


RESEARCH

Open Access



Performance evaluation of the maximum complex correntropy criterion with adaptive kernel width update

Manoel B. L. Aquino^{1,2*}, João P. F. Guimarães^{1,3}, Leandro L. S. Linhares¹, Aluísio I. R. Fontes¹  and Allan M. Martins²

Abstract

The complex correntropy is a recently defined similarity measure that extends the advantages of conventional correntropy to complex-valued data. As in the real-valued case, the maximum complex correntropy criterion (MCCC) employs a free parameter called kernel width, which affects the convergence rate, robustness, and steady-state performance of the method. However, determining the optimal value for such parameter is not always a trivial task. Within this context, several works have introduced adaptive kernel width algorithms to deal with this free parameter, but such solutions must be updated to manipulate complex-valued data. This work reviews and updates the most recent adaptive kernel width algorithms so that they become capable of dealing with complex-valued data using the complex correntropy. Besides that, a novel gradient-based solution is introduced to the Gaussian kernel and its respective convergence analysis. Simulations compare the performance of adaptive kernel width algorithms with different fixed kernel sizes in an impulsive noise environment. The results show that the iterative kernel adjustment improves the performance of the gradient solution for complex-valued data.

Keywords: Adaptive filter, Adaptive kernel width, Maximum complex correntropy criterion (MCCC), Complex-valued data

1 Introduction

The correntropy consists in a similarity measure based on Rényi entropy capable of extracting high-order statistical information from real-valued data [1]. This is why it has been widely used as a cost function in optimization problems such as adaptive filtering in an approach called maximum correntropy criterion (MCC), thus providing better performance than second-order methods in non-Gaussian noise environments [2–6]. Recently, the correntropy concept has been extended to complex-valued random variables using the maximum complex correntropy criterion (MCCC) [7, 8].

Both MCC and MCCC employ a free parameter called kernel width or kernel size. It essentially controls the nature of the performance surface over which the system parameters are adapted, as it has important effects on distinct aspects, e.g., convergence speed, presence of local optima, and stability of weight tracks [9]. Since the task of obtaining an optimum value for this parameter is time consuming and not trivial, a series of adaptive kernel width algorithms has been proposed in order to choose a proper value for this parameter at each iteration in optimization problems.

An algorithm called adaptive kernel width MCC (AMCC) was proposed in [10] aiming to improve the learning speed, especially when the initial weight vector is far from being optimal. Another method called switch kernel width method of correntropy (SMCC) in [11] updates the kernel width based on the instantaneous error between the estimate and the desired signal in order to adjust such parameter for each iteration. Recently, the

*Correspondence: bonfim.aquino@ifrn.edu.br

¹Federal University of Rio Grande do Norte (UFRN), Campus Universitário Lagoa Nova, 59078-970 Natal, RN, Brazil

²Federal Institute of Education, Science and Technology of Rio Grande do Norte (IFRN), BR 405, KM 154, S/N, Chico Cajá, CEP 59900-000 Pau dos Ferros, RN, Brazil

Full list of author information is available at the end of the article

technique developed in [12] called variable kernel width-maximum correntropy criterion (VKW-MCC) has been suggested as a solution capable of searching for the best kernel width at each iteration, thus implying reduced error. This strategy is able to provide fast convergence rate and stable steady-state performance. All the aforementioned algorithms were proposed for real-valued data. However, literature apparently does not present any work that evaluates the application of adaptive kernel width algorithms involving complex-valued data.

This paper updates the most recently adaptive kernel width algorithms in order to deal with complex-valued data. Besides that, Wirtinger calculus is applied to propose a novel gradient-based solution to Gaussian kernels using the complex correntropy as a cost function. A convergence analysis of the gradient-based algorithm is presented, as well as simulations comprising a comparative analysis regarding the performance of all adaptive kernel width algorithms in an impulsive noise environment. The results show that the novel adaptive kernel methods improve the performance of the gradient solution for complex-valued data considering the channel identification scenario of a 16-QAM (quadrature amplitude modulation) modulation signal.

The remainder of the paper is organized as follows. Section 2 reviews the complex correntropy function concepts and its use as a cost function to define a new ascendant gradient solution. Section 3 provides the analysis of each adaptive kernel width algorithm evaluated in this study updating its strategy to deal with complex-valued data. Section 4 presents simulations to analyze the performance of the proposed methods and compared it with classical solution from the literature. Finally, Section 5 discusses the main contributions and results regarding the study developed in this work.

2 Methods

2.1 Complex correntropy

Recently, the correntropy function was extended to the case of complex-valued data. This approach is called complex correntropy and is defined as [7]:

$$V_{\sigma}^c(Q, B) = E[\kappa_{\sigma}(Q, B)], \quad (1)$$

where $\kappa_{\sigma}(\cdot)$ is any positive-definite kernel with kernel width σ , and Q, B are complex random variables. $E[\cdot]$ is the expected value operator.

The work presented in [8] demonstrated that the complex correntropy generalizes the regular correntropy concept to complex-valued data while keeping important properties such as symmetry, bounded, high-order statistical measure, and a probabilistic meaning, specially when the complex Gaussian kernel, defined in (2) is used.

$$G_{\sigma}^c(q - b) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(q - b)(q - b)^*}{2\sigma^2}\right), \quad (2)$$

where $(\cdot)^*$ is the complex conjugate operator.

Let q (negrito) and b (negrito) columns vectors with N complex-valued samples of the random variables Q and B . Then, using the complex Gaussian Kernel, one can estimates the complex correntropy between Q and B as

$$\hat{V}_{\sigma}^c(Q, B) = \frac{1}{2\pi\sigma^2} \frac{1}{N} \sum_{i=1}^N \exp\left(-\frac{(q_i - b_i)(q_i - b_i)^*}{2\sigma^2}\right). \quad (3)$$

2.2 Maximum complex correntropy criterion (MCCC)

The use of the complex correntropy as a cost function was first proposed in [7] to solve a linear system identification problem. The goal is to maximize the complex correntropy between a desired complex signal $\mathbf{d} \in \mathbb{C}^N$ and the estimated system output $\mathbf{y} = \mathbf{w}^H \mathbf{x}$, where $\mathbf{w} \in \mathbb{C}^N$ is a complex column vector representing the system weights and $\mathbf{X} \in \mathbb{C}^{N \times M}$ is the system input. $[\cdot]^H = ([\cdot]^T)^*$ is the Hermitian operator. Summarizing, let J_{MCCC} be the cost function

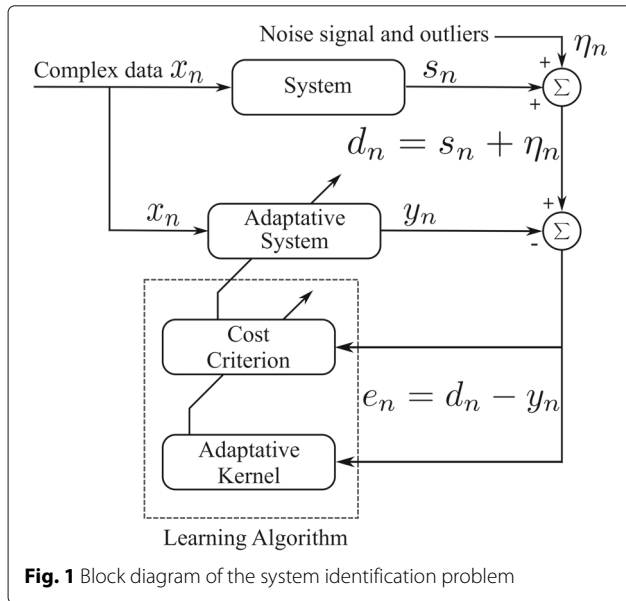
$$\begin{aligned} J_{MCCC} &= V_{\sigma}^c(d, y) = E[G_{\sigma}^c(d, y)] \\ &= \frac{1}{2\pi\sigma^2} \frac{1}{L} \sum_{i=1}^L \exp\left(-\frac{(d_i - y_i)(d_i - y_i)^*}{2\sigma^2}\right) \\ &= \frac{1}{2\pi\sigma^2} \frac{1}{L} \sum_{i=1}^L \exp\left(-\frac{(d_i - \mathbf{w}^H \mathbf{x}_i)(d_i - \mathbf{w}^H \mathbf{x}_i)^*}{2\sigma^2}\right), \end{aligned} \quad (4)$$

which needs to be maximized. \mathbf{x}_i is the i th column of the input matrix \mathbf{X} . This would lead to maximization of the similarity between \mathbf{y} and \mathbf{d} , causing the error $\mathbf{e} = \mathbf{d} - \mathbf{y}$ to zero. This approach is called maximum complex correntropy criterion (MCCC). Figure 1 summarizes a system identification problem in which the MCCC was successful applied in [7]. The MCCC has been also applied to a channel equalization problem [8], but always employing a fixed-point solution algorithm. This approach depends on a matrix inversion, and this operation is sometimes unavailable or the increase in the computational cost is not ideal [13]. Furthermore, the gradient solution will always provide the solution with the least norm, which may be useful in some scenarios [14]. Besides that, all the adaptive kernel size algorithms mentioned in this work employ the gradient solution, which still needs to be introduced in the MCCC (Additional file 1).

So, in order to obtain the update rule, it is possible to write:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu \nabla J_n, \quad (5)$$

where μ is the step size.



To obtain ∇J_n , the most obvious choice would be differentiating Eq. 4 with respect to \mathbf{w}^* . However, Eq. 4 depends on complex-valued parameter (\mathbf{d}, \mathbf{y}) , although it is always a real-valued function when the complex Gaussian kernel from Eq. (2) is applied [15]. This violates the Cauchy-Riemann conditions, thus making the gradient function not analytical in the complex domain [16]. Hence, standard differentiation cannot be applied. One possible alternative to overcome this problem is to consider the cost function defined in the Euclidean domain with double dimensionality (\mathbb{R}^2), although this approach leads to onerous computations [17]. The Wirtinger calculus, which will be briefly presented in this section later on, provides an elegant way to obtain a gradient of real-valued cost function that is defined in complex domains.

2.3 Wirtinger calculus

Based on the duality between spaces \mathbb{C} and \mathbb{R}^2 , the Wirtinger calculus was firstly introduced in [18]. Let $f : \mathbb{C} \rightarrow \mathbb{C}$ be a complex function defined in \mathbb{C} . Such function can also be defined in \mathbb{R}^2 (i.e., $f(x + jy) = f(x, y)$).

The Wirtinger's derivative of f at a point c is defined as follows [17]

$$\frac{\partial f}{\partial z}(c) = \frac{1}{2} \left(\frac{\partial f}{\partial x}(c) - j \frac{\partial f}{\partial y}(c) \right) \quad (6)$$

On the other hand, the conjugate Wirtinger's derivative of f at c is given by:

$$\frac{\partial f}{\partial z^*}(c) = \frac{1}{2} \left(\frac{\partial f}{\partial x}(c) + j \frac{\partial f}{\partial y}(c) \right) \quad (7)$$

In other words, in order to compute the Wirtinger derivative of a given function f , it can be expressed in terms of z and z^* . Then, the usual differentiation rules can

be applied after considering z^* as a constant. The same concept can be used to compute the conjugate Wirtinger derivative of a function f , also expressed in terms of z and z^* . In this case, usual differentiation rules must be employed considering z as a constant [17], i.e., considering f as $f(z) = zz^*$, which leads to:

$$\frac{\partial f}{\partial z} = z^* \quad \text{and} \quad \frac{\partial f}{\partial z^*} = z \quad (8)$$

2.4 Gradient ascent solution

Using the Wirtinger calculus to obtain the gradient ∇J_n :

$$\nabla J_n = \frac{\partial J_n}{\partial \mathbf{w}^*} = \frac{1}{2\pi\sigma^2} \frac{1}{N} \sum_{i=1}^N \exp \left(-\frac{(d_i - y_i)(d_i - y_i)^*}{2\sigma^2} \right) \frac{(-1)}{2\sigma^2} \frac{\partial (e_i e_i^*)}{\partial \mathbf{w}^*}, \quad (9)$$

where $e_i = d_i - \mathbf{w}_i^H \mathbf{x}_i$.

Thus, it leads to the follow update rule:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \frac{\mu}{N4\pi\sigma^3} \sum_{i=1}^N \exp \left(-\frac{e_i e_i^*}{2\sigma^2} \right) e_i^* \mathbf{x}_i. \quad (10)$$

Finally, applying the stochastic gradient gives:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \frac{\mu}{N4\pi\sigma^3} \exp \left(-\frac{e_n e_n^*}{2\sigma^2} \right) e_n^* \mathbf{x}_n \quad (11)$$

A complete step-by-step derivation can be seen in (5).

2.5 Convergence analysis

In this section, the convergence of the proposed weight update method is investigated based on stochastic gradient for complex valued-data. It can be considered as an extension of the convergence analysis realized in [19]. Initially, the algorithm described by Eq. 11 can be written in a simplified form:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \eta f[e_n] \mathbf{x}_n, \quad n \geq 0, \quad (12)$$

where η is the step size, and $f[e_n]$ is a nonlinear function of the estimation error $e(i)$, being expressed as:

$$f[e(i)] = \exp \left(-\frac{e_n e_n^*}{2\sigma^2} \right) e_n^*. \quad (13)$$

Let us assume that the desired system output signal d_n can be expressed as:

$$d_n = \mathbf{w}_o^H \mathbf{x}_n + v_n, \quad (14)$$

where \mathbf{w}_o is the optimum weight vector that must be estimated, and v_n represents the disturbance noise. Then, the estimation error at instant time n is given by:

$$e_n = d_n - \mathbf{w}_n^H \mathbf{x}_n = \mathbf{w}_o^H \mathbf{x}_n - \mathbf{w}_n^H \mathbf{x}_n + v_n. \quad (15)$$

Considering that the weight-error vector is defined as $\tilde{\mathbf{w}}_n = \mathbf{w}_o - \mathbf{w}_n$, the a priori and a posteriori errors are denoted by:

$$e_a(n) = \tilde{\mathbf{w}}_n^H \mathbf{x}_n, \quad e_p(n) = \tilde{\mathbf{w}}_{n+1}^H \mathbf{x}_n. \quad (16)$$

The update rule in Eq. 12 can be rewritten in terms of the weight-error vector as:

$$\tilde{\mathbf{w}}_{n+1} = \tilde{\mathbf{w}}_n + \eta f[e_n] \mathbf{x}_n. \quad (17)$$

Post-multiplying both sides of the conjugate transpose version of Eq. 17 by \mathbf{x}_n , as well as replacing some terms in Eq. 16 for expressions, it is possible to determine a relationship between estimation errors $e_a(n)$, $e_p(n)$, and e_n in the form:

$$e_p(n) = e_a(n) - \eta f^*[e_n] \|\mathbf{x}_n\|^2. \quad (18)$$

In order to eliminate the non linearity $f[e_n]$ in Eq. 17, it is possible to combine such expression with Eq. 18 to obtain the following representation:

$$\tilde{\mathbf{w}}_{n+1} = \tilde{\mathbf{w}}_n - (e_a(n) - e_p(n))^* \frac{\mathbf{x}_n}{\|\mathbf{x}_n\|^2}. \quad (19)$$

With the objective of following an energy-based approach, both sides of Eq. 19 are squared,

$$\begin{aligned} \|\tilde{\mathbf{w}}_{n+1}\|^2 &= \left(\tilde{\mathbf{w}}_n - (e_a(n) - e_p(n))^* \frac{\mathbf{x}_n}{\|\mathbf{x}_n\|^2} \right)^H \\ &\quad \times \left(\tilde{\mathbf{w}}_n - (e_a(n) - e_p(n))^* \frac{\mathbf{x}_n}{\|\mathbf{x}_n\|^2} \right). \end{aligned} \quad (20)$$

After some algebraic manipulation of Eq. 20, it is possible to obtain an energy relation as:

$$\|\tilde{\mathbf{w}}_{n+1}\|^2 + \frac{\|e_a(n)\|^2}{\|\mathbf{x}_n\|^2} = \|\tilde{\mathbf{w}}_n\|^2 + \frac{\|e_p(n)\|^2}{\|\mathbf{x}_n\|^2}. \quad (21)$$

Since the mean-square behavior of the algorithm is of interest for the proposed study, expectations of both sides of Eq. 21 are obtained, which are then substituted in Eq. 18, representing the *a posteriori* error $e_p(n)$.

$$\begin{aligned} E[\|\tilde{\mathbf{w}}_{n+1}\|^2] &= E[\|\tilde{\mathbf{w}}_n\|^2] - 2\eta E[Re\{e_a(n)f[e_n]\}] \\ &\quad + \eta^2 E[f^2[e_n] \|\mathbf{x}_n\|^2] \end{aligned} \quad (22)$$

The convergence of the proposed algorithm depends on the choice of the learning rate. Therefore, a Lyapunov approach is adopted to obtain convergence in an upper bound for which $E[\|\tilde{\mathbf{w}}_n\|^2]$ remains uniformly bounded. Analyzing Eq. 22, it is possible to write:

$$\begin{aligned} E[\|\tilde{\mathbf{w}}_{n+1}\|^2] &\leq E[\|\tilde{\mathbf{w}}_n\|^2] \\ &\iff -2\eta E[Re\{e_a(n)f[e_n]\}] \\ &\quad + \eta^2 E[f^2[e_n] \|\mathbf{x}_n\|^2] \leq 0 \end{aligned} \quad (23)$$

From Eq. 23, it can be stated that the learning rate can be chosen for all n in the form:

$$\eta \leq 2 \frac{E[Re\{e_a(n)f[e_n]\}]}{E[\|\mathbf{x}_n\|^2]}, \quad (24)$$

Then, the sequence $E[\|\tilde{\mathbf{w}}_n\|^2]$ of weight error power will be decreasing and bounded from below, which ensures the

convergence. Thus, a sufficient condition for convergence can be alternatively expressed by:

$$\eta \leq 2 \inf_{n \geq 0} \frac{E[Re\{e_a(n)f[e_n]\}]}{E[f^4[e_n]]^{1/2} E[\|\mathbf{x}_n\|^4]^{1/2}}. \quad (25)$$

Assuming that the filter is long enough so that $e_a(n)$ is a zero-mean Gaussian and the noise process v_n is i.i.d., it is possible to define the following statements [19, 20]:

$$h_G[E[e_a^2(n)]] \triangleq \frac{E[Re\{e_a(n)f[e_n]\}]}{E[e_a^2(n)]}. \quad (26)$$

$$h_C[E[e_a^2(n)]] \triangleq E[f^4[e_n]]. \quad (27)$$

Therefore, a sufficient convergence condition can be established substituting Eqs. 26 and 27 in Eq. 25, resulting in:

$$\eta \leq \frac{2}{E[\|\mathbf{x}_n^4\|]^{1/2}} \left(\inf_{n \geq 0} \frac{E[e_a^2(n)] \cdot h_G[E[e_a^2(n)]]}{\sqrt{h_C[E[e_a^2(n)]]}} \right). \quad (28)$$

Since all terms in Eq. 28 are functions of $E[e_a^2(n)]$, it is possible to emphasize this aspect in Eq. 29 in order to indicate that the minimization takes place over the values of $E[e_a^2(n)]$.

$$\eta \leq \frac{2}{E[\|\mathbf{x}_n^4\|]^{1/2}} \left(\inf_{E[e_a^2(n)]} \frac{E[e_a^2(n)] \cdot h_G[E[e_a^2(n)]]}{\sqrt{h_C[E[e_a^2(n)]]}} \right). \quad (29)$$

Then, if the step size follow the condition described in Eq. 29, one can say that the algorithm will converge.

3 Adaptive kernel size algorithms

Analogously to the real-valued case, the complex correntropy is directly related to the estimation of how similar two random variables are when the Parzen estimator is applied to the joint probability [7]. Thus, the kernel size, also called kernel width, is a free parameter that is inherent to the kernel used to estimate the complex correntropy. It works as a scale parameter that controls the steady-state performance, convergence rate, and impulsive noise rejection [15]. Since it is a free parameter, the kernel width must be chosen by the user, whose value changes according to data and application nature. Then, the definition of an optimal value for the kernel width is not a trivial task [21].

In this context, many works have been proposed in order to help determining the optimal kernel width, e.g., [11, 12, 22, 23]. However, the aforementioned studies only deal with real-valued data. In this section, the algorithms

are then updated using the complex correntropy definition and the Wirtinger calculus in order to make them applicable to complex-valued data.

3.1 Adaptive kernel width MCCC (AMCCC)

According to [23], AMCC consists in selecting the kernel as a combination of a fixed-kernel bandwidth, which could be defined using Silverman rule [24] and the squared prediction error e_n^2 . The authors also state that this approach causes the algorithm to converge faster, especially when the initial weight vector is far away from the optimal one. Besides the fast convergence rate, prominent advantages of the method lie in simplicity, as well as no extra computational burden, as no additional free parameters are required. Since the kernel size must be always a positive and real value, it is possible to define a new update rule called AMCCC, which can be expressed by:

$$\sigma_n^2 = e_n e_n^* + \sigma^2 \quad (30)$$

where σ is the predefined kernel width and e_n is the error at the iteration n .

3.2 Switch kernel width MCCC (SMCCC)

In order to improve the convergence rate of the method, the SMCC algorithm was introduced in [11]. This work defines the new kernel update rule to MCCC based on [11] and defined as:

$$\sigma_n^2 = \max\left(\frac{e_n e_n^*}{2}, \sigma^2\right) \quad (31)$$

This is another example of a simple update rule for the kernel that does not add new free parameters to the MCCC algorithm, although robustness is maintained.

3.3 Complex variable kernel width—CVKW

The VKW-MCC algorithm calculates the kernel size at each iteration by maximizing $\exp(-e^2/2\sigma^2)$ with respect to the kernel width [12]. For this purpose, the authors employ a modified cost function to reduce the interference of the kernel size. Instead of making $J_n = E[G_{\sigma(e)}]$, a new cost function is defined as $J_k = E[\sigma^2 G_{\sigma}(e)]$. Applying the same methodology to the complex-valued case gives:

$$J_n = E[\sigma^4 G_{\sigma}(e)] = E[\sigma^4 G_{\sigma}(D - Y)] \quad (32)$$

Then, the updated stochastic gradient would be:

$$W_{n+1} = W_n + \frac{\mu}{4\pi N} \exp\left(-\frac{e_n e_n^*}{2\sigma_n^2}\right) e_n^* X_n. \quad (33)$$

At each iteration, after calculating the error e_n at the n th iteration, the kernel size is updated regarding the direction to minimize the error, resulting in:

$$\max_{\sigma_n} J(e_n) = \exp\left(-\frac{e_n e_n^*}{2\sigma_n^2}\right). \quad (34)$$

Differentiating (34) using Wirtinger calculus with respect to e_n leads to:

$$\nabla J_n(e_n) = -\exp\left(-\frac{e_n e_n^*}{2\sigma_n^2}\right) \frac{e_n^* \sigma_n (\sigma_n - 2e_n \sigma_n')}{2\sigma_n^4} \quad (35)$$

Then, making $j'(e_n) = 0$ gives:

$$\sigma_n = 2e_n \sigma_n' = 2k_{\sigma} |e_n| \quad (36)$$

4 Results and discussion

In this section, the system identification problem from [7] is revisited to evaluate the performance from the proposed ascendant gradient MCCC using a fixed kernel size and compare it with the variable kernel size strategies,

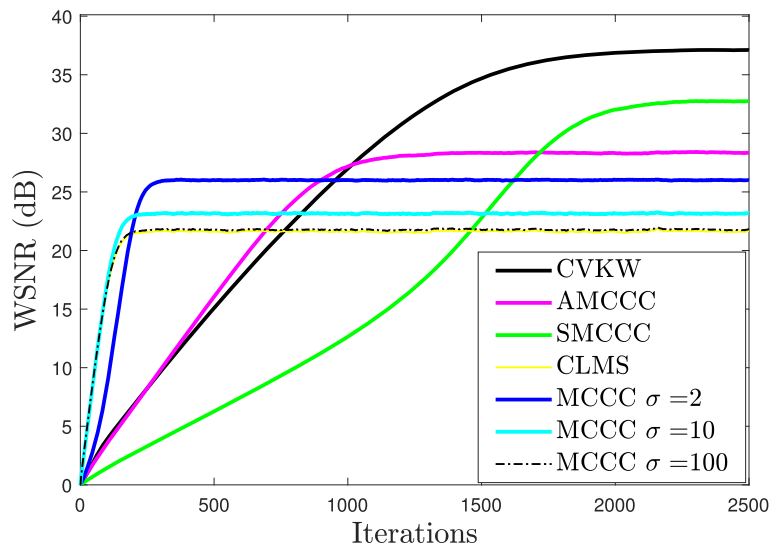


Fig. 2 Performance comparison among several methods in terms of WSNR as a function of the number of iterations for $\alpha=1.5$ and $\text{GSNR} = 20$ dB

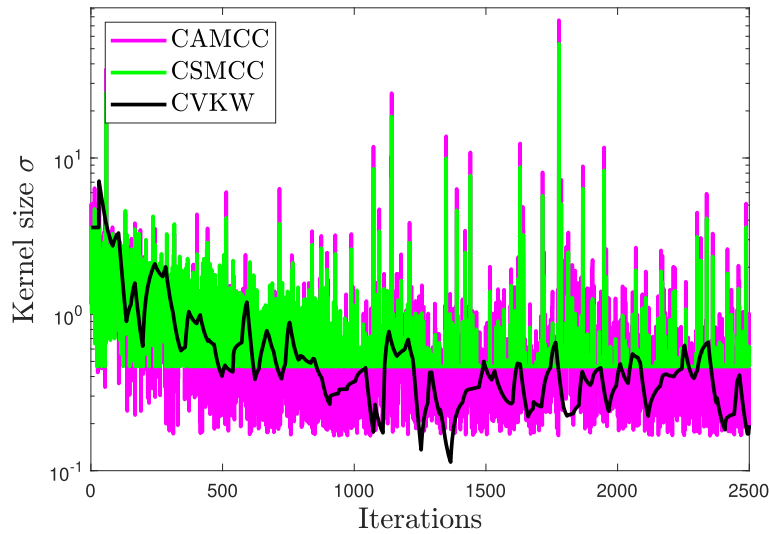


Fig. 3 Typical evolution of the Kernel size σ used by AMCCC, SMCCC, and CVKW in the scenario of Fig. 2

i.e., SMCCC, AMCCC, and CVKW. For reference, the complex least mean square (CLMS) [16], which is a classical solution from the literature, is also considered in the simulations.

The performance of the adaptive filters is evaluated by the weight signal-to-noise ratio (WSNR), which is defined as

$$\text{WSNR}_{db} = 10 \log_{10} \left(\frac{\bar{\mathbf{w}}^H \bar{\mathbf{w}}}{(\bar{\mathbf{w}} - \mathbf{w}_i)^H (\bar{\mathbf{w}} - \mathbf{w}_i)} \right), \quad (37)$$

where $\bar{\mathbf{w}}$ is the correct weights, which are randomly select at each Monte Carlo trial from a Gaussian distribution with mean 0 and variance 1. \mathbf{w}_i is the complex weights computed by the aforementioned methods in the i th iteration. The WSNR is used to quantify both convergence and misadjustment rates properly in decibels [25].

The desired signal is formed by the product of the proper weights $\bar{\mathbf{w}}$ and the input signal $\mathbf{X} \in \mathbb{C}^{2 \times 2500}$, which elements follow a Gaussian distribution with mean 0.5 and variance 1 for the real part and mean 1.5 and variance 4 for the imaginary part. Then, an additive noise signal is added.

The symmetric stable distribution [26] was used to model an impulsive noise environment to the simulations. Since its symmetric, the shift and skewness parameters are always set to 0. The index of stability $0 < \alpha \leq 2$ controls the tail of the distribution, while the scale γ parameter is obtained from a given generalized signal-to-noise ratio in dB (GSNR) [27], which is given by:

$$\text{GSNR} = 10 \log \frac{P_S}{\gamma^\alpha}, \quad (38)$$

where P_S is the power of the noiseless signal.

Figure 2 shows the performance of the proposed MCCC algorithm with three different fixed kernel sizes $\sigma = 2, 10$, and 100. Besides that, the adaptive kernel strategies and the CLMS are also included. All plots in this section are made by the average of 10^3 Monte Carlo trials, and the initial values adopted for the weights are always zero. One can notice that the best result with a fixed kernel size was $\sigma = 2$. In the simulations, a value smaller than 2 made the algorithm not converge. As in the real-valued case, the MCCC with a large value of kernel size $\sigma = 100$ made the results almost identical to the CLMS. For large kernel sizes, the complex correntropy-based algorithms tend to perform as a second order one [7, 15]. The MCCC gradient ascendant with kernel size $\sigma = 10$ had the WSNR levels between the $\sigma = 2$ and $\sigma = 100$. Also, it can be notice that the convergence speed is affected by the kernel size choice. In summary, the smaller that still makes the algorithm converge, the higher the WSNR level. In the other hand, increasing the kernel size makes the WSNR level drop and increases the convergence speed.

Table 1 Kernel size σ and step size μ used in the simulations for each tested algorithm

Algorithm	Kernel width	Step size
CVKW	6.16	0.088
SMCCC	0.21	0.088
AMCCC	0.13	0.088
MCCC	0.2	0.25

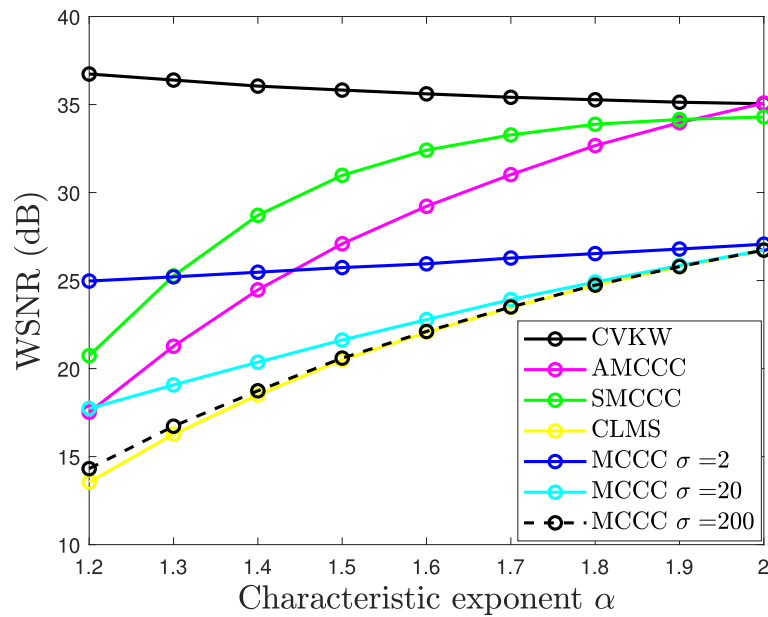


Fig. 4 Performance comparison among several methods in terms of WSNR as a function of the characteristic exponent and GSNR = 20

Analyzing Fig. 2, it is possible to see that the adaptive kernel size strategies could overcome the performance of a fixed kernel size selection after 2500 iterations. The CVKW was the algorithm that achieved the highest WSNR levels. The AMCCC had a better WSNR than the SMCCC but the SMCCC had a better convergence rate. It is important to highlight that, although the adaptive kernel size strategies have better WSNR levels, the

fixed kernel size methods, and the CLMS have a fast convergence rate.

A typical evolution of the kernel size by each adaptive algorithms compared in Fig. 2 is shown in Fig. 3. The initial values are initially based on Table 1. It is possible to see how much more aggressive the AMCCC and the SMCCC are when compared with the CVKW, due to the updates rules shown in Section 3. This is due to the

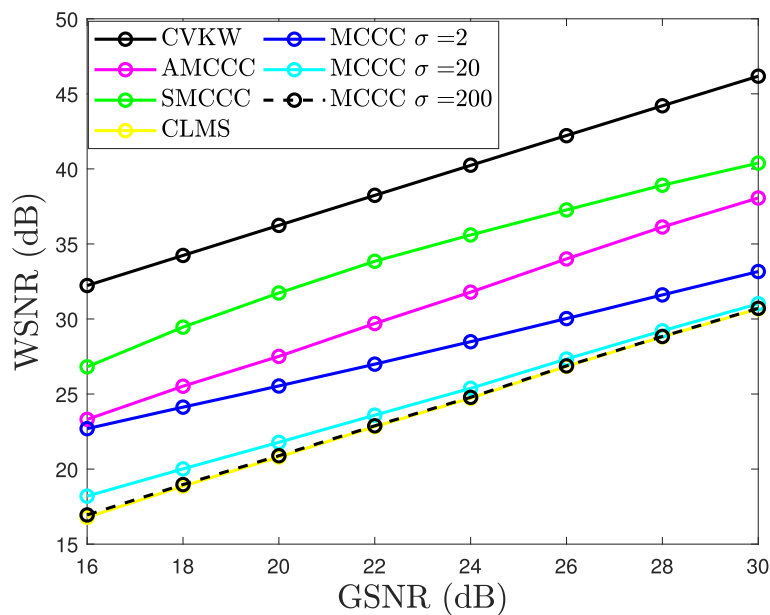


Fig. 5 Performance comparison among several methods in terms of WSNR as a function of GSNR for $\alpha = 1.5$

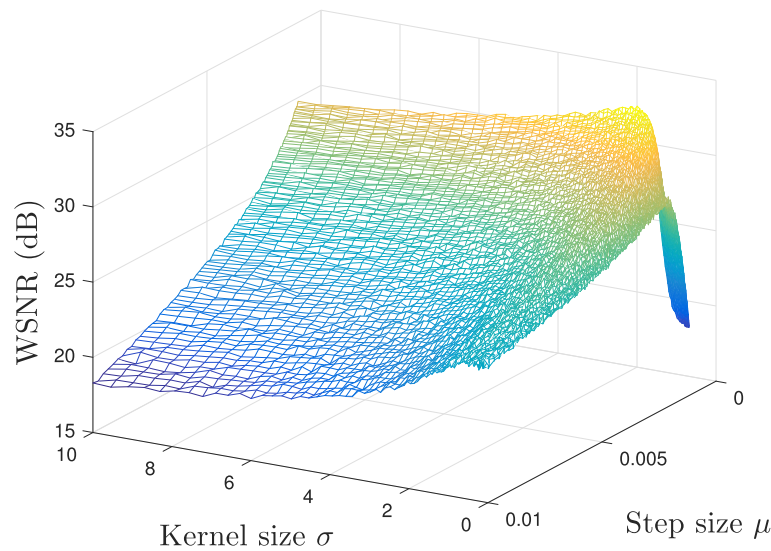


Fig. 6 WSNR performance of the AMCCC algorithm as a function of the kernel size parameter σ and the gradient step size μ . The noise environment is model by a stable distribution with $\alpha = 1.5$ and GSNR = 20 dB

smoothing factor presented in the VKW-MCC algorithm [12] which was preserved by this paper for the complex-valued case. The algorithms were tested in different noise parameters. Figure 4 compares the performance of the algorithms as a function of index of stability α with a fixed GSNR = 20dB. When $\alpha = 2$, the stable distribution behaves as a Gaussian and the smaller the value of α , the more impulsive is the noise. As expected, the CLMS performances deteriorate faster than the complex correntropy-based methods, except for the one with large kernel size, $\sigma = 100$. Also, regarding the noise

environment, Fig. 5 shows the behavior of each algorithm as a function of the GSNR and a fixed index of stability $\alpha = 1.5$. As expected, as the noise power decreases the WSNR levels increase to all algorithms.

Although the simulations showed that MCCC could deal well under impulsive noise, using the complex correntropy as a cost function includes a new free parameter that is the kernel size. This is what motivated the development of the adaptive kernel size strategies showed in this paper. However, each adaptive kernel size strategy still needs a kernel parameter as the updates Eq. (30) for the

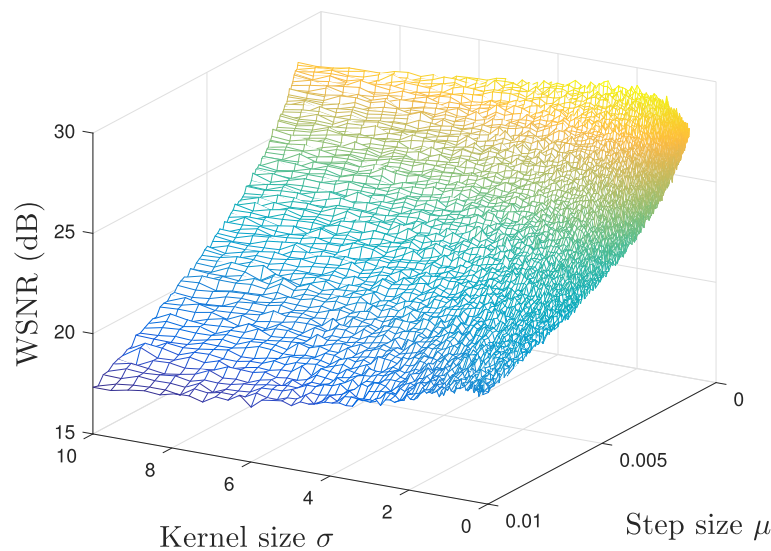


Fig. 7 WSNR performance of the SMCCC algorithm as a function of the kernel size parameter σ and the gradient step size μ . The noise environment is model by a stable distribution with $\alpha = 1.5$ and GSNR = 20 dB

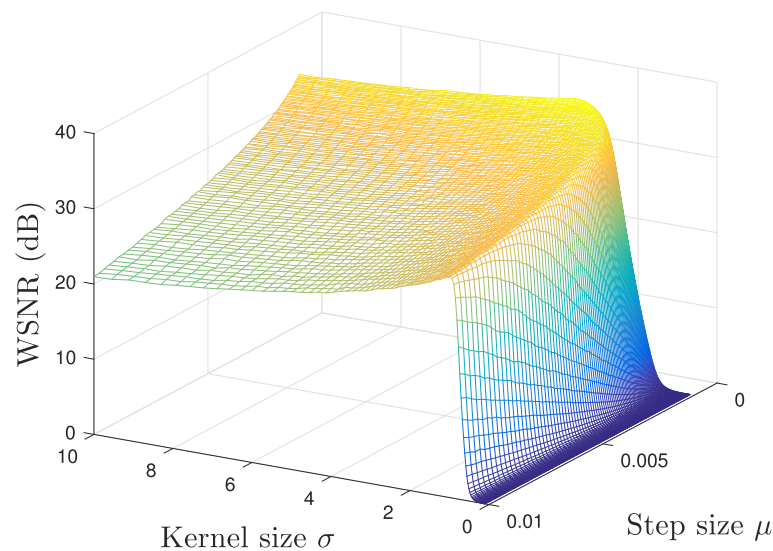


Fig. 8 WSNR performance of the CVKW algorithm as a function of the kernel size parameter σ and the gradient step size μ . The noise environment is model by a stable distribution with $\alpha = 1.5$ and GSNR = 20 dB

AMCCC, (31) for the SMCCC, and (36) for the CVKW. The choice of this value had showed also important in the methods performance. Also, since all methods presented in this paper are based in the optimization using a gradient ascendant, the analysis of the step size choice in the algorithm performance is relevant. Figures 6, 7, and 8 shows the WSNR performance of each proposed adaptive kernel size strategies with the MCCC as a function of the kernel size σ and step size μ . As one can notice, the performance is strict related to the choice of both the free parameters: the step μ and the kernel size σ .

In summary, the use of the complex correntropy as a cost function in a gradient ascendant strategy has shown a valid approach to deal with system identification problems in non-Gaussian noise environments, achieving better results than the classical CLMS solution. Even that the adaptive kernel size strategies could overcome the performance of the MCCC with a fixed kernel size, the dependence of free parameters is still present.

5 Conclusion

This paper has proposed a novel gradient method employing the complex correntropy as a cost function based on the Wirtinger calculus. Moreover, a convergence analysis has been provided for this gradient solution. This new solution was used in order to update the most recently adaptive kernel size algorithms reported in literature to deal with complex-valued data.

Simulations shown that, as in the real-valued case, adjusting the kernel size makes the gradient MCCC solution an effective mechanism to deal with non-Gaussian noise. Moreover, the performances of the proposed

adaptive methods, e.g., CVKW, SMCCC, and AMCCC, improve significantly the performance of the MCCC when compared with the CLMS and MCCC with fixed kernel size in a system identification problem. Future work includes investigating the application of the introduced methods to other problems such as complex-valued nonlinear adaptive filters and telecommunication with baseband signal.

Supplementary information

Supplementary information accompanies this paper at <https://doi.org/10.1186/s13634-019-0652-2>.

Additional file 1: MCCC Gradient ascendant step-by-step solution.

Abbreviations

MCCC: Maximum complex correntropy criterion; MCC: Maximum correntropy criterion; VKW-MCC: Variable kernel width-maximum correntropy criterion; AMCC: Adaptive kernel width MCC; SMCC: Switch kernel width method of correntropy; 16-QAM: Quadrature amplitude modulation; WSNR: Weight signal-to-noise ratio.

Acknowledgements

The authors would like to thank the Federal Institute of Rio Grande do Norte and Federal University of Rio Grande do Norte by technical support.

Authors' contributions

The authors declare that they all contributed to the manuscript. All authors read and approved the final manuscript.

Availability of data and materials

Data and MATLAB source code are available from the corresponding author upon request.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Federal University of Rio Grande do Norte (UFRN), Campus Universitário Lagoa Nova, 59078-970 Natal, RN, Brazil. ²Federal Institute of Education, Science and Technology of Rio Grande do Norte (IFRN), BR 405, KM 154, S/N, Chico Cajá, CEP 59900-000 Pau dos Ferros, RN, Brazil. ³Federal Institute of Education, Science and Technology of Rio Grande do Norte (IFRN), BR 406, Km 73, nº 3500, Perímetro Rural, 59550-000 João Câmara, RN, Brazil.

Received: 8 October 2018 Accepted: 23 October 2019

Published online: 08 November 2019

References

1. I. Santamaria, P. P. Pokharel, J. C. Principe, Generalized correlation function: definition, properties, and application to blind equalization. *IEEE Trans. Signal Process.* **54**(6), 2187–2197 (2006). <https://doi.org/10.1109/TSP.2006.872524>
2. Y. Wang, Y. Li, J. C. M. Bermudez, X. Han, An adaptive combination constrained proportionate normalized maximum correntropy criterion algorithm for sparse channel estimations. *EURASIP J. Adv. Sig. Process.* **2018**(1), 58 (2018)
3. R. He, W. Zheng, B. Hu, Maximum correntropy criterion for robust face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(8), 1561–1576 (2011). <https://doi.org/10.1109/TPAMI.2010.220>
4. S. Hakimi, G. Abed Hodtani, Generalized maximum correntropy detector for non-Gaussian environments. *Int. J. Adapt. Control. Sig. Process.* **32**(1), 83–97 (2018). <https://doi.org/10.1002/acs.2827>
5. A. I. R. Fontes, A. de M. Martins, L. F. Q. Silveira, J. C. Principe, Performance evaluation of the correntropy coefficient in automatic modulation classification. *Expert Syst. Appl.* **42**(1), 1–8 (2015). <https://doi.org/10.1016/j.eswa.2014.07.023>
6. S. Wang, L. Dang, W. Wang, G. Qian, C. K. Tse, Kernel adaptive filters with feedback based on maximum correntropy. *IEEE Access.* **6**, 10540–10552 (2018). <https://doi.org/10.1109/ACCESS.2018.2808218>
7. J. P. F. Guimarães, A. I. R. Fontes, J. B. A. Rego, A. de M. Martins, J. C. Principe, Complex correntropy: probabilistic interpretation and application to complex-valued data. *IEEE Signal Process. Lett.* **24**(1), 42–45 (2017). <https://doi.org/10.1109/LSP.2016.2634534>
8. J. P. Guimarães, A. I. Fontes, J. B. Rego, A. d. M. Martins, J. C. Principe, Complex correntropy function: properties, and application to a channel equalization problem. *Expert Syst. Appl.* **107**, 173–181 (2018). <https://doi.org/10.1016/j.eswa.2018.04.020>
9. A. Singh, J. C. Principe, in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, Kernel width adaptation in information theoretic cost functions, (2010), pp. 2062–2065. <https://doi.org/10.1109/ICASSP.2010.5495035>
10. W. Wang, J. Zhao, H. Qu, B. Chen, J. C. Principe, in *2015 IEEE International Conference on Digital Signal Processing (DSP)*, An adaptive kernel width update method of correntropy for channel estimation, (2015), pp. 916–920. <https://doi.org/10.1109/ICDSP.2015.7252010>
11. W. Wang, J. Zhao, H. Qu, B. Chen, J. C. Principe, in *IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, A switch kernel width method of correntropy for channel estimation, (2015), pp. 1–7. <https://doi.org/10.1109/IJCNN.2015.7280632>
12. F. Huang, J. Zhang, S. Zhang, Adaptive filtering under a variable kernel width maximum correntropy criterion. *IEEE Trans. Circuits Syst. II, Exp. Briefs.* **64**(10), 1247–1251 (2017). <https://doi.org/10.1109/TCSII.2017.2671339>
13. A. Singh, J. C. Principe, in *Acoustics Speech and Signal Processing (ICASSP)*, 2010 IEEE International Conference On. A closed form recursive solution for maximum correntropy training (IEEE, 2010), pp. 2070–2073. <https://doi.org/10.1109/icassp.2010.5495055>
14. J. P. F. Guimarães, A. I. R. Fontes, J. B. A. Rego, L. F. Q. Silveira, A. M. Martins, in *2016 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, Performance evaluation of the maximum correntropy criterion in identification systems, (2016), pp. 110–113. <https://doi.org/10.1109/EAIS.2016.7502500>
15. J. P. F. Guimarães, A. I. R. Fontes, J. B. A. Rego, A. de M. Martins, J. C. Principe, Complex correntropy function: properties, and application to a channel equalization problem. *Expert Syst. Appl.* **107**, 173–181 (2018). <https://doi.org/10.1016/j.eswa.2018.04.020>
16. D. P. Mandic, V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*, ser. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. (Wiley, 2009)
17. P. Bouboulis, S. Theodoridis, Extension of Wirtinger's calculus to reproducing kernel hilbert spaces and the complex kernel lms. *IEEE Trans. Sig. Process.* **59**(3), 964–978 (2011). <https://doi.org/10.1109/TSP.2010.2096420>
18. W. Wirtinger, Zur formalen theorie der funktionen von mehr komplexen veränderlichen. *Math. Ann.* **97**, 357–376 (1927)
19. T. Y. Al-Naffouri, A. H. Sayed, Adaptive filters with error nonlinearities: mean-square analysis and optimum design. *EURASIP J. Appl. Sig. Process.* **2001**(1), 192–205 (2001). <https://doi.org/10.1155/S1110865701000348>
20. B. Chen, L. Xing, B. Xu, H. Zhao, N. Zheng, J. C. Principe, Kernel risk-sensitive loss: definition, properties and application to robust adaptive filtering. *IEEE Trans. Sig. Process.* **65**(11), 2888–2901 (2017). <https://doi.org/10.1109/TSP.2017.2669903>
21. J. C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. (Springer, New York, 2010)
22. S. Zhao, B. Chen, J. C. Principe, in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, An adaptive kernel width update for correntropy, (2012), pp. 1–5. <https://doi.org/10.1109/IJCNN.2012.6252495>
23. W. Wang, J. Zhao, H. Qu, B. Chen, J. C. Principe, Convergence performance analysis of an adaptive kernel width mcc algorithm. *AEU - Int. J. Electron. Commun.* **76**, 71–76 (2017). <https://doi.org/10.1016/j.aeue.2017.03.028>
24. B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. (Chapman and Hall/CRC, London, 1986)
25. A. Singh, J. C. Principe, in *2009 International Joint Conference on Neural Networks*, Using correntropy as a cost function in linear adaptive filters, (2009), pp. 2950–2955. <https://doi.org/10.1109/IJCNN.2009.5178823>
26. A. Weron, R. Weron, Computer simulation of Lévy α -stable variables and processes, 379–392 (1995). https://doi.org/10.1007/3-540-60188-0_67
27. C. L. Nikias, M. Shao, *Signal processing with alpha-stable distributions and applications*. (Wiley-Interscience, New York, 1995)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)