. The initial size of the newly formed cluster is one, and its initial centroid is $\boldsymbol{x}(i)$. Therefore, Eq. (8) can be calculated with fewer computational resources as follows:

$$\hat{f}(x(i)) = \sum_{l=1}^{Z_{j^*}} \alpha_{j^*}^{(l)} \exp(-A \| C_{j^*}^{(l)} - x(i) \|^2), \tag{12}$$

where $\alpha_{j^*}^{(l)}$ denotes the weight of the ℓ -th element within the j^* -th cluster ($\boldsymbol{C}_{j^*}^{(\ell)}$). Table 1 summarizes the proposed NICE-KMSLMS. Note that the proposed modified sign function significantly affects the weight values associated with the j^* -th cluster.

Table 1. Proposed NICE-KMSLMS algorithm

Initialization:

The centroid distance threshold and: d_c

The kernel parameter: A

The learning rate: μ

The initial weight: $\alpha_1 - [\mu d(1)]$

Set of the clusters: $\boldsymbol{C} = \{\boldsymbol{C}_1\}$

The centroid of the first cluster: $\bar{\boldsymbol{c}}_1 = \boldsymbol{x}(1)$

The size of the first cluster: $Z_1 - 1$

The initial cluster: $\boldsymbol{C}_1 = \{\boldsymbol{x}(1)\}$

Computation:

while $\{\boldsymbol{x}(i), d(i)\}$ available **do**

Compute the minimum centroid distance

$$d_{\min} = \min_{1< \ <|\boldsymbol{C}|} \left\| \boldsymbol{x}(i) - \bar{\boldsymbol{c}}_j \right\|^2$$

Select the nearest-neighbor cluster

$$j^* = \arg\min_{1<j<|\boldsymbol{C}|} \left\| \boldsymbol{x}(i) - \bar{\boldsymbol{c}}_j \right\|^2$$

Compute the output

$$\hat{f}(\boldsymbol{x}(i)) = \sum_{l=1}^{Z_j} \alpha_j^{(l)} \exp(-A \left\| C_j^{(l)} - \boldsymbol{x}(i) \right\|^2)$$

$$e(i) = d(i) - \hat{f}(\boldsymbol{x}(i))$$

Nearest-Instance-Centroid-Estimation:

If $d_{\min} < d_c$ **then**

Update the weights of filter j^*

$$\alpha_{j^*} = [\alpha_{j^*}, \mu MS(e(i))]$$

Update the cluster j^*:

$$C_{j^*} = \{C_{j^*}, x(i)\}$$

Update the centroid and size of the j^* -th cluster:

$$\bar{\boldsymbol{c}}_j = \frac{Z_{j^*} \bar{\boldsymbol{c}}_j + \boldsymbol{x}(i)}{Z_{j^*} + 1}, Z_{j^*} = Z_{j^*} + 1$$

else

Form a new cluster

$\boldsymbol{C}_{|c|+1} = \{\boldsymbol{x}(i)\}$

Set the centroid and size for the new cluster:

$$\bar{\boldsymbol{c}}_{|\boldsymbol{C}|+1} = \boldsymbol{x}(i), Z_{|\boldsymbol{C}|+1} = 1$$

The weight of the newly formed cluster:

$$\acute{\boldsymbol{a}}_{|\boldsymbol{C}|+1} = \left[\sum_{l=1}^{Z_{j^*}} {}_{\mathrm{j}^*}^{(\ell)} + \mu MS(e(i)) \right]$$

Cluster Updating:

$\boldsymbol{C} = \{\boldsymbol{C}, \boldsymbol{C}_{|\boldsymbol{C}|+1}\}$

end

end while

4 Simulation and Analysis

To verify the superiority of the NICE-KMSLMS algorithm, we have conducted simulations for the nonlinear channel equalization and chaotic time series prediction problems, in which the observed data was contaminated by impulse noise $\eta(i)$. The occurrence probability and strength are chosen as $p_c = 0.05$ and $L = 6$, respectively (see Eq. (2)). The performance metric to assess the performance of adaptive filters is testing mean-square-error (MSE); the computational complexity was evaluated by using the mean size, which is the average number of centers within clusters [12]. In addition, we use the execution time for various adaptive filtering algorithms to evaluate the computational cost. Three comparable works were considered in the simulations: LMS, KLMS [16], and NICE-KLMS [12]. The statistics of all performance metrics were calculated by averaging over 100 independent Monte Carlo simulations. We generate 3,000 and 150 data as training and testing data, respectively. We select the average of the last 100 testing MSEs in the simulations to measure convergence accuracy. The testing MSE can be calculated as follows [17]:

$$\text{Testing MSE} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{n}\sum_{j=1}^{n}\left\|\boldsymbol{x}(j)-\hat{\boldsymbol{x}}(j)\right\|^2\right), \quad \textbf{(13)}$$

where $\hat{x}(j)$ is the estimation of $\boldsymbol{x}(j)$; n and m denote the numbers of testing data and Monte Carlo simulations, respectively.

4.1 Case-1: Nonlinear Channel Equalization Problem

Figure 3 illustrates the system model of the nonlinear channel equalization problem. The desired signal $d(i)$ is the time delay of the input signal $u(i)$, which is +1 or −1 with equal probability. The time delay was set as two in case-1. In this simulation, the transfer function of the system model is set as follows:

$$H(z) = 1 + 0.5z^{-1}. \quad \textbf{(14)}$$

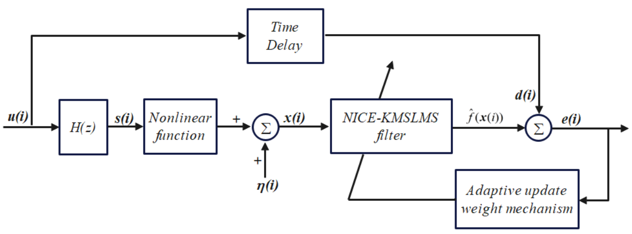


Figure 3. System model of a nonlinear channel equalization problem

The observed data at the output of the nonlinear channel $\boldsymbol{x}(i)$ can be modeled as follows:

$$x(i) = s(i) - 0.7s^2(i) + 0.15s^3(i) + \eta(i). \quad \textbf{(15)}$$

We use an exhaustive search method to choose suitable parameters for δ and β to achieve the lower values of the testing MSE in case-1. Table 2 and Table 3 summarize part of the evaluation results on case-1 with various values of δ and β, respectively. Eventually, we choose $\delta = 0.8$ and $\beta = 2.67$ for case-1.

Table 2. The testing MSE of NICE-KMSLMS on case-1 with various δ

δ	0.1	0.3	0.5	0.7	0.9
Testing MSE (x10^{-1})	3.94	3.77	3.49	2.82	3.20

Table 3. The testing MSE of NICE-KMSLMS on case-1 with various β

β	0.5	1.5	2.5	3.5	4.5
Testing MSE (x10^{-1})	4.19	3.38	2.42	3.22	3.73

Figure 4 illustrates the learning curves for various adaptive algorithms. Obviously, the conventional linear LMS algorithm diverged during the adaptation processes. Although the KLMS and NICE-KLMS converged, the impulse noise can seriously degrade the resulting testing MSE. However, our proposed KMSLMS significantly outperforms these related works. It can confirm that our proposed modified sign method can effectively combat the impulse noise in this case. The resulting testing MSE improvements over the KLMS are 2.32 dB on average (from 0.4079 to 0.2391). Compared with the KMSLMS algorithm, its low-complexity version (i.e., NICE-KMSLMS) has a negligible performance loss (0.0021 dB, from 0.2412 to 0.2391) but can reduce the mean size of clusters by 55.59% (from 3204.2 to 1423.1) on average. Table 4 summarizes the comparison results.

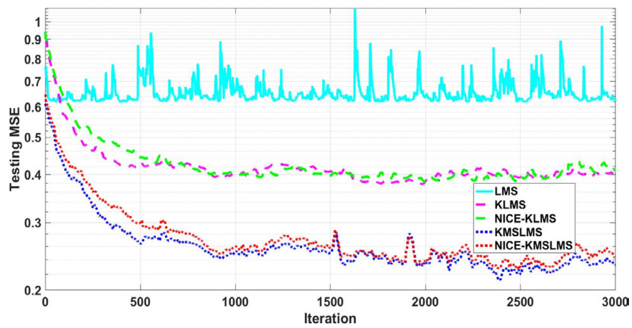


Figure 4. The learning curves comparisons in case-1 with $\mu = 0.31$, $A = 0.2$, $d_c = 2.96$, and $\gamma = 3.0$

Table 4. Comparison results for case-1

Algorithm	Mean size	Time	Testing MSE
LMS	-	11.8s	-
KLMS	2300.0	38.2s	0.4079
NICE-KLMS	1071.4	17.9s	0.4247
KMSLMS	3204.2	46.1s	0.2391
NICE-KMSLMS	1423.1	19.2s	0.2412

4.2 Case-2: Mackey-Glass (MG) Chaotic Time Series Prediction Problem

In this case, the short-term prediction of the Mackey-Glass (MG) chaotic time series was used further to evaluate the superiority of the proposed NICE-KMSLMS algorithm.

The chaotic time series $u(t)$ are generated according to the time-delay ordinary differential equation [18] as follows:

$$\frac{du(t)}{dt} = -bu(t) + \frac{au(t-\tau)}{1+\left[u(t-\tau)\right]^{10}}, \quad \textbf{(16)}$$

where $a = 0.2$, $b = 0.1$ and $\tau = 20$. The time series are sampled for a period of eight seconds. In this case, the contaminated input vector $\boldsymbol{x}(i) = [u(i-10) + \eta(i-10), u(i-9) + \eta(i-9), ..., u(i-1) + \eta(i-1)$ (i.e., the filter length is 10) was used to predict the desired data $u(i)$ at time i.

Similar to case-1, Table 5 and Table 6 summarize part of the evaluation results on case-2 with various values of δ and β, respectively. In this case, we choose $\delta = 0.6$ and $\beta = 1.74$. Figure 5 depicts the learning curves for various adaptive algorithms. Again, the conventional linear LMS algorithm cannot converge well during the adaptation processes. However, our proposed KMSLMS still significantly outperforms these related works in this case. The resulting testing MSE improvements over the KLMS are 7.39 dB on average (from 0.0263 to 0.0048). Compared with the KMSLMS algorithm, the NICE-KMSLMS algorithm has an insignificant performance loss (0.59 dB, from 0.0055 to 0.0048) but can lower the mean size of clusters by 54.55% (from 2364.6 to 1074.6) on average. Table 7 summarizes the comparison results.

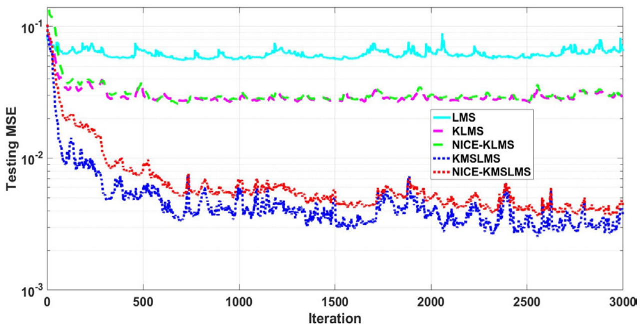


Figure 5. The learning curves comparisons in case-2 with $\mu = 0.23$, $A = 0.8$, $d_c = 1.53$, and $\gamma = 2.0$

Table 5. The testing MSE of NICE-KMSLMS on case-2 with various δ

δ	0.1	0.3	0.5	0.7	0.9
Testing MSE (x10^{-3})	15.7	14.0	7.2	9.2	22.4

Table 6. The testing MSE of NICE-KMSLMS on case-2 with various β

β	0.5	1.5	2.5	3.5	4.5
Testing MSE (x10^{-3})	17.3	6.2	14.8	19.2	22.7

Table 7. Comparison results for case-2

Algorithm	Mean size	Time	Testing MSE
LMS	-	7.2s	-
KLMS	1769.2	26.2s	0.0263
NICE-KLMS	723.8	12.1s	0.0271
KMSLMS	2364.6	31.9s	0.0048
NICE-KMSLMS	1074.6	14.7s	0.0055

5 Conclusions

In this paper, we devise a modified sign (MS) function and apply it to the KLMS algorithm to combat the impact caused by the impulse noise during the adaptation process.

Moreover, the MS function makes a better cluster formation in the presence of the impulse noise in the NICE-KMSLMS algorithm. Simulation results show that our KMSLMS algorithm improves the averaged testing MSE by 2.32 dB and 7.39 dB for the nonlinear channel equalization and MG chaotic time series prediction problems, respectively. In addition, the combination of NICE and KMSLMS results in inconsiderable performance loss but gains a computational saving of about 55% in both problems. However, the limitation of this work is the lack of an adaptation process to determine the optimal parameters of the MS function. Our further work is to optimize the values of δ and β of the proposed MS function via systematic approaches.

Acknowledgments

This work was partly supported by the Ministry of Science and Technology of the Republic of China under Grant MOST 110-2221-E-197 -022 and 111-2221-E-197 -013.

References

[1] E. M. Ar-Reyouchi, An Efficient Filtering Technique of Gap Filler System for Multi-standards Digital Terrestrial TV in an SFN, *Journal of Internet Technology*, Vol. 21, No. 5, pp. 1325-1334, September, 2020.

[2] Y.-R. Chien, Iterative Channel Estimation and Impulsive Noise Mitigation Algorithm for OFDM-Based Receivers With Application to Power-Line Communications, *IEEE Transactions on Power Delivery*, Vol. 30, No. 6, pp. 2435-2442, December, 2015.

[3] Y.-R. Chien, S. S.-D. Xu, S.-H. Lu, Cyclostationary Impulsive Noise Mitigation Algorithm for Narrowband Powerline Communications, *Journal of the Franklin Institute*, Vol. 357, No. 1, pp. 687-703, January, 2020.

[4] F. N. Buzylev, S. N. Shcherbakova, Using of Genetic Algorithms for Adaptive Filter Selection in Special-Purpose Systems, *Russian Journal of General Chemistry*, Vol. 91, No. 12, pp. 2734-2736, December 2021.

[5] G. X. Tong, G. N. Wu, J. Tan, N. X. Xiong, A. V. Vasilakos, A Digital Noise Reduction Scheme in Communication Systems for Internet of Things, *Journal of Internet Technology*, Vol. 17, No. 5, pp. 879-887, September, 2016.

[6] P. Śliwiński, A. Marconato, P. Wachel, G. Birpoutsoukis, Non-linear System Modeling Based on Constrained Volterra Series Estimates, *IET Control Theory & Applications*, Vol. 11, No. 15, pp. 2623-2629, October, 2017.

[7] Q. Chen, L. Li, A Deep Learning Based Equalization Scheme for Bandwidth-compressed Non-orthogonal

Multicarrier Communication, *Journal of Internet Technology*, Vol. 22, No. 5, pp. 1001-1009, September, 2021.

[8] A. Zara, S. U. Rehman, F. Ahmad, S. Kouser, A. Pervaiz, Numerical Approximation of Modified Kawahara Equation Using Kernel Smoothing Method, *Mathematics and Computers in Simulation*, Vol. 194, pp. 169-184, April, 2022.

[9] S. Zhao, B. Chen, Z. Cao, P. Zhu, J. C. Principe, Self-organizing Kernel Adaptive Filtering, *EURASIP Journal on Advances in Signal Processing*, Vol. 2016, pp. 1-12, October, 2016.

[10] Y. R. Chien, C. H. Yu, Impulse-noise-tolerant data-selective LMS Algorithm, *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E105.A, No. 2, pp. 114-117, February, 2022.

[11] Y. Guo, W. Wen, P. Wu, M. Xia, A Sparsity Adaptive Algorithm to Recover NB-IoT Signal from Legacy LTE Interference, *IEEE Wireless Communications Letters*, Vol. 10, No. 12, pp. 2703-2707, December, 2021.

[12] K. Li, J. C. Príncipe, Transfer Learning in Adaptive Filters: The Nearest Instance Centroid-estimation Kernel Least-mean-square Algorithm, *IEEE Transactions on Signal Processing*, Vol. 65, No. 24, pp. 6520-6535, December, 2017.

[13] Y. Zou, S.-C. Chan, T.-S. Ng, Least Mean M-estimate Algorithms for Robust Adaptive Filtering in Impulse Noise, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 47, No. 12, pp. 1564-1569, December, 2000.

[14] P. Wen, J. Zhang, Robust Variable Step-size Diffusion Sign-error Algorithm over Adaptive Networks, *Circuits, Systems, and Signal Processing*, Vol. 39, No. 6, pp. 3007-3018, June, 2020.

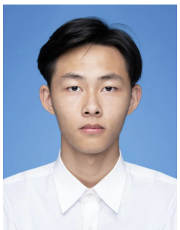
[15] Y. R. Chien, Variable Regularization Affine Projection Sign Algorithm in Impulsive Noisy Environment, *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E102.A, No. 5, pp. 725-728, May, 2019.

[16] W. Liu, P. P. Pokharel, J. C. Principe, The Kernel Least-mean-square Algorithm, *IEEE Transactions on Signal Processing*, Vol. 56, No. 2, pp. 543-554, February, 2008.

[17] L. Dang, B. Chen, S. Wang, Y. Gu, J. C. Principe, Kernel Kalman Filtering with Conditional Embedding and Maximum Correntropy Criterion, *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 66, No. 11, pp. 4265-4277, November, 2019.

[18] W. Liu, J. C. Principe, S. Haykin, *Kernel Least-Mean-Square Algorithm*, in: *Kernel Adaptive Filtering: A Comprehensive Introduction*, John Wiley & Son, 2010, ch. 2, pp. 27-68.

Biographies



Yijie Tang is currently working toward the B.Eng. degree in electronic information engineering with the College of Electronic and Information Engineering, Southwest University, Chongqing, China. His research interests include adaptive signal processing and array signal processing.

Guobing Qian received the Ph.D. degree in signal and information processing from the University of Electronic Science and Technology of China, Chengdu, China. He is currently with the College of Electronic and Information Engineering, Southwest University, Chongqing, China. His current research interests mainly include adaptive signal processing.

Wenqi Wu is currently working toward the B.Eng. degree in electronic information engineering with the College of Electronic and Information Engineering, Southwest University, Chongqing, China. Her research interests include signal processing and communication engineering.

Ying-Ren Chien received the Ph.D. degree in communication engineering from National Taiwan University, Taipei, Taiwan, in 2009. Since 2012, he has been with the Department of Electrical Engineering, National Ilan University, Yilan, Taiwan, where he is currently a Full Professor. His research interest focuses on adaptive signal processing theory.