Sparse Dictionary Construction for Kernel Adaptive Filtering with non-Gaussian functions

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

***Abstract*— Kernel Adaptive Filtering (KAF) has gained attention mainly due to its ability to deal with nonlinear channel equalization and impulsive noise. In this context, we focus on the Kernel Maximum Correntropy (KMC), in which the Gaussian and Epanechnikov kernels were used. However, the latter presents a numerical instability in which the algorithm diverges after a training period. In this paper, we address this problem using data dictionaries, which can be constructed through online sparsification methods such as the Novelty Criterion (NC), the Coherence Criterion (CC), and the Surprise Criterion (SC). We simulated the KMC using both kernels and sparse dictionaries in linear and nonlinear scenarios, comparing their performance with the Kernel Least-Mean-Square (KLMS). The algorithms presented desirable behavior and did not present divergence.**

***Keywords*— Kernel Adaptive Filtering, Epanechnikov Kernel, Correntropy, Information Theoretic Learning, Sparsification**

I. Introduction

Kernel Adaptive Filtering (KAF) algorithms have become popular recently due to their ability to deal with nonlinear problems by mapping the input data to a high-dimensional space, known as Reproducing kernel Hilbert Space (RKHS), thereby allocating a kernel function for each data. The objective of this approach is to express the system output in terms of the transformed input data, which is possible by using the "kernel trick". Therefore the method presents linearity, convexity and universal approximation capabilities [1].

The topology of KAF algorithms is similar to a network of Radial Basis Functions (RBF), where the parameter of each node is adaptively computed during the training period using a criterion. In this context, the Kernel Least-Mean-Square (KLMS) is the result of the combination of KAF algorithms with the traditional Least-Mean-Square (LMS), therefore inheriting its simplicity and robustness [2]. However, this criterion is based on the minimization of the Mean Squared Error (MSE), which is not optimal when dealing with non-Gaussian distributed errors. Alternatively, Information Theoretic Learning (ITL) criteria may lead to better solutions, such as the Maximum Correntropy Criterion (MCC). This criterion inspired the creation of the Kernel Maximum Correntropy (KMC) [3], which has proven to be able to deal with impulsive noise scenarios.

It is also necessary to be careful when choosing a kernel function. Historically in the literature, the Gaussian kernel is the default choice for the KAF filters in general, due to its numerical stability [1]. However, the Epanechnikov kernel is optimal for pdf estimation [4] and shown to be superior to the Gaussian kernel in channel equalization problems in certain scenarios [5]–[9].

In [9], the KMC algorithm was developed using both Gaussian and Epanechnikov kernel [4]. However, the KMC using the Epanechnikov kernel diverges after a certain number of training samples, a problem that may happen due to numerical instability, since the computational complexity increases proportionally to the input dimensionality. A possible solution to this problem is the use of sliding windows to reduce the complexity of the algorithm and avoid its divergence [10].

Even though divergence was avoided, another interesting approach is to limit the size of the KMC through the use of a dictionary, constructed online using sparsification methods [1]. These methods usually initialize an empty dictionary and add new income units to it according to a chosen criterion. Among the criteria known in the literature, we have the Novelty Criterion (NC) [11], the Coherence Criterion (CC) [12], and the Surprise Criterion (SC) [13]. In this work, we propose using these three online sparsification criteria and comparing their performance when implemented with the KLMS and the KMC algorithms using Gaussian and Epanechnikov kernels.

This work was structured as follows: in Section II there is an overview of the KMC using the Gaussian and Epanechnikov kernels. Section III discusses online sparsification techniques. In Section IV we have the algorithms performance in the channel equalization problem. Finally, the conclusions of this work are presented in Section V.

II. Foundations

In this section, we present Kernel Adaptive Filtering and its contextualization in the channel equalization problem. Furthermore, we discuss the Kernel Maximum Correntropy and its two versions using Gaussian and Epanechnikov kernels.

A. Kernel Adaptive Filtering

The objective of Kernel Adaptive Filtering (KAF) is to learn a continuous input-output mapping $f : \mathbb{U} \rightarrow \mathbb{R}$ based on a sequence of paired samples $\{s_1, x_1\}, \{s_2, x_2\}, \dots, \{s_i, x_i\}$, where $\mathbb{U} \subseteq \mathbb{R}$ is the input domain, s_i is the initially transmitted signal, and x_i is the received signal distorted by

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, Universidade Estadual de Campinas, Campinas - SP, e-mail: denisf@unicamp.br.

a transmission channel [10], [13]. It uses kernel functions to map x_i to $\varphi(x_i)$ in a high-dimensional space $\mathbb{F}$, known as the Reproducing kernel Hilbert space (RKHS), where linear operations are applied [1]. The KAF filters are usually formulated so that their output is a function of the inner product of two transformed data, getting the advantage of the "kernel trick". The output can then be expressed through the "Representer Theorem" as [14]:

$$f = \sum_{i=1}^{N} a_i \kappa(\mathbf{x}_i, \cdot), \tag{1}$$

where a_i are the weight coefficients and $\kappa(\cdot)$ is a Mercer kernel, i.e. symmetric, continuous, and positive semi-definite [1]. It is possible to notice that the topology of the KAF algorithm (1) is similar to a growing Radial Basis Function (RBF) network that expands linearly with the size of the training sequence. The coefficients connecting each node to the output are adaptively adjusted within the RKHS during the training period through the gradient descent method. Thus, we can find the equalizer weights, denoted by $\mathbf{\Omega}$, using the following equation:

$$\mathbf{\Omega}_n = \mathbf{\Omega}_{n-1} + \mu \nabla J_n, \tag{2}$$

where μ is the learning rate and J_n is an objective function.

B. Kernel Maximum Correntropy (KMC)

Correntropy is a measure of ITL which can be defined as a "generalized" correlation for pairs of random variables [15]. This measure contains second and higher-order pdf moments, which are expressed implicitly by the kernel used for its estimation [16].

Considering two arbitrary random variables X and Y, the cross-correntropy between them can be defined by the following equation:

$$V_{XY}(m) = E_{XY}\left[\kappa(X,Y)\right], \tag{3}$$

where $E[\cdot]$ denotes the expected value.

Since it measures the generalized similarity between these variables, the cross-correntropy reaches its maximum when $X = Y$, which inspired the Maximum Correntropy Criterion (MCC). In the context of adaptive signal equalization, we can replace the random variables in (3) by the transmitted signal s_i and the equalized signal $y_i = \mathbf{\Omega}_n \boldsymbol{\varphi}_n(x)$, so the MCC objective function can be defined by the equation:

$$J_{MCC} = \frac{1}{N} \sum_{i=1}^{n} \kappa_\sigma(s_i, y_i). \tag{4}$$

The MCC can be used by a KAF filter (2), to adjust the coefficients weights. This algorithm is called KMC and was first proposed by [3].

1) KMC with the Gaussian kernel: The Gaussian kernel is the primary choice among most of the works found in the literature, due to its stability and properties, which facilitate some operations, such as convolutions [8]. The Gaussian kernel function corresponds to a normal distribution and can be defined by the following equation [4]:

$$\kappa_G(s_i, y_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(s_i - y_i)^2}{2\sigma^2}}, \tag{5}$$

where σ is the kernel size or bandwidth. The KMC was originally proposed using the Gaussian kernel [3], thus following (2) and (4), the equalizer weights can be determined using the stochastic gradient approximation of (4):

$$\mathbf{\Omega}_{n+1} = \mathbf{\Omega}_n + \mu \frac{\partial \kappa_G(s_n, \mathbf{\Omega}_n^T \boldsymbol{\varphi}_n)}{\partial \mathbf{\Omega}_n} = \mu \sum_{i=1}^{n} \exp\left(\frac{-e_i^2}{2\sigma^2}\right) e_i \boldsymbol{\varphi}_i, \tag{6}$$

where $\boldsymbol{\varphi}_n = \boldsymbol{\varphi}(x_n)$, $e_n = s_n - y_n = s_n - \mathbf{\Omega}_n^T \boldsymbol{\varphi}_n$ and κ_G is the Gaussian kernel. The output of the equalizer can be obtained using the "kernel trick" [1]–[3]:

$$y_{n+1} = \mu \sum_{i=1}^{n} \exp\left(\frac{-e_i^2}{2\sigma^2}\right) e_i \kappa_G(x_i, x_{n+1}). \tag{7}$$

In this work, the KMC algorithm using the Gaussian kernel will be referred to as KMC-GAU.

2) KMC with the Epanechnikov kernel: The Epanechnikov kernel [17] is frequently used in statistical and machine learning methods. This kernel function is considered optimal for pdf estimation and is defined as a second-order polynomial function, adjusted to correspond to a density function [4]. The Epanechnikov kernel function is described by the equation below:

$$\kappa_E(s_i, y_i) = \begin{cases} \frac{3}{4\sigma}\left(1 - \left(\frac{s_i - y_i}{\sigma}\right)^2\right), & -\sigma < s_i - y_i < \sigma \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The KMC can be developed using the Epanechnikov kernel (8) through the gradient descent approach (2):

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

where κ_E is the Epanechnikov kernel. Using the "kernel trick" again, the output of the system can be computed similarly to (7):

$$y_{n+1} = \mu \frac{3}{2\sigma^3} \sum_{i=1}^{n} e_i \kappa_E(x_i, x_{n+1}),\ -\sigma < x_i - x_{n+1} < \sigma. \tag{10}$$

This algorithm will be called KMC-EPA since it uses the Epanechnikov kernel.

III. Sparsification

KAF filters use a learning system that dynamically allocates new computational units, $D_i = \{x_j\}_{j=1}^{i}$, which are stored in memory during the training. It usually converges after a reasonable period in a stationary environment. However, in KAF filters, the computational units increase linearly with the number of training data, which becomes a problem when dealing with nonstationary environments [1], [11]. According to Occam's Razor principle, it is possible to construct a better solution with the smallest possible set of elements [1]. Therefore, a network with as few kernels as possible would be desirable since it will reduce the complexity in terms of computation and memory. Besides that, it often results in a better generalization of the model [18]. There are many methods of sparsification of kernel-based solutions, some offline and others online. Online sparsification usually starts from an empty dictionary that grows dynamically according to the chosen criterion.

A. Novelty Criterion (NC)

The Novelty Criterion (NC) was proposed in [11]. This method verifies each new input data and decides if it will become a new center in the dictionary by evaluating a two-part novelty condition. First, the NC evaluates if the minimum Euclidean distance of the new input data from other centers in the dictionary is greater than a preset threshold, δ_{NC}:

$$\min_{d_j \in D_i} (\|x_i - d_j\|) \geq \delta_{NC}. \tag{11}$$

After this, the NC checks if the prediction error, i.e. the difference between the desired signal and the equalized signal, is greater than another preset threshold, δ_{E-NC}:

$$|e_i| \geq \delta_{E-NC}. \tag{12}$$

If these two conditions are met, then the new input data will be added as a new center in the dictionary [1]. Furthermore, the δ_{E-NC} can be seen as the desired accuracy of the output and it is related to the desired steady-state error [11], [18]. After adding a few samples to the dictionary, the system may find an equalizer that leads to the desired accuracy, thereby stopping allocating new units [11].

B. Coherence Criterion (CC)

Another way to characterize a dictionary in approximation problems is using coherence as a parameter [19]. In the kernel-based context, it was first proposed in [12] as a new sparsification rule known as Coherence Criterion (CC):

$$\max_{d_i \in D} |\kappa(x_n, d_i)| \leq \delta_{CC}. \tag{13}$$

If the coherence measure is below a given threshold $\delta_{CC} \in [0,1)$, then the input unit will be added to the dictionary as a new center. This parameter determines both the level of sparsity and the coherence of the dictionary [12]. Although it is not present in the original formulation, we also considered the condition (12), since it improved the performance during simulations. This threshold will be referred to as δ_{E-CC}.

C. Surprise Criterion (SC)

Surprise is a subjective information measure that quantifies how much information a new data contains relative to the knowledge of the learning system. This measure is based on the Negative Log Likelihood (NLL) and can be calculated by the following equation [13]:

$$S_i = \frac{1}{2}\ln(r_i) + \frac{e_i^2}{2r_i} - \ln(\rho(x_i|\mathcal{T}_i)), \tag{14}$$

where $\rho(x_i|\mathcal{T}_i)$ is the input distribution hypothesized by the learning system $\mathcal{T}_i$ and r_i is defined as

$$r_i = \lambda + \kappa(x_i, x_i) - \max_{\forall d_j \in D_i} \frac{\kappa^2(x_i, d_j)}{\kappa(d_j, d_j)}, \tag{15}$$

where λ is the regularization parameter. In general, $\rho(x_i|\mathcal{T}_i)$ can be assumed as a constant if no information is available *a priori* [13].

A redundant data results in little surprise whereas an outlier data returns a high surprise value. The Surprise Criterion (SC) uses this logic to construct a dictionary based on the calculated S_i. At each iteration, the SC classifies new data into three categories: abnormal if $S_i > T_1$; learnable if $T_1 \geq S_i \geq T_2$; and redundant if $S_i < T_2$. Both T_1 and T_2 are problem-dependent parameters, but T_1 usually is a large value to disable abnormal detection [13].

IV. Results

In this section, we will analyze the performance of the KMC-GAU and KMC-EPA using online sparsification methods in linear and nonlinear scenarios, which are based on [10]. For comparison, we include the KLMS using the Gaussian kernel in the analysis, a simple and robust KAF filter presented in the literature [1]. Each input data and center of the dictionary is composed of a vector of size $M = 5$. The evaluation metric used will be the Mean Square Error (MSE), in addition, we will also analyze the final size of each dictionary obtained. The parameters were chosen in order to lead to the best MSE performance. When using NC and CC, μ varied linearly within a given interval.

A. Linear Scenario

In this simulated scenario, we transmitted a Binary Phase-Shift Keying (BPSK) signal that first passes through a pre-coder $f(z) = 1 + 0.5z^{-1}$, adding correlation to the signal. This filter output will be the transmitted signal, s_n. It then passes through a linear channel $h(z) = 0.2 + 1z^{-1} + 0.4z^{-2}$ followed by an additive impulsive noise [3], whose probability density function is described by $p_{noise} = 0.9\mathcal{N}(0,\sigma_1) + 0.1\mathcal{N}(0,\sigma_2)$, where $\mathcal{N}$ is a Normal distribution, with $\sigma_2 = 0.8$ and σ_1 adjusted to obtain a resulting SNR of 20 dB [9]. It is worth mentioning that, as

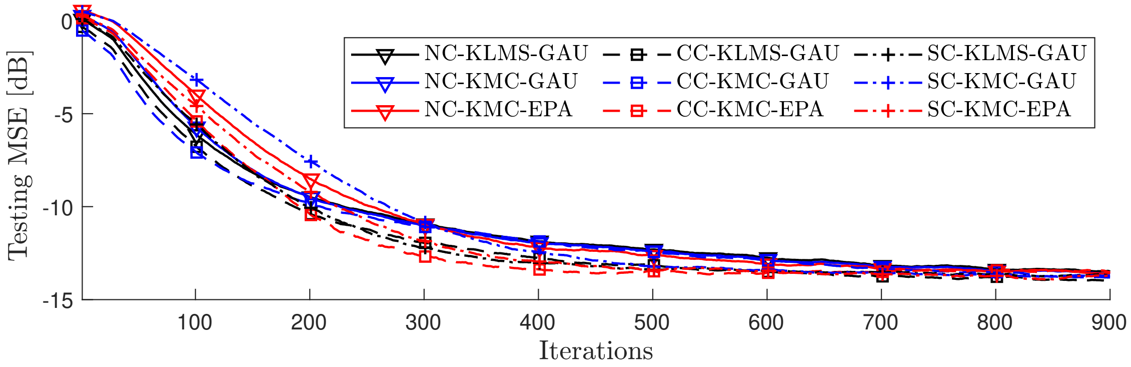

Fig. 1. Convergence curve in a linear channel with impulsive noise using a correlated signal

in [9] and [10], the error e_i was calculated with a 1-sample delay in s_n to improve the performance. The parameters used in the algorithms during this simulation can be found in Table I.

TABLE I

PARAMETERS USED IN THE LINEAR CHANNEL SCENARIO

	NC $\delta_{NC} = 1.8$ $\delta_{E-NC} = 0.35$		CC $\delta_{CC} = 0.35$ $\delta_{E-CC} = 0.2$		SC $T_1 = 50$ $T_2 = 0.5$ $\lambda = 0.001$	
	μ	σ	μ	σ	μ	σ
KLMS-GAU	2 ~ 3	1.2	1 ~ 2	1.1	2	0.8
KMC-GAU	2 ~ 3	1.2	3 ~ 4	1.2	3	0.8
KMC-EPA	0.5 ~ 1	1.7	1 ~ 5	1.5	2	1.3

Figure 1 results from an average of 1000 simulations. We can notice that all the algorithms converged to similar MSE values and that the dictionary criteria have the main influence on the performance. The algorithms using CC and SC achieved the best results regarding the MSE level. There is approximately no performance difference between the use of the two kernels.

In Figure 2 we can see box plots describing the distribution of the dictionary size for the same 1000 simulations of Figure 1. The algorithms using NC and SC resulted in dictionaries with sizes between 100 and 200 centers, whereas algorithms using CC resulted in varied sizes of dictionaries. The CC-KMC-GAU, for example, generated the largest dictionaries with sizes around 400 centers. On the other hand, the shortest dictionary was obtained by the CC-KMC-EPA.

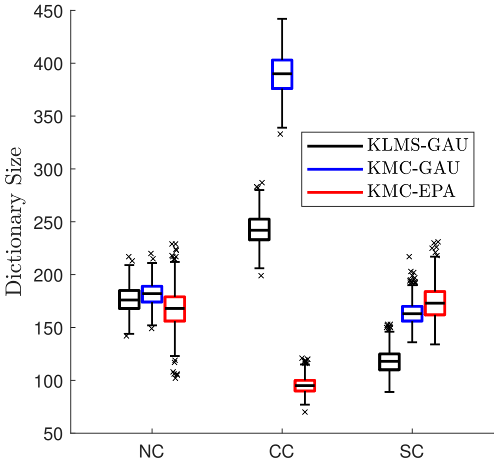

Fig. 2. Dictionary size box plots in a linear channel with impulsive noise using a correlated signal

B. Nonlinear Scenario

In this scenario, we transmitted a BPSK signal through the nonlinear channel defined by $z_n = s_n + 0.2s_{n-1}$ and $x_n = z_n - 0.9z_n^2 + \nu_\sigma$, where ν_σ is an AWGN noise [2]. We considered an SNR of 20 dB and there is no delay of s_n on the obtention of the error. In Table II, we can find the parameters set in this simulation, which were chosen in order to achieve the best MSE performance.

TABLE II

PARAMETERS USED IN THE NONLINEAR CHANNEL SCENARIO

	NC $\delta_{NC} = 1.8$ $\delta_{E-NC} = 0.2$		CC $\delta_{CC} = 0.5$ $\delta_{E-CC} = 0.4$		SC $T_1 = 50$ $T_2 = 0.38$ $\lambda = 0.001$	
	μ	σ	μ	σ	μ	σ
KLMS-GAU	1 ~ 2	1	3	1	2	0.8
KMC-GAU	1 ~ 3	1	3 ~ 4	1	2	0.8
KMC-EPA	0.01 ~ 0.1	1.5	2 ~ 3	1.4	2	1.5

In Figure 3 we have the results of an average of 1000 simulations. In this scenario, we can notice that the algorithms using the Gaussian kernel achieved a better performance in terms of MSE level. In terms of the dictionary criteria, SC had the best performance. NC converged to the same MSE but had a slower convergence.

Figure 4 displays the box plots of the dictionary size distribution of the same 1000 simulations of Figure 4. We can see that the algorithms using CC resulted in the smallest dictionaries, between 50 and 100 centers. In contrast, the dictionaries generated by the algorithms using NC and SC are larger, with around 150 centers stored. In this case, the NC-KMC-EPA generated the largest dictionaries, around 300 centers. It is interesting to note that neither the method with the largest dictionaries nor the method with the smallest ones led to the best MSE performance, which was achieved by the SC criterion.

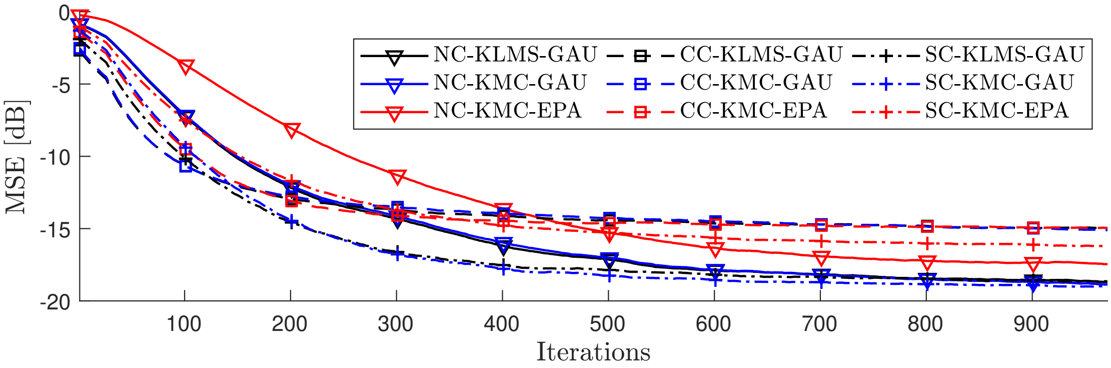

Fig. 3. Convergence curve in a nonlinear channel with additive noise using a BPSK signal

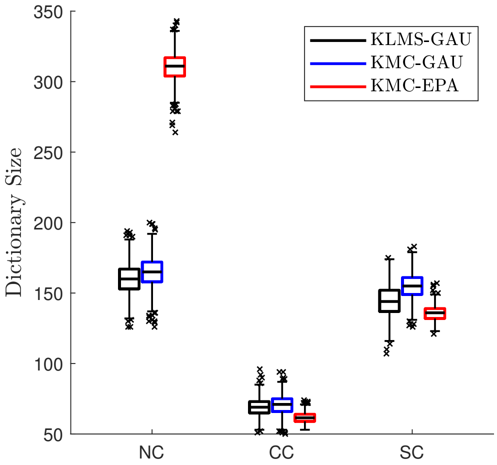

Fig. 4. Dictionary size box plots in a nonlinear channel with additive noise using a BPSK signal

V. Conclusion

The Kernel Maximum Correntropy has shown to be an efficient tool for dealing with nonlinear channel equalization and impulsive noise scenarios. However, it presented a numerical instability when implemented with the Epanechnikov kernel in previous work [9]. In this work, we propose the use of data dictionaries to address this issue and reduce the computational complexity of the algorithm. In this sense, we implemented the KMC-EPA alongside online sparsification methods and simulated it in linear and nonlinear scenarios, comparing it with the KMC-GAU and the KLMS. The algorithms presented stability and a desirable performance in terms of MSE level. Furthermore, simulation results showed that the sparsification methods can reduce the complexity of the algorithms without affecting performance. In both linear and nonlinear simulation scenarios, the SC criterion is among the best in terms of MSE performance, with a reasonable dictionary size.

Acknowledgments

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

[2] 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.

[3] 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.

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

[5] E. Martins, *Análise do Desempenho de Algoritmos de Equalização Cega Baseados em ITL com Kernel Epanechnikov*. Mestrado, Universidade Federal do ABC, 2018.

[6] 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.

[7] 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.

[8] L. Gois and A. Neves, “Equalização adaptativa cega de sinais com kernel não-gaussiano,” *XXXVII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 2019.

[9] 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.

[10] L. H. Gois, D. G. Fantinato, and A. Neves, “New approaches for the kernel-based adaptive filter with epanechnikov kernel,” *XLI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 2023.

[11] J. Platt, “A resource-allocating network for function interpolation,” *Neural computation*, vol. 3, no. 2, pp. 213–225, 1991.

[12] C. Richard, J. C. M. Bermudez, and P. Honeine, “Online prediction of time series data with kernels,” *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2009.

[13] W. Liu, I. Park, and J. C. Principe, “An information theoretic approach of designing sparse kernel adaptive filters,” *IEEE transactions on neural networks*, vol. 20, no. 12, pp. 1950–1961, 2009.

[14] B. Schölkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem,” in *International conference on computational learning theory*, pp. 416–426, Springer, 2001.

[15] I. Santamaria, 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, *Information theoretic learning: Renyi’s entropy and kernel perspectives*. Springer Science & Business Media, 2010.

[17] V. A. Epanechnikov, “Non-parametric estimation of a multivariate probability density,” *Theory of Probability & Its Applications*, vol. 14, no. 1, pp. 153–158, 1969.

[18] F. Tobar, *Kernel-based adaptive estimation: Multidimensional and state-space approaches*. PhD thesis, Citeseer, 2014.

[19] J. Tropp, “Greed is good: algorithmic results for sparse approximation,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.