New approaches for the Kernel-based Adaptive Filter with Epanechnikov kernel

Lucas H. Gois, Denis G. Fantinato and Aline Neves

***Abstract*— Kernel Adaptive Filtering has proven to be an effective solution for nonlinear channel equalization, outperforming traditional linear filters. In this context, we highlight the Kernel Maximum Correntropy (KMC) filter, in which the use of the Epanechnikov kernel has shown to be a promising approach. However, this method presents two drawbacks: the numerical instability that caused divergence and the need of training constantly. In this paper, to address the first problem, a sliding window procedure was proposed. To address the second problem, a decision direct mode was implemented. Both versions showed a desirable behavior, with no performance loss. A study in noisy scenarios was also considered.**

***Keywords*— Channel Equalization, Kernel Adaptive Filter, Correntropy, Epanechnikov kernel, Sliding Window, Decision-Directed Algorithm**

I. Introduction

Adaptive filtering plays a vital role in various signal processing applications, including channel equalization. While linear algorithms, such as the Least Mean Squares (LMS) [1], have been widely used due to their quick convergence and accuracy, their effectiveness is limited in nonlinear systems [2]. To address this limitation, kernel adaptive filtering (KAF) algorithms have gained popularity for their ability to handle nonlinear problems through mapping input data to higher-dimensional spaces (referred to as reproducing kernel Hilbert space - RKHS) using kernel functions [3]. The Kernel Least-Mean-Square (KLMS) algorithm and Kernel Recursive Least Squares (KRLS) are well-known examples of this class of filter [4].

The choice of an appropriate cost function is crucial in kernel adaptive filtering. Traditional measures like Mean Squared Error (MSE) used in KLMS may yield poor performance in non-Gaussian scenarios [5]. To overcome such issue, Information Theoretic Learning (ITL) criteria, which capture higher-order statistics, have been employed [2], [6]. Among these criteria, the Maximum Correntropy Criterion (MCC) has gained attention due to its simplicity and robustness. The Kernel Maximum Correntropy (KMC) algorithm combines MCC with KAF and has shown promising performance, particularly in impulsive noise environments [7], [8].

In [7], a study considering different kernel functions for calculating the correntropy measure in the context of KAF was presented. The paper considered the classical Gaussian kernel and proposed the use of the Epanechnikov kernel. The later has been considered in several works, showing good performance [9]–[11]. However, even though the KMC with Epanechnikov kernel (KMC-EPA) performed well, specially in nonlinear channel scenarios, the resulting algorithm had a drawback: it suffered from a numerical instability that led it to diverge after converging through a certain number of iterations. Since the size of the KMC filter increases linearly with the number of training data, this paper proposes the use of a sliding window to reduce the complexity of the KMC-EPA algorithm and avoid its divergence. In [12], the authors proposed using this approach on the KRLS, resulting in the Sliding Window KRLS (SW-KRLS) algorithm and demonstrating good performance in a nonlinear Wiener system.

In addition, this paper also proposes the use of the decision-directed (DD) approach [13] with KAF filters. To the knowledge of the authors, such approach was not yet presented in the literature. In this technique, the algorithm is able to continue the process of adaptation of the filter in a blind fashion, after a short period of supervised training using the transmitted signal.

The structure of this paper is as follows: Section II provides an overview of the KMC and the KLMS algorithm. Section III discusses the algorithms using the sliding window. Section IV explores the algorithms with the DD approach. Section V presents the performance analysis of the algorithms in various equalization scenarios. Finally, Section VI concludes the work.

II. Foundations

In this section, we present the channel equalization problem using the Kernel Adaptive Filtering technique, along with the relevant algorithms discussed in the literature.

A. Channel Equalization Problem

Figure 1 depicts the block diagram of channel equalization performed by a Kernel Adaptive Filter. The objective is to recover the initially transmitted signal s_n. The algorithm updates the filter using the channel output $\mathbf{x}_n = h(\mathbf{s}_n)$ and the error, e_n, computed between s_n and the filter output y_n during the training phase. Once the training period concludes, the algorithm is able to switch to the Directed-Decision (DD) approach, where a symbol decision device receives y_n to reconstruct the original transmitted signal, producing the estimated signal $\hat{y}_n$. This estimated signal is then used to calculate the error $e_n = y_n - \hat{y}_n$.

Lucas H. Gois and Aline Neves, Federal University of ABC, Santo André - SP, e-mail: lucas.gois@aluno.ufabc.edu.br and aline.neves@ufabc.edu.br. Denis G. Fantinato, State University of Campinas, Campinas - SP, e-mail: denisf@unicamp.br

Further details about this approach will be explained in subsequent sections.

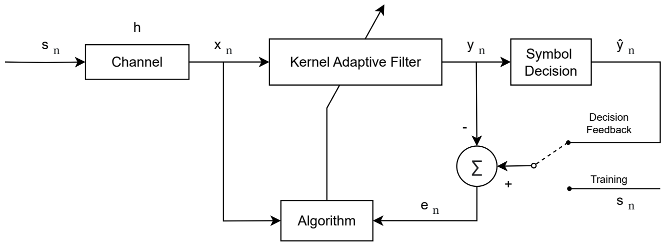

Fig. 1. Block Diagram of Communication System.

B. Maximum Correntropy Criterion

Considering the channel equalization context, the maximum correntropy criterion (MCC) aims to maximize the correntropy between the samples of the transmitted signal s_i and the estimated signal y_i at the output of the equalizer. The MCC cost function is given by [8]:

$$J_n = \frac{1}{N} \sum_{i=n-N+1}^{n} \kappa_\sigma(s_i, y_i), \tag{1}$$

where y_i is the filter output, N is the size of the training data, and $\kappa(\cdot)$ denotes a symmetric positive definite kernel function.

Correntropy can be seen as a similarity measure between two random variables [14]. It is interesting in the equalization process since it is able to explore the temporal characteristics of the signal [14]. Mathematically, it is defined as:

$$V_\sigma(s, y) = E[\kappa_\sigma(s - y)], \tag{2}$$

where σ represents the kernel width and $E[\cdot]$ denotes the expectation operator. In the literature, the kernel is usually chosen as the Gaussian function:

$$\kappa_G(s_i, y_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(s_i - y_i)^2}{2\sigma^2}}, \tag{3}$$

but the Epanechnikov kernel [15], shown below, has also been recently used successfully in certain applications [9]–[11]:

$$\kappa_E(s_i, y_i) = \frac{3}{4\sigma}\left(1 - \left(\frac{s_i - y_i}{\sigma}\right)^2\right), -\sigma < s_i - y_i < \sigma. \tag{4}$$

C. Kernel Adaptive Filtering

Linear adaptive filters often struggle to achieve high performance in nonlinear systems. Therefore, kernel methods are a robust choice for this task due to their universal approximation and convex optimization capabilities [4]. According to Mercer's Theorem, kernel-induced mappings transform the input data x_i to a high-dimensional feature space $\mathbb{F}$, known as the reproducing kernel Hilbert space (RKHS) and denoted as $\varphi(x_i)$ in kernel adaptive algorithms [2], [4].

Additionally, it is viable to use the inner product of the transformed input data, which allows the application of proper linear operations. This approach is known as "kernel trick" [4]. As per the representer theorem [4], [8], it is possible to express the system output in terms of the data by the following equation:

$$y = \sum_{i \in N} c_i \kappa_\sigma(x_i, .), \tag{5}$$

where c_i represents the weight coefficients obtained from the training data, and κ is a symmetric positive definite kernel function. The topology of the KAF algorithm, as discussed in [4], is similar to a growing radial basis function (RBF) network that expands linearly with the size of training data. The parameters of each node are computed within the RKHS using the gradient ascent approach. The coefficients of the KAF filter, $\mathbf{\Omega}$, can be iteratively updated:

$$\mathbf{\Omega}_n = \mathbf{\Omega}_{n-1} + \mu \nabla J_n, \tag{6}$$

where μ is the step size. Kernel width and step size effects on the algorithm have been studied in [7], [11].

D. KMC with the Gaussian kernel

Using the MCC cost function (1) in (6) with the paired sample $\{s_n, \varphi(x_n)\}$, the adaptive filter weights $\mathbf{\Omega}$ can be computed using the Gaussian kernel (3) and the stochastic gradient approximation [8]:

$$\mathbf{\Omega}_{n+1} = \mathbf{\Omega}_n + \mu \frac{\partial \kappa_G(s_n, \mathbf{\Omega}_n^T \varphi_n)}{\partial \mathbf{\Omega}_n} = \mu \sum_{i=1}^{n} \exp\left(\frac{-e_i^2}{2\sigma^2}\right) e_i \varphi_i, \tag{7}$$

where φ_i is a simplified notation for $\varphi(x_i)$, and $e_n = s_n - \mathbf{\Omega}_n^T \varphi_n$. The system output is now obtained using the "kernel trick", which is expressed in terms of the inner product between the new input and the previous inputs weighted by prediction errors [3], [4], [8]:

$$\begin{aligned} y_{n+1} &= \mathbf{\Omega}_{n+1}^T \varphi_{n+1} \\ &= \mu \sum_{i=1}^{n} \exp\left(\frac{-e_i^2}{2\sigma^2}\right) e_i \varphi_i^T \varphi_{n+1} \\ &= \mu \sum_{i=1}^{n} \exp\left(\frac{-e_i^2}{2\sigma^2}\right) e_i \kappa_G(x_i, x_{n+1}). \end{aligned} \tag{8}$$

The algorithm, initially named Kernel Maximum Correntropy (KMC) [8], will be referred to as KMC-GAU in this work since it uses the Gaussian kernel.

E. KMC with the Epanechnikov kernel

As proposed in [7], the stochastic gradient, given by (5), can be computed using the Epanechnikov kernel (4) as follows:

$$\mathbf{\Omega}_{n+1} = \mathbf{\Omega}_n + \mu \frac{\partial \kappa_E(s_n, \mathbf{\Omega}_n^T \varphi_n)}{\partial \mathbf{\Omega}_n} = \mu \frac{3}{2\sigma^3} \sum_{i=1}^{n} e_i \varphi_i. \tag{9}$$

Similar to the KMC-GAU (8), the "kernel trick" is used to obtain the following system output:

$$y_{n+1} = \mathbf{\Omega}_{n+1}^T \varphi_{n+1} = \mu \frac{3}{2\sigma^3} \sum_{i=1}^{n} e_i \kappa_E(x_i, x_{n+1}). \tag{10}$$

The algorithm with the Epanechnikov kernel will be called KMC-EPA.

III. Sliding Window

In [7], it was observed that the KMC-EPA algorithm tends to diverge after a certain number of iterations, indicating potential numerical instability. This issue can arise due to the fact that the size of the network increases with the number of training data [4]. To address this problem, regularization techniques are commonly applied to control the size and complexity of the RBF network [4].

One possible solution to mitigate this instability is to employ a sliding window technique, which consists in taking only the last N_w of the training data into account for the filter output calculation. In [12], the authors proposed a similar approach applied to the KRLS. In this case, the size of the RBF network would be limited to N_w, also known as the window size. The pseudo-code of this approach applied to the KMC-EPA can be found in Algorithm 1, named KMC-EPASW.

Algorithm 1 KMC-EPA with Sliding Window

for $i = N_w + 1$ **to** N **do**
  for $j = i - N_w$ **to** $i - 1$ **do**
    if $|x(j) - x(i)| < \sigma$
      $y(i) = y(i) + \mu\left(\frac{3}{2\sigma^3}\right) e(j)\, \kappa_E(x(j) - x(i))$
    else
      $y(i) = y(i) + 0$**;**
    end if
  end for
  $e(i) = s(i) - y(i)$
end for

This approach was also applied to the KMC-GAU that will be named KMC-GAUSW.

IV. Decision-Directed Algorithm

The decision-directed (DD) algorithm can be described as a sort of "modified Wiener criterion" [13], in which the desired symbol s_n used during the training period is replaced by the estimate given by a decision-device, that receives the filter output y_n and approximates the symbols using the constellation of the transmitted signal [1]. The idea behind this method is to give a practical solution to allow the filter adaptation after the training period, where the transmitted signal is no longer known at the receiver. Thus, adaptation continues in a blind mode.

Considering that the training period is capable of bringing the equalizer to a situation where the equalizer output is sufficiently close to the original transmitted sequence, also known as an opened eye pattern, it is possible to guide the adaptation process using the estimated signal given by the decision device [13]. By calculating the error e_i using the estimated signal, we are able to modify the algorithm presented by switching between training mode and DD mode:

$$e_{predict} = \hat{y}_n - \mathbf{\Omega}_n^T \varphi_n = \hat{y}_n - y_n. \tag{11}$$

where $\hat{y}_n$ is the estimated signal obtained at the decision device output (see Fig. 1). Applying (11) to the KMC-EPA algorithm, we obtain the KMC-EPADD, detailed in Algorithm 2.

Algorithm 2 KMC-EPA Decision-Directed

for $i = N + 1$ **to** $TotalSamples$ **do**
  for $j = 1$ **to** $i - 1$ **do**
    if $|x(j) - x(i)| < \sigma$
      $y(i) = y(i) + \mu\left(\frac{3}{2\sigma^3}\right) e_{predict}(j)\, \kappa_E(x(j) - x(i))$
    else
      $y(i) = y(i) + 0$
    end if
  end for
  $\hat{y}(i) = DecisionDevice(y(i))$
  $e_{predict}(i) = \hat{y}(i) - y(i)$
end for

In parallel, (11) was applied to KMC-GAU leading to the KMC-GAUDD.

V. Results

In this section, we will examine the performance of the KMC-EPASW and the KMC-EPADD algorithms in linear and nonlinear scenarios, based on [7]. For comparison, we include the KMC-GAUSW and KMC-GAUDD, which use the classical Gaussian kernel, and also the KLMS [3], the first kernel-based adaptive filter presented in the literature. As developed in [7], KLMS with the Epanechnikov kernel is identical to the KMC-EPA and for this reason, KLMS is only considered with the Gaussian kernel. We also considered its sliding window version, KLMS-GAUSW and its DD version, KLMS-GAUDD. The parameters were chosen considering the best performance varying σ in $[0.1, 5]$ and μ in $[0.001, 0.9]$. The performance will be evaluated by measuring the Mean Square Error (MSE).

A. Sliding Window Simulations

First, we will use a linear scenario with a correlated transmitted signal, which consists of a Binary Phase Shift Keying (BPSK) signal filtered by $F(z) = 1 + 0.5z^{-1}$, giving s_n, and distorted by the channel $H(z) = 0.2 + 1z^{-1} + 0.4z^{-2}$ accompanied by impulsive noise [8], whose probability density function is described by the following equation [7]:

$$p_{noise} = 0.9\mathcal{N}(0, \sigma_1) + 0.1\mathcal{N}(0, \sigma_2), \tag{12}$$

with $\sigma_2 = 0.8$ and σ_1 adjusted to obtain a resulting SNR of 20 dB. As in [7], since H(z) is a nonminimum-phase channel, the error is calculated with a 1-sample delay in s_n to improve the performance of the three algorithms.

In Fig. 2, we have the results of an average of 10000 simulations for the algorithms KMC-GAUSW, KMC-EPASW and KLMS-GAUSW in the linear scenario. For KMC-GAUSW the parameters used were $\mu = 0.9$ and $\sigma = 1$; for KMC-EPASW $\mu = 0.02$ and $\sigma = 0.5$ and for KLMS-GAUSW, $\mu = 0.9$ and $\sigma = 1$. The three algorithms used window size, $N_w = 200$. Analysing Fig. 2, it is possible to notice that the three algorithms presented a singular behavior in which the convergence curves tend to oscillate in a frequency equal to the window size. After a certain number of samples, the amplitude of the oscillation decays, and the algorithms converge. This behavior will be explored with further details in the future. In general, the KMC-GAUSW shows the best performance of all three algorithms in this scenario. Despite the oscillation, the attaint MSE level is similar to the one obtained in [7] for the same scenario, which means that the algorithms do not lose performance with the sliding windows approach. Furthermore, the KMC-EPASW does not diverge and the algorithms present a lower computational cost since the size of the resulting RBF network is constant.

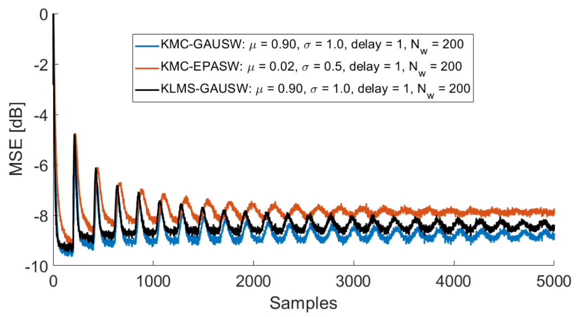

Fig. 2. Convergence curve in a linear channel with impulsive noise using a correlated signal.

In Fig. 3, we present the MSE value after convergence as a function of the window size, N_w. Results show an average of 1000 simulations. The parameters used were the same ones as Fig. 2. In this case, the MSE level achieved by KMC-GAUSW and KLMS-GAUSW are stable for $N_w > 200$. On the other hand, KMC-EPASW starts to present instability for $N_w > 900$, diminishing its performance. Thus, smaller window sizes should be chosen.

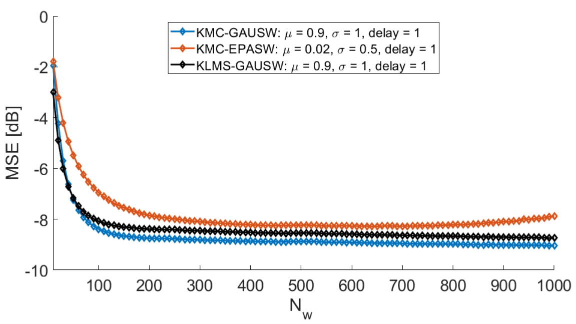

Fig. 3. MSE by the window size in a linear channel with impulsive noise using a correlated signal.

For the nonlinear scenario, we will use a binary signal s_n, in order to compare with the result found in [7]. The nonlinear channel model is defined by $z_n = s_n + 0.2s_{n-1}$, $x_n = z_n - 0.9z_n^2 + \nu_\sigma$, where ν_σ is an additive white Gaussian noise (AWGN) considering an SNR of 20 dB. It is important to point out that, in this scenario, the error is calculated without delay [7].

The results are shown in Fig. 4, considering an average of 10000 simulations. For KMC-GAUSW the parameters used were $\mu = 0.5$ and $\sigma = 2$; for KMC-EPASW $\mu = 0.05$ and $\sigma = 1$ and for KLMS-GAUSW, $\mu = 0.4$ and $\sigma = 2$. $N_w = 200$ for the three algorithms. We can note that KMC-GAUSW and KLMS-GAUSW converge to the same MSE level, whilst KMC-EPASW shows the best performance among the three algorithms. In terms of MSE level, the three algorithms had a similar result when compared to the ones shown in [7].

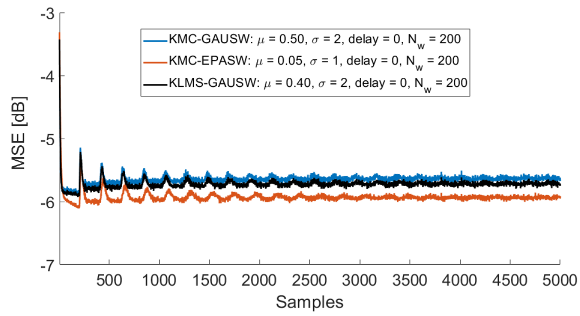

Fig. 4. Convergence curve in a nonlinear channel with additive noise using a binary signal.

Fig. 5 considers the MSE as a function of the window size, N_w, for the nonlinear scenario. An average of 1000 simulations was considered. These results were obtained using the same parameters as Fig. 4. We can note that by varying the N_w the MSE threshold of the three algorithms are stable for $N_w > 300$ approximately, even though KMC-EPASW performs better since $N_w = 80$.

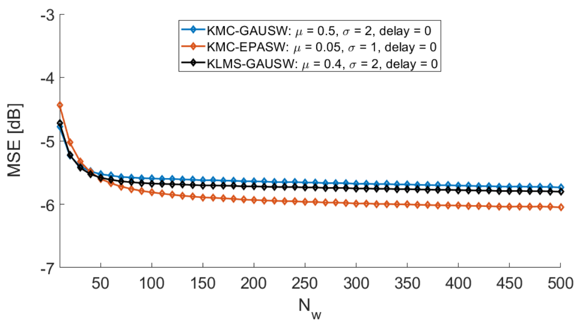

Fig. 5. MSE by the window size in a nonlinear channel with additive noise using a binary signal.

B. Decision-Directed Mode Simulations

Using the same linear scenario with a correlated s_n, we simulated the KMC-GAUDD, KMC-EPADD and KLMS-GAUDD. The results can be found in Fig. 6 and are an average of 10000 simulations. The training stage considered 200 samples for the three algorithms and the algorithm changed to DD mode, the following parameters were used:

for KMC-GAUDD were $\mu = 0.9$ and $\sigma = 2$; for KMC-EPADD, $\mu = 0.01$ and $\sigma = 0.5$; and for KLMS-GAUDD, $\mu = 0.7$ and $\sigma = 2$. We also considered 500 training samples for the KMC-EPADD in order to improve its performance, using $\mu = 0.9$ and $\sigma = 3.1$. Thus KMC-EPADD demands more training to converge to a similar MSE threshold as the other algorithms.

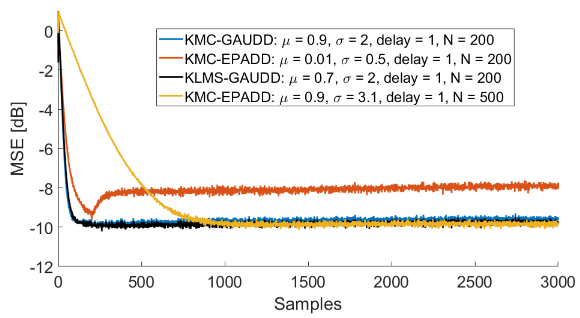

Fig. 6. Convergence curve in a linear channel with impulsive noise using a correlated signal.

Employing the nonlinear scenario used for Fig. 4 and 5, and a BPSK transmitted signal, the results are shown in Fig. 7, considering an average of 10000 simulations. For KMC-GAUDD the parameters used were $\mu = 0.9$ and $\sigma = 1$; for KMC-EPADD $\mu = 0.002$ and $\sigma = 0.9$ and for KLMS-GAUDD, $\mu = 0.7$ and $\sigma = 1$. For training, 200 samples were set and the rest for decision-directed mode. In Fig. 7, we can notice that KMC-GAUDD and KLMS-GAUDD achieve the same MSE threshold, and KMC-EPADD presents the lowest MSE level even though it takes longer to converge.

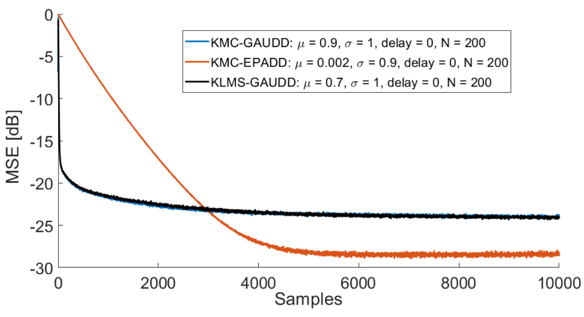

Fig. 7. Convergence curve in a nonlinear channel with additive noise using a BPSK signal.

VI. Conclusions

The Kernel Maximum Correntropy algorithm is an efficient and robust algorithm that deals particularly well with impulsive noise and nonlinear channel equalization. Its implementation with the Epanechnikov kernel improved the algorithms performance in certain situations, however, this new algorithm suffered from numerical instability that led it to diverge after a certain number of iterations. In this work, we proposed the use of sliding windows to reduce the complexity of the KMC-EPA and avoid its divergence. Simulations in linear and nonlinear scenarios showed that the algorithms did not lose performance in this new approach, with KMC-EPASW performing better in the later. Another approach to deal with numerical instability is the use of dictionaries [4], which will be further studied in the future.

In addition, we proposed the use of a decision-directed mode with the KAF filters enabling the adaptation in blind mode after a training stage. The three algorithms were first simulated using a correlated signal with a linear channel and impulsive noise, achieving the same MSE level. In a nonlinear scenario with additive white Gaussian noise, the KMC-EPADD presented the best performance in terms of the MSE threshold.

Acknowledgements

The authors would like to thank the São Paulo Research Foundation (Projeto BIOS Fapesp Processo No 2020/09838-0) and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES - Finance Code 001).

References

[1] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall information and system sciences series, Prentice Hall, 1996.

[2] Z. Wu, J. Shi, X. Zhang, W. Ma, and B. Chen, “Kernel recursive maximum correntropy,” *Signal Processing*, vol. 117, pp. 11–16, 2015.

[3] W. Liu, P. P. Pokharel, and J. C. Principe, “The kernel least-mean-square algorithm,” *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 543–554, 2008.

[4] J. C. Príncipe, W. Liu, and S. Haykin, *Kernel adaptive filtering: a comprehensive introduction*. John Wiley & Sons, 2011.

[5] Y. He, F. Wang, J. Yang, H. Rong, and B. Chen, “Kernel adaptive filtering under generalized maximum correntropy criterion,” in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 1738–1745, IEEE, 2016.

[6] W. Liu, P. P. Pokharel, and J. C. Principe, “Correntropy: Properties and applications in non-gaussian signal processing,” *IEEE Transactions on signal processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

[7] L. H. Gois, D. G. Fantinato, and A. Neves, “A comparison between kernel-based adaptive filters including the epanechnikov function,” *XL Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 2022.

[8] S. Zhao, B. Chen, and J. C. Principe, “Kernel adaptive filtering with maximum correntropy criterion,” in *The 2011 International Joint Conference on Neural Networks*, pp. 2012–2017, IEEE, 2011.

[9] C. P. Moraes, D. G. Fantinato, and A. Neves, “An Epanechnikov kernel based method for source separation in post-nonlinear mixtures,” *XXXVII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, vol. 1, 2019.

[10] C. P. Moraes, D. G. Fantinato, and A. Neves, “Epanechnikov kernel for PDF estimation applied to equalization and blind source separation,” *Signal Processing*, vol. 189, p. 108251, 2021.

[11] L. Gois, D. Fantinato, R. Suyama, and A. Neves, “Relationship between criteria based on correntropy and second order statistics for equalization of communication channels,” *IEEE Signal Processing Letters*, 2022.

[12] S. Van Vaerenbergh, J. Via, and I. Santamaria, “A sliding-window kernel rls algorithm and its application to nonlinear channel identification,” in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5, pp. V–V, 2006.

[13] C. C. Cavalcante, R. F. Colares, and P. C. Barbosa, “Telecommunications: Advances and trends in transmission, networking and applications,” *Universidade de Fortaleza, Brazil*, 2006.

[14] I. Santamaría, P. P. Pokharel, and J. C. Principe, “Generalized correlation function: definition, properties, and application to blind equalization,” *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2187–2197, 2006.

[15] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.