A Comparison Between Kernel-based Adaptive Filters Including the Epanechnikov Function

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

***Abstract*— Kernel Adaptive Filtering is an effective solution for nonlinear channel equalization, offering remarkable results in scenarios where linear filters often fail. In this context, the Kernel Maximum Correntropy (KMC) is an efficient and resilient technique. In most cases, the Gaussian kernel is used to calculate correntropy. In this article, we propose to use the Epanechnikov kernel to estimate correntropy and analyze its performance. The filter performance is compared to the KMC with Gaussian kernel and also to the Kernel Least-Mean-Square algorithm.**

***Keywords*— Channel Equalization, Kernel Adaptive Filter, Correntropy, Epanechnikov kernel**

I. INTRODUCTION

Adaptive filtering is used in a broad range of applications in the signal processing field, including channel equalization. Because of its quick convergence and accuracy, the least mean squares (LMS) is one of the most used methods [1]. However, the potential of linear algorithms is limited, since many real-world problems demand more expressive hypothesis spaces than linear functions [1]. Since this approach is based on a linear model, it performs poorly when the underlying system is highly nonlinear [2]. In order to solve this problem, numerous methods have been proposed, with the kernel adaptive filtering (KAF) algorithms becoming increasingly popular in recent years for their universal approximation capability and convex optimization nature [1]. Furthermore, input data can be implicitly/nonlinearly mapped to a higher-dimensional space (referred to as reproducing kernel Hilbert space - RKHS) using kernel functions, where appropriate linear operations can be performed [3]. In this context, the algorithm Kernel Least-Mean-Square (KLMS) is a well-known example.

Choosing the right cost function is critical in kernel adaptive filtering. For example, KLMS uses the Mean Squared Error (MSE) as a cost function. However, MSE-based kernel adaptive filters will have poor performance in a non-Gaussian scenario [4]. To work around this issue, criteria based on Information Theoretic Learning (ITL) have been used to replace traditional second-order statistical measures like MSE. Capturing higher-order statistics may provide potentially significant performance improvements in the adaptation [2], [5]. Unlike the MSE-based criterion, that uses error energy as the cost function, ITL uses the estimated probability density function (pdf) of the data, which is computed using the Parzen kernel estimator [6].

In this context, the maximum correntropy criterion (MCC) has gotten a lot of attention in recent years because of its simplicity and robustness [2], [4], [7]. Correntropy is a generalized correlation measure induced by a kernel function that is capable of extracting various statistical moments from signals and explore their temporal structure. Face recognition [8], categorization [9], and robust principal component analysis [10] are just a few of the situations where MCC has been applied effectively. For that reason, the Kernel Maximum Correntropy (KMC) algorithm was proposed combining MCC with kernel adaptive filtering [7], demonstrating a good performance in impulsive noise environment.

Due to its smoothness and rigorous positive definition, the Gaussian kernel is commonly used as the kernel function in the estimation of correntropy [7], [8], [11]. However, is this a good decision, given the availability of alternative kernel functions? In [4], the authors propose using the generalized Gaussian density (GGD) function as the kernel, implying that the Gaussian kernel is not always the best approach. In pdf estimation through algorithms based on Parzen windows, the Epanechnikov kernel has outperformed the Gaussian kernel in some situations [12], [13]. To the best of our knowledge, there is no detailed investigation of kernel adaptive filtering algorithms within this context in the literature. As a result, given the work done in [13] and [14], it becomes interesting to develop the Kernel Maximum Correntropy using the Epanechnikov kernel.

Thus, the aim of this article is to analyze the performance of this new criteria, and compare it with the results obtained using the Gaussian kernel and the KLMS algorithm in the channel equalization problem. To this end, the work was structured as follows: Section II presents the maximum correntropy criterion, its kernel-based version and also the KLMS. The new proposed algoritm, KMC with the Epanechnikov kernel, is discussed in Section III. The relationship between the algorithm using the Epanechnikov kernel and KLMS is exhibit in Section IV. Section V shows the performance of the algorithms in the equalization of different scenarios. Finally, the conclusions of the work are presented in Section VI.

II. FOUNDATIONS

In this section we will first present the channel equalization problem and the well known Maximum Corren-

Lucas H. Gois, Denis G. Fantinato and Aline Neves, Federal University of ABC, Santo André - SP, e-mail: lucas.gois@aluno.ufabc.edu.br and {denis.fantinato,aline.neves}@ufabc.edu.br.

tropy Criterion (MCC), together with the Kernel Adaptive Filters presented in the literature.

A. Channel Equalization Problem

In Fig. 1, a block diagram of the channel equalization performed by a Kernel Adaptive Filter is shown. In an effort to recover the initially transmitted signal d_n, the algorithm updates the KAF using the channel output $\mathbf{u}_n = \mathbf{h}_n^T \mathbf{s}_n$ and the error e_n, computed between d_n and the filter output y_n.

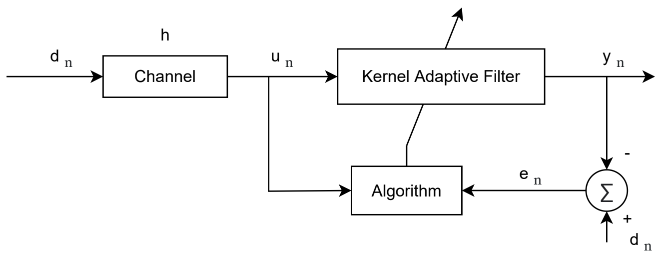

Fig. 1. Block Diagram of Communication System.

B. Maximum Correntropy Criterion

Correntropy, introduced initially in [15], is a generalized correlation measure that probabilistically estimates the similarity between two arbitrary random variables. This metric is linked to the Parzen window method to calculate Rényi's quadratic entropy [16]. Correntropy has been applied to channel equalization because it can also exploit the temporal characteristics of the signal [15], making it suitable for dealing with correlated signals. The following equation can be used to obtain this measure:

$$V_\sigma(X,Y) = E[\kappa_\sigma(X-Y)], \tag{1}$$

where σ is the kernel width, X and Y are two arbitrary random variables, $E[\cdot]$ is the expectation operator, $\kappa(\cdot)$ is a symmetric positive definite kernel function. Correntropy is calculated using the Gaussian kernel in most works found in the literature [15], [17] :

$$\kappa_G(X,Y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X-Y)^2}{2\sigma^2}}. \tag{2}$$

One of the most important parameters in estimating correntropy is σ, commonly known as kernel size or width, which influences the performance of the criterion in a significant way. It is important to mention that using the Gaussian kernel implies using all even order statistical moments of the signal [14]. The measure presented in (1) can be estimated using a time average of N signal samples of a discrete-time stochastic process:

$$\hat{V}_{N,\sigma}(X,Y) = \frac{1}{N}\sum_{i=1}^{N} \kappa_\sigma(x_i - y_i) \tag{3}$$

The maximum correntropy criterion (MCC), in this context, aims to maximize the correntropy between the transmitted signal d_i and the estimated signal y_i at the output of the equalizer, leading to the following cost function:

$$J_n(\mathbf{w}) = \frac{1}{N}\sum_{i=n-N+1}^{n} \kappa_\sigma(d_i, y_i) \tag{4}$$

where $\mathbf{w}$ is the set of filter weights and $y_i = \mathbf{w}^T u_i$ is the filter output. The gradient ascent approach may be used to update the equalizer coefficients based on the cost function given by (4).

The MCC method is similar to the LMS algorithm in terms of computational simplicity [7]. This criterion is a robust statistical approach due to the smooth dependence of correntropy on the kernel bandwidth. Experiments have shown that MCC has an advantage in linear adaptive filters when compared to other criteria [7].

C. KMC with Gaussian Kernel

Linear adaptive filters cannot achieve high performance if the mapping between d and u is nonlinear. For that reason, kernel methods are a strong choice for this task due to their universal approximation and convex optimization capabilities [1]. According to the Mercer's Theorem, the kernel-induced mapping transforms the input data u_i to a high-dimensional feature space $\mathbb{F}$ as $\varphi(u_i)$ in kernel adaptive algorithms [1], [2]. This feature space is known as reproducing kernel Hilbert space (RKHS). Furthermore, a linear model is also constructed in the RKHS to compute the system output using the transformed data [1], [7]. As discussed in [1], the KAF algorithm creates a growing radial-basis function (RBF) network that increases linearly with the number of training data. Following the representer theorem [1], [7], the linear filter weights applied in the feature space can be described by:

$$\mathbf{\Omega} = \sum_{i \in N} c_i < \varphi(u_i), . >= \sum_{i \in N} c_i \kappa_\sigma(u_i, .) \tag{5}$$

where c_i are weight coefficients obtained from the training data and κ is a symmetric positive definite kernel function. Then, using the gradient ascent approach, the coefficients can be updated iteratively:

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

where μ is the step size. With the new paired sample $\{\varphi(u_n), d_n\}$, the adaptive filter weights $\mathbf{\Omega}$ is computed using the MCC criterion and the stochastic gradient approximation [7]:

$$\begin{aligned}
\mathbf{\Omega}_0 &= 0 \\
\mathbf{\Omega}_{n+1} &= \mathbf{\Omega}_n + \mu \frac{\partial \kappa_G(d_n, \mathbf{\Omega}_n^T \varphi_n)}{\partial \mathbf{\Omega}_n} \\
&= \mathbf{\Omega}_n + \mu \exp\left(\frac{-e_n^2}{2\sigma^2}\right) e_n \varphi_n \\
&= \mathbf{\Omega}_{n-1} + \mu \sum_{i=n-1}^{n} \exp\left(\frac{-e_i^2}{2\sigma^2}\right) e_i \varphi_i \\
&\cdots \\
&= \mu \sum_{i=1}^{n} \exp\left(\frac{-e_i^2}{2\sigma^2}\right) e_i \varphi_i
\end{aligned} \tag{7}$$

where φ_i is a simplified notation for $\varphi(u_i)$, $e_n = d_n - \mathbf{\Omega}_n^T \varphi_n$ and κ_G is the Gaussian kernel. The system output is now obtained using the "kernel trick", which can be expressed in terms of the inner product between the new input and the previous inputs weighted by prediction errors [1], [3], [7]:

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

This algorithm, named Kernel Maximum Correntropy (KMC) [7], will be addressed as KMC-GAU since it uses the Gaussian kernel.

D. KLMS

The Kernel Least-Mean-Square (KLMS) is one of the most popular kernel adaptive filtering methods due to its robustness and simplicity [7]. Since it is based on the LMS algorithm, the KLMS uses the same criterion based on minimizing the MSE between the desired signal and the filter output [3]:

$$J(\mathbf{w}) = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (d_i - \mathbf{w}^T \mathbf{u}_i)^2 \tag{9}$$

where N is the size of training data. Then, transforming the input into a high dimensional feature space, the adaptive filter weights can be calculated using the gradient based algorithm [3]:

$$\begin{aligned}
\mathbf{\Omega}_0 &= 0 \\
e_n &= d_n - \mathbf{\Omega}_n \varphi(u_n) \\
\mathbf{\Omega}_n &= \mathbf{\Omega}_{n-1} + \mu e_n \varphi_n \\
&= \mu \sum_{i=1}^{n} e_i \varphi_i
\end{aligned} \tag{10}$$

The system output, y, can be computed through the "kernel trick".

$$\begin{aligned}
y_{n+1} &= \mathbf{\Omega}_{n+1}^T \varphi_{n+1} \\
&= \mu \sum_{i=1}^{n} e_i \varphi_i^T \varphi_{n+1} \\
&= \mu \sum_{i=1}^{n} e_i \kappa_\sigma(u_i, u_{n+1})
\end{aligned} \tag{11}$$

where $\kappa(\cdot)$ is a symmetric positive definite kernel function. It is important to point out that the kernel used by the KLMS in this work is the Gaussian kernel. Section IV will explore such issue in more details.

III. KMC WITH THE EPANECHNIKOV KERNEL

The Epanechnikov kernel has been used in numerous applications involving algorithms based on Parzen window [12], [13], demonstrating that it can be used to achieve good results. According to the study carried out in [18], using the average mean integrated squared error (AMISE) of the estimation of pdfs, the Epanechnikov kernel is considered optimal. This kernel is a quadratic polynomial function given by:

$$\kappa_E(X,Y) = \frac{3}{4\sigma}\left(1 - \left(\frac{X-Y}{\sigma}\right)^2\right), \; -\sigma < X - Y < \sigma \tag{12}$$

where σ is the kernel width. Outside of the support, $\kappa_E(X,Y) = 0$. Further information on the kernel characteristics can be found in [18].

Considering the stochastic gradient given by (7) and using the Epanechnikov kernel (12), the gradient is given by:

$$\begin{aligned}
\mathbf{\Omega}_{n+1} &= \mathbf{\Omega}_n + \mu \frac{\partial \kappa_E(d_n, \mathbf{\Omega}_n^T \varphi_n)}{\partial \mathbf{\Omega}_n} \\
&= \mathbf{\Omega}_n + \mu \frac{3}{2\sigma^3} e_n \varphi_n \\
&= \mathbf{\Omega}_{n-1} + \mu \frac{3}{2\sigma^3} \sum_{i=n-1}^{n} e_i \varphi_i \\
&\cdots \\
&= \mu \frac{3}{2\sigma^3} \sum_{i=1}^{n} e_i \varphi_i
\end{aligned} \tag{13}$$

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

$$\begin{aligned}
y_{n+1} &= \mathbf{\Omega}_{n+1}^T \varphi_{n+1} \\
&= \mu \frac{3}{2\sigma^3} \sum_{i=1}^{n} e_i \varphi_i^T \varphi_{n+1} \\
&= \mu \frac{3}{2\sigma^3} \sum_{i=1}^{n} e_i \kappa_E(u_i, u_{n+1})
\end{aligned} \tag{14}$$

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

IV. Comparison Between Algorithms

We start by comparing the algorithm KMC implemented with the Epanechnikov kernel (KMC-EPA) with the KLMS. Observing equations (14) and (11), it is possible to notice that both equations are similar. Considering that the coefficient $\frac{3}{2\sigma^3}$ can be incorporated to the step size and that the KLMS is using the Epanechnikov kernel, both equations become identical:

$$y_{KMC-EPA} \approx y_{KLMS-EPA}$$
$$\mu\frac{3}{2\sigma^3}\sum_{i=1}^{n} e_i\kappa_E(u_i,u_{n+1}) \approx \mu\sum_{i=1}^{n} e_i\kappa_E(u_i,u_{n+1}) \qquad (15)$$

Thus, using the Epanechnikov kernel, both algorithms lead to exactly the same equations, i.e., maximising the correntropy of the error or minimizing the mean-square-error are equivalent. This result comes from the fact that the Epanechnikov kernel is a second-order polynomial (12). Thus, using it to estimate correntropy is equivalent to obtain the mean square error. In addition, the Epanechnikov kernel is given by a constant minus the squared error, what explains the fact that, even though one criterion is given by a maximization and the other is a minimization, both result in the same expression.

For this reason, we will only consider the analysis of the KMC-EPA (and not the KLMS-EPA). The KLMS algorithm will always be computed using the Gaussian kernel.

V. Simulation Results

In this section, we will analyze the performance of the KMC-EPA algorithm in linear and nonlinear scenarios. We compare it with KMC-GAU and KLMS, the latter due to its popularity and robustness. After varying their parameters and obtaining the ones that led to the best result, the performance will be evaluated by measuring the MSE and speed of convergence. In order to obtain the best parameters, the kernel width was varied between $0.1 \le \sigma \le 5$ and the step size between $0.001 \le \mu \le 0.9$.

A. Linear Channel Equalization

First, we will use a Binary Phase Shift Keying (BPSK) signal d_n distorted by the channel $H(z) = 0.2+1z^{-1}+0.4z^{-2}$ in the presence of impulsive noise [7], whose probability density function is given by:

$$p_{noise} = 0.9\mathcal{N}(0,\sigma_1) + 0.1\mathcal{N}(0,\sigma_2) \qquad (16)$$

with $\sigma_2 = 0.8$ and σ_1 adjusted to obtain a resulting SNR of 20 dB. To improve the performance of the three algorithms on a nonminimum-phase channel, the error is calculated with 1-sample delay in d_n, i.e., $e_n = d_{n-1} - \mathbf{\Omega}_n\varphi(u_n)$. Simulations varying from 0 to 3 showed that this delay led to the best performance. In Fig. 2, we see the results obtained in an average of 1000 simulations for the algorithms KMC-GAU, KMC-EPA and KLMS.

For KMC-GAU the parameters used were $\mu = 0.9$ and $\sigma = 0.4$; for KMC-EPA $\mu = 0.01$ and $\sigma = 0.5$ and for KLMS, $\mu = 0.1$ and $\sigma = 0.3$. Analysing Fig. 2 it is possible to notice that all algorithms converge to a similar MSE threshold. In terms of speed, KMC-GAU converges slightly faster than the others. It is important to mention that the KMC-EPA tends to diverge after a certain number of iterations, which means that the algorithm may suffer from numerical instability. In this case, it is necessary to apply regularizations to control the size of the radial-basis function network created by the algorithm [1]. Such techniques are not applied in the present work and we intend to explore them in the future.

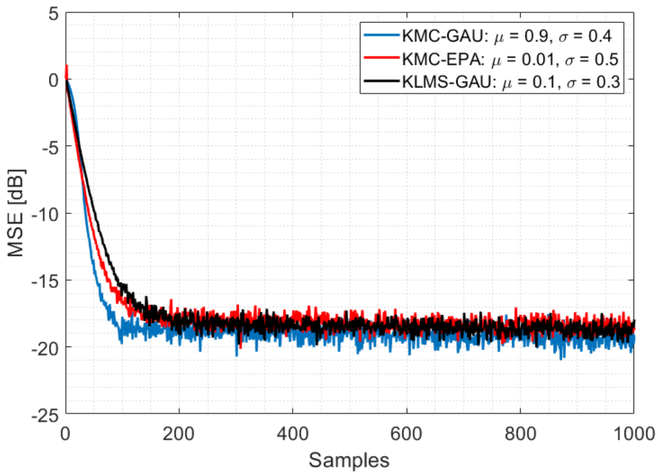

Fig. 2. Convergence curve in linear channel with impulsive noise using uncorrelated signal.

In Fig. 3, we used a temporally correlated d_n given by a signal BPSK filtered by $F(z) = 1+0.5z^{-1}$. The channel was kept the same. For KMC-GAU we used $\mu = 0.7$ and $\sigma = 1$; KMC-EPA, $\mu = 0.09$ and $\sigma = 1.4$ and KLMS, $\mu = 0.6$ and $\sigma = 1$. Fig. 3 shows an average of 1000 simulations. In this case KMC-GAU and KLMS converge at the same speed and to the same MSE level. Both KMC-GAU and KLMS are faster than KMC-EPA, although KMC-EPA achieves the lowest MSE level. This indicates that the Epanechnikov kernel might be able to preserve the temporal dependency more efficiently.

B. Nonlinear Channel Equalization

In this experiment, we will use a binary signal d_n in a nonlinear channel model, defined by $z_n = d_n + 0.2d_{n-1}$, $u_n = z_n - 0.9z_n^2 + \nu_\sigma$, where ν_σ is an additive white Gaussian noise (AWGN) . An SNR of 20 dB was considered. KMC-GAU parameters were defined as $\mu = 0.7$ and $\sigma = 1$; KMC-EPA used $\mu = 0.09$ and $\sigma = 1.4$ and KLMS, $\mu = 0.6$ and $\sigma = 1$. Fig. 4 represents an average of 1000 simulations. We can note that KMC-GAU and KMC-EPA converge to the same MSE level, and both performed better than the KLMS. Furthermore, KMC-EPA converges faster than the KMC-GAU, presenting the best performance among the three algorithms in a nonlinear scenario.

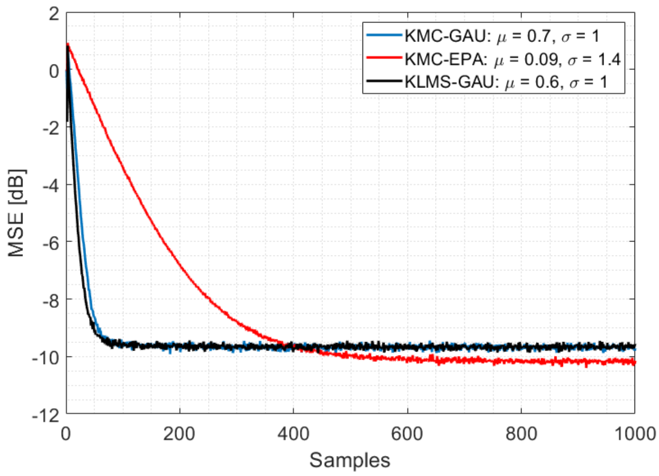

Fig. 3. Convergence curve in linear channel with impulsive noise using correlated signal.

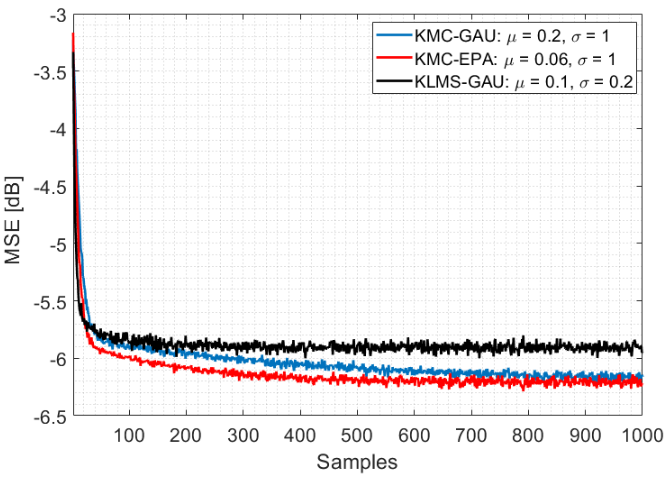

Fig. 4. Convergence curve in nonlinear channel with AWGN.

VI. Conclusion

Kernel Adaptive Filtering is an efficient approach for nonlinear channel equalization, due to its universal approximation and convexity, achieving notable results in scenarios where linear filters normally fail. Among this class of algorithms, there is the Kernel Maximum Correntropy, an efficient and robust algorithm that is based on the generalized correlation measure called correntropy. Normally in literature, it is common to estimate the correntropy using the Gaussian kernel. In this work, we proposed to estimate the correntropy using the Epanechnikov kernel and analyzed its performance. For this reason, we compare the resulting algorithm with the KMC using the Gaussian kernel and the Kernel-Least-Mean Square algorithm in different scenarios. Using an uncorrelated signal, with linear channel and impulsive noise, the three algorithms converge to the same MSE level. When using a correlated signal, even if the algorithm using the Epanechnikov kernel achieves the lowest MSE level, it needs more samples to converge. However, this might indicate that the Epanechnikov kernel is able to preserve the temporal structure of the data in comparison with the Gaussian kernel. Finally, in a nonlinear scenario with additive white Gaussian noise, the algorithm using the Epanechnikov kernel presented the best performance.

Acknowledgement

The authors would like to thank the São Paulo Research Foundation (FAPESP - Process 2018/17678-3) and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES - Finance Code 001).

References

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

[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] 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.

[5] 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.

[6] E. Parzen, “On estimation of a probability density function and mode,” *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[7] 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.

[8] R. He, W.-S. Zheng, and B.-G. Hu, “Maximum correntropy criterion for robust face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1561–1576, 2010.

[9] J. Cao, H. Dai, B. Lei, C. Yin, H. Zeng, and A. Kummert, “Maximum correntropy criterion-based hierarchical one-class classification,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3748–3754, 2020.

[10] R. He, B.-G. Hu, W.-S. Zheng, and X.-W. Kong, “Robust principal component analysis based on maximum correntropy criterion,” *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1485–1494, 2011.

[11] B. Chen, X. Wang, Y. Li, and J. C. Principe, “Maximum correntropy criterion with variable center,” *IEEE Signal Processing Letters*, vol. 26, no. 8, pp. 1212–1216, 2019.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] J. C. Principe, D. Xu, J. Fisher, and S. Haykin, “Information theoretic learning,” *Unsupervised Adaptive Filtering*, vol. 1, pp. 265–319, 2000.

[17] Z. Yang, A. T. Walden, and E. J. McCoy, “Correntropy: Implications of nongaussianity for the moment expansion and deconvolution,” *Signal Processing*, vol. 91, no. 4, pp. 864–876, 2011.

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