The Complex Gaussian Kernel LMS algorithm

Pantelis Bouboulis and Sergios Theodoridis

Department of Informatics and Telecommunications,
University of Athens, Athens, Greece.
{bouboulis, stheodor}@di.uoa.gr

Abstract. Although the real reproducing kernels are used in an increasing number of machine learning problems, complex kernels have not, yet, been used, in spite of their potential interest in applications such as communications. In this work, we focus our attention on the complex gaussian kernel and its possible application in the complex Kernel LMS algorithm. In order to derive the gradients needed to develop the complex kernel LMS (CKLMS), we employ the powerful tool of Wirtinger's Calculus, which has recently attracted much attention in the signal processing community. Writinger's calculus simplifies computations and offers an elegant tool for treating complex signals. To this end, the notion of Writinger's calculus is extended to include complex RKHSs. Experiments verify that the CKLMS offers significant performance improvements over the traditional complex LMS or Widely Linear complex LMS (WL-LMS) algorithms, when dealing with nonlinearities.

Key words: Kernel Methods, LMS, Reproducing Kernel Hilbert Spaces, Complex Kernels, Wirtinger Calculus, Kernels

1 Introduction

In recent years, kernel based algorithms have become the state of the art for many problems, especially in the machine learning community. The common feature of these problems is that they are casted as optimization problems over a Reproducing Kernel Hilbert Space (RKHS). The main advantage of mobilizing the tool of RKHSs is that the original nonlinear task is "transformed" into a linear one, where one can employ an easier "algebra". Moreover, different types of nonlinearities can be treated in a unifying way, that does not affect the derivation of the algorithms, except at the final implementation stage. The main concepts of this procedure can be summarized in the following two steps: 1) Map the finite dimensionality input data from the input space F (usually $F \subset \mathbb{R}^{\nu}$) into a higher dimensionality (possibly infinite) RKHS $\mathcal{H}$ and 2) Perform a linear processing (e.g., adaptive filtering) on the mapped data in $\mathcal{H}$. The procedure is equivalent with a non-linear processing (non-linear filtering) in F.

An alternative way of describing this process is through the popular *kernel trick* [1], [2]: "Given an algorithm, which is formulated in terms of dot products, one can construct an alternative algorithm by replacing each one of the dot products with a positive definite kernel κ". The specific choice of kernel, implicitly, defines a RKHS with an appropriate inner product. Furthermore, the choice of a kernel also defines the type of nonlinearity that underlies the model to be used. Although there are several kernels available in the relative literature, in most cases the powerful real Gaussian kernel is adopted.

The main representatives of this class of algorithms are the celebrated *support vector machines* (SVMs), which have dominated the research in machine learning over the last decade. Moreover, processing in Reproducing Kernel Hilbert Spaces (RKHSs) in the context of online adaptive processing is also gaining in popularity within the signal processing community [3], [4], [5], [6], [7]. Besides SVMs and the more recent applications in adaptive filtering, there is a plethora of other scientific domains that have gained from adopting kernel methods (e.g., image processing and denoising [8], [9], principal component analysis [10], clustering [11], e.t.c.).

Although the real Gaussian RBF kernel is quite popular in the aforementioned context, the existence of the corresponding complex Gaussian kernel is relatively unknown to the machine learning community. This is partly due to the fact, that in classification tasks (which is the dominant application of kernel methods) the use of complex kernels is prohibitive, since no arrangement can be derived in complex domains and the necessary separating hyperplane of SVMs cannot be defined. Consequently, all known kernel based applications, since they emerged from the specific background, use real-valued kernels and they are able to deal with real valued data sequences only. While the complex gaussian RBF kernel is known to the mathematicians (especially those working on Reproducing Kernel Hilbert Spaces or Functional Analysis), it has remained in obscurity in the machine learning society. In this paper, however, we use the complex gaussian kernel to address the problem of adaptive filtering of complex signals in RKHSs, focusing on the recently developed Kernel LMS (KLMS) [3], [12]. The main goals of this paper are: a) to elevate from obscurity the complex Gaussian kernel as an effective tool for kernel based adaptive processing of complex signals, b) the extension of *Wirtinger's Calculus* in complex RKHSs as a means for the elegant and efficient computation of the gradients, that are involved in many adaptive filtering algorithms, and c) the development of the Complex Kernel LMS (CKLMS) algorithm, by exploiting the extension of Wirtinger's calculus and the RKHS of complex gaussian kernels. Wirtinger's calculus [13] is enjoying increasing popularity, recently, mainly in the context of *Widely Linear* complex adaptive filters [14], [15], [16], [17], [18], providing a tool for the derivation of gradients in the complex domain.

The paper is organized as follows. In section 2 we provide a minimal introduction to complex RKHSs focusing on the complex gaussian kernel and its relation with the real one. Next, in section 3 we summarize the main notions of the extended Wirtinger's Calculus. Section 4 presents the gaussian complex kernel LMS algorithm. Finally, experimental results and conclusions are provided in Section 5. We will denote the set of all real and complex numbers by $\mathbb{R}$ and $\mathbb{C}$ respectively. Vector or matrix valued quantities appear in boldfaced symbols.

2 Reproducing Kernel Hilbert Spaces

In this section we briefly describe the Reproducing Kernel Hilbert Spaces. Since we are mainly interested on the complex case, we recall the basic facts on RKHS associated with complex kernels. The material presented here may be found with more details in [19] and [20]. Given a function $\kappa : X \times X \to \mathbb{C}$ and $x_1, \ldots, x_N \in X$, the matrix[1] $K = (K_{i,j})^N$ with elements $K_{i,j} = \kappa(x_i, x_j)$, for $i, j = 1, \ldots, N$, is called the *Gram matrix* (or *kernel matrix*) of κ with respect to $x_1, \ldots, x_N$. A complex hermitian matrix $K = (K_{i,j})^N$ satisfying

$$c^H \cdot K \cdot c = \sum_{i=1,j=1}^{N,N} c_i^* c_j K_{i,j} \geq 0,$$

for all $c_i \in \mathbb{C}$, $i = 1, \ldots, N$, is called *Positive Definite*[2]. Let X be a nonempty set. Then a function $\kappa : X \times X \to \mathbb{C}$, which for all $N \in \mathbb{N}$ and all $x_1, \ldots, x_N \in X$ gives rise to a positive definite Gram matrix K is called a *Positive Definite Kernel*. In the following we will frequently refer to a positive definite kernel simply as *kernel*.

Next, consider a linear class $\mathcal{H}$ of complex valued functions f defined on a set X. Suppose further, that in $\mathcal{H}$ we can define an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ with corresponding norm $\| \cdot \|_{\mathcal{H}}$ and that $\mathcal{H}$ is complete with respect to that norm, i.e., $\mathcal{H}$ is a Hilbert space. We call $\mathcal{H}$ a *Reproducing Kernel Hilbert Space (RKHS)*, if for all $x \in X$ the evaluation functional $T_x : \mathcal{H} \to \mathbb{C} : \ T_x(f) = f(x)$ is a continuous (or, equivalently, bounded) operator. If this is true, then by the Riesz's representation theorem, for all $x \in X$ there is a function $g_x \in \mathcal{H}$ such that $T_x(f) = f(x) = \langle f, g_x \rangle_{\mathcal{H}}$. The function $\kappa : X \times X \to \mathbb{C} : \ \kappa(y, x) = g_x(y)$ is called a *reproducing kernel* of $\mathcal{H}$. It can be easily proved that the function κ is a positive definite kernel.

Alternatively, we can define a RKHS as a Hilbert space $\mathcal{H}$ for which there exists a function $\kappa : X \times X \to \mathbb{C}$ with the following two properties:

1. For every $x \in X$, $\kappa(\cdot, x)$ belongs to $\mathcal{H}$.
2. κ has the so called *reproducing property*, i.e.

$$f(x) = \langle f, \kappa(\cdot, x) \rangle_{\mathcal{H}}, \text{ for all } f \in \mathcal{H}, \tag{1}$$

in particular $\kappa(x, y) = \langle \kappa(\cdot, y), \kappa(\cdot, x) \rangle_{\mathcal{H}}$.

It has been proved (see [21]) that to every positive definite kernel κ there corresponds one and only one class of functions $\mathcal{H}$ with a uniquely determined inner product in it, forming a Hilbert space and admitting κ as a reproducing kernel. In fact the kernel κ produces the entire space $\mathcal{H}$, i.e., $\mathcal{H} = \overline{\text{span}\{\kappa(x, \cdot) | x \in X\}}$. The map $\Phi : X \to \mathcal{H} : \Phi(x) = \kappa(\cdot, x)$ is called the *feature map* of $\mathcal{H}$. Recall, that in the case of complex Hilbert spaces the inner product is sesqui-linear and

[1] The term $(K_{i,j})^N$ denotes a square $N \times N$ matrix.

[2] In matrix analysis literature, this is the definition of the positive semidefinite matrix.

Hermitian. In the real case the condition $\kappa(x,y) = \langle\kappa(\cdot,y),\kappa(\cdot,x)\rangle_{\mathcal{H}}$ may be replaced by the well known equation $\kappa(x,y) = \langle\kappa(\cdot,x),\kappa(\cdot,y)\rangle_{\mathcal{H}}$. However, since in the complex case the inner product is Hermitian, the aforementioned condition is equivalent to $\kappa(x,y) = (\langle\kappa(\cdot,x),\kappa(\cdot,y)\rangle_{\mathcal{H}})^*$.

Consider the complex valued function

$$\kappa_{\sigma,\mathbb{C}^d}(\boldsymbol{z},\boldsymbol{w}) := \exp\left(-\frac{\sum_{i=1}^d (z_i - w_i^*)^2}{\sigma^2}\right), \tag{2}$$

defined on $\mathbb{C}^d \times \mathbb{C}^d$, where $\boldsymbol{z},\boldsymbol{w} \in \mathbb{C}^d$, z_i denotes the i-th component of the complex vector $\boldsymbol{z} \in \mathbb{C}^d$ and exp is the extended exponential function in the complex domain. It can be shown that $\kappa_{\sigma,\mathbb{C}^d}$ is a $\mathbb{C}$-valued kernel on $\mathbb{C}^d$, which we call the *complex Gaussian kernel* with parameter σ. Its restriction $\kappa_\sigma := \left(\kappa_{\sigma,\mathbb{C}^d}\right)_{|\mathbb{R}^d\times\mathbb{R}^d}$ is the well known *real Gaussian kernel*:

$$\kappa_{\sigma,\mathbb{R}^d}(\boldsymbol{x},\boldsymbol{y}) := \exp\left(-\frac{\sum_{i=1}^d (x_i - y_i)^2}{\sigma^2}\right). \tag{3}$$

An explicit description of the RKHSs of these kernels, together with some important properties can be found in [22].

3 Wirtinger's Calculus in complex RKHS

Wirtinger's calculus [13] has become very popular in the signal processing community mainly in the context of complex adaptive filtering [14], [23], [15], [16], [24], as a means of computing, in an elegant way, gradients of real valued cost functions defined on complex domains ($\mathbb{C}^\nu$). The Cauchy-Riemann conditions dictate that such functions are not holomorphic and therefore the complex derivative cannot be used. Instead, if we consider that the cost function is defined on a Euclidean domain with a double dimensionality ($\mathbb{R}^{2\nu}$), then the real derivatives may be employed. The price of this approach is that the computations become cumbersome and tedious. Wirtinger's calculus provides an alternative equivalent formulation, that is based on simple rules and principles and which bears a great resemblance to the rules of the standard complex derivative. A self-consistent presentation of the main ideas of Wirtinger's calculus may be found in the excellent and highly recommended introductory report of K. Kreutz-Delgado [25].

In the case of a simple non-holomorphic complex function T defined on $U \subseteq \mathbb{C}$, Wirtinger's calculus considers two forms of derivatives, the $\mathbb{R}$-*derivative* and the *conjugate* $\mathbb{R}$-*derivative*, which are defined as follows:

$$\frac{\partial T}{\partial z} = \frac{1}{2}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \frac{i}{2}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right),$$
$$\frac{\partial T}{\partial z^*} = \frac{1}{2}\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) + \frac{i}{2}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)$$

where $T(z) = T(x + iy) = T(x, y) = u(x, y) + iv(x, y)$. Note that any such non-holomorphic function can be written in the form $T(z, z^*)$. Having this in mind, $\frac{\partial T}{\partial z}$, can be easily evaluated as the standard complex partial derivative taken with respect to z (thus treating z^* as a constant). Consequently, $\frac{\partial T}{\partial z^*}$ is evaluated as the standard complex partial derivative taken with respect to z^* (thus treating z as a constant). For example, if $T(z, z^*) = z(z^*)^2$, then $\frac{\partial T}{\partial z} = (z^*)^2$, $\quad \frac{\partial T}{\partial z^*} = 2zz^*$. Similar principles and rules hold for a function of many complex variables (i.e., $U \subseteq \mathbb{C}^\nu$) [25].

Wirtinger's calculus has been developed only for operators defined on finite dimensional spaces, $\mathbb{C}^\nu$. Hence, this calculus cannot be used in RKH spaces, where the dimensionality of the function space can be infinite, as, for example, it is the case for the Gaussian RKHSs. To this end, Wirtinger's calculus needs to be generalized to a general Hilbert space. A rigorous presentation of this extension is out of the scope of the paper (due to lack of space). Nevertheless, we will present the main ideas and results. We employ the Fréchet derivative, a notion that generalizes differentiability on abstract Banach or Hilbert spaces. Consider a Hilbert space H over the field F (typically $\mathbb{R}$ or $\mathbb{C}$). The operator $T : H \to F$ is said to be *Fréchet differentiable* at f_0, if there exists a $u \in H$, such that

$$\lim_{\|h\|_H \to 0} \frac{T(f_0 + h) - T(f_0) - \langle u, h \rangle_H}{\|h\|_H} = 0, \tag{4}$$

where $\langle \cdot, \cdot \rangle_H$ is the dot product of the Hilbert space H and $\| \cdot \|_H = \sqrt{\langle \cdot, \cdot \rangle_H}$ is the induced norm. The element u is usually called the gradient of T at f_0.

Assume that $\boldsymbol{T} = (T_1, T_2)^T$, $\boldsymbol{T}(\boldsymbol{f}) = \boldsymbol{T}(f_1 + if_2) = \boldsymbol{T}(f_1, f_2) = T_1(f_1, f_2) + iT_2(f_1, f_2)$, is differentiable as an operator defined on the RKHS $\mathcal{H}$ and let $\nabla_1 T_1$, $\nabla_2 T_1$, $\nabla_1 T_2$ and $\nabla_2 T_2$ be the partial derivatives, with respect to the first (f_1) and the second (f_2) variable respectively. It turns out, proofs are omitted due to lack of space, that if $\boldsymbol{T}(f_1, f_2)$ has derivatives of any order, then it can be written in the form $\boldsymbol{T}(\boldsymbol{f}, \boldsymbol{f}^*)$, where $\boldsymbol{f}^* = f_1 - if_2$, so that for fixed $\boldsymbol{f}^*$, $\boldsymbol{T}$ is $\boldsymbol{f}$-holomorphic and for fixed $\boldsymbol{f}$, T is $\boldsymbol{f}^*$-holomorphic. We may define the $\mathbb{R}$-derivative and the conjugate $\mathbb{R}$-derivative of $\boldsymbol{T}$ as follows:

$$\nabla_{\boldsymbol{f}} \boldsymbol{T} = \frac{1}{2}\left(\nabla_1 T_1 + \nabla_2 T_2\right) + \frac{i}{2}\left(\nabla_1 T_2 - \nabla_2 T_1\right) \tag{5}$$

$$\nabla_{\boldsymbol{f}^*} \boldsymbol{T} = \frac{1}{2}\left(\nabla_1 T_1 - \nabla_2 T_2\right) + \frac{i}{2}\left(\nabla_1 T_2 + \nabla_2 T_1\right). \tag{6}$$

The following properties can be proved (among others):

1. The first order Taylor expansion around $\boldsymbol{f} \in \mathcal{H}$ is given by

$$\boldsymbol{T}(\boldsymbol{f} + \boldsymbol{h}) = \boldsymbol{T}(\boldsymbol{f}) + \langle \boldsymbol{h}, (\nabla_{\boldsymbol{f}} \boldsymbol{T}(\boldsymbol{f}))^* \rangle_{\mathcal{H}} + \langle \boldsymbol{h}^*, (\nabla_{\boldsymbol{f}^*} \boldsymbol{T}(\boldsymbol{f}))^* \rangle_{\mathcal{H}}.$$

2. If $\boldsymbol{T}(\boldsymbol{f}) = \langle \boldsymbol{f}, \boldsymbol{w} \rangle_{\mathcal{H}}$, then $\nabla_{\boldsymbol{f}} \boldsymbol{T} = \boldsymbol{w}^*$, $\nabla_{\boldsymbol{f}^*} \boldsymbol{T} = \boldsymbol{0}$.
3. If $\boldsymbol{T}(\boldsymbol{f}) = \langle \boldsymbol{w}, \boldsymbol{f} \rangle_{\mathcal{H}}$, then $\nabla_{\boldsymbol{f}} \boldsymbol{T} = \boldsymbol{0}$, $\nabla_{\boldsymbol{f}^*} \boldsymbol{T} = \boldsymbol{w}$.
4. If $\boldsymbol{T}(\boldsymbol{f}) = \langle \boldsymbol{f}^*, \boldsymbol{w} \rangle_{\mathcal{H}}$, then $\nabla_{\boldsymbol{f}} \boldsymbol{T} = \boldsymbol{0}$, $\nabla_{\boldsymbol{f}^*} \boldsymbol{T} = \boldsymbol{w}^*$.

5. If $\boldsymbol{T}(\boldsymbol{f}) = \langle \boldsymbol{w}, \boldsymbol{f}^* \rangle_{\mathcal{H}}$, then $\nabla_{\boldsymbol{f}} \boldsymbol{T} = \boldsymbol{w}$, $\nabla_{\boldsymbol{f}^*} \boldsymbol{T} = \boldsymbol{0}$.

An important consequence of the above properties is that if $\boldsymbol{T}$ is a real valued operator defined on $\mathcal{H}$, then its first order Taylor's expansion is given by:

$$\begin{aligned}\boldsymbol{T}(\boldsymbol{f}+\boldsymbol{h}) &= \boldsymbol{T}(\boldsymbol{f}) + \langle \boldsymbol{h}, (\nabla_{\boldsymbol{f}} \boldsymbol{T}(\boldsymbol{f}))^* \rangle_{\mathcal{H}} + \langle \boldsymbol{h}^*, (\nabla_{\boldsymbol{f}^*} \boldsymbol{T}(\boldsymbol{f}))^* \rangle_{\mathcal{H}} \\ &= \boldsymbol{T}(\boldsymbol{f}) + \langle \boldsymbol{h}, \nabla_{\boldsymbol{f}^*} \boldsymbol{T}(\boldsymbol{f}) \rangle_{\mathcal{H}} + (\langle \boldsymbol{h}, \nabla_{\boldsymbol{f}^*} \boldsymbol{T}(\boldsymbol{f}) \rangle_{\mathcal{H}})^* \\ &= \boldsymbol{T}(\boldsymbol{f}) + 2 \cdot \Re \left[\langle \boldsymbol{h}, \nabla_{\boldsymbol{f}^*} \boldsymbol{T}(\boldsymbol{f}) \rangle_{\mathcal{H}} \right].\end{aligned}$$

However, in view of the Cauchy Riemann inequality we have:

$$\Re \left[\langle \boldsymbol{h}, \nabla_{\boldsymbol{f}^*} \boldsymbol{T}(\boldsymbol{f}) \rangle_{\mathcal{H}} \right] \leq |\langle \boldsymbol{h}, \nabla_{\boldsymbol{f}^*} \boldsymbol{T}(\boldsymbol{f}) \rangle_{\mathcal{H}}| \leq \|\boldsymbol{h}\|_{\mathcal{H}} \cdot \|\nabla_{\boldsymbol{f}^*} \boldsymbol{T}(\boldsymbol{f})\|_{\mathcal{H}}.$$

The equality in the above relationship holds if $h \propto \nabla_{\boldsymbol{f}^*} \boldsymbol{T}$. Hence, the direction of increase of $\boldsymbol{T}$ is $\nabla_{\boldsymbol{f}^*} \boldsymbol{T}(\boldsymbol{f})$. Therefore, any gradient descent based algorithm minimizing $\boldsymbol{T}(\boldsymbol{f})$ is based on the update scheme:

$$\boldsymbol{f}_n = \boldsymbol{f}_{n-1} - \mu \cdot \nabla_{\boldsymbol{f}^*} \boldsymbol{T}(\boldsymbol{f}_{n-1}). \tag{7}$$

4 Complex Kernel LMS

As an application of the complex gaussian kernel in adaptive filtering of complex signals, we focus on the recently developed *Kernel Least Mean Squares Algorithm* (KLMS), which is the LMS algorithm in RKHSs [3], [12]. KLMS, as all the known kernel methods that use real-valued kernels, was developed for real valued data sequences only. Here, the KLMS is extended to include the complex case. To our knowledge, no kernel-based strategy has been developed, so far, that is able to effectively deal with complex valued signals. Wirtinger's calculus is exploited to derive the necessary gradient updates.

Consider the sequence of examples $(\boldsymbol{z}(1), d(1)), (\boldsymbol{z}(2), d(2)), \ldots, (\boldsymbol{z}(N), d(N))$, where $d(n) \in \mathbb{C}$, $\boldsymbol{z}(n) \in V \subset \mathbb{C}^\nu$, $\boldsymbol{z}(n) = \boldsymbol{x}(n) + i\boldsymbol{y}(n)$, $\boldsymbol{x}(n), \boldsymbol{y}(n) \in \mathbb{R}^\nu$, for $n = 1, \ldots, N$. We map the points $\boldsymbol{z}(n)$ to the gaussian complex RKHS $\mathcal{H}$ using the feature map $\varPhi$, for $n = 1, \ldots, N$. The objective of the complex Kernel LMS is to minimize $E\left[\mathcal{L}_n(\boldsymbol{w})\right]$, where

$$\begin{aligned}\mathcal{L}_n(\boldsymbol{w}) &= |e(n)|^2 = |d(n) - \langle \varPhi(\boldsymbol{z}(n)), \boldsymbol{w} \rangle_{\mathbb{H}}|^2 \\ &= \left(d(n) - \langle \varPhi(\boldsymbol{z}(n)), \boldsymbol{w} \rangle_{\mathbb{H}}\right) \left(d(n) - \langle \varPhi(\boldsymbol{z}(n)), \boldsymbol{w} \rangle_{\mathbb{H}}\right)^* \\ &= \left(d(n) - \langle \boldsymbol{w}^*, \varPhi(\boldsymbol{z}(n)) \rangle_{\mathbb{H}}\right) \left(d(n)^* - \langle \boldsymbol{w}, \varPhi(\boldsymbol{z}(n)) \rangle_{\mathbb{H}}\right),\end{aligned}$$

at each instance n. We then apply the complex LMS to the transformed data, using the rules of Wirtinger's calculus to compute the gradient $\nabla_{\boldsymbol{w}^*} \mathcal{L}_n(\boldsymbol{w}) = -e(n)^* \cdot \varPhi(\boldsymbol{z}(n))$. Therefore the CKLMS update rule becomes $\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \mu e(n)^* \cdot \varPhi(\boldsymbol{z}(n))$, where $\boldsymbol{w}(n)$ denotes the estimate at iteration n.

Assuming that $\boldsymbol{w}(0) = \boldsymbol{0}$, the repeated application of the weight-update equation gives:

$$\boldsymbol{w}(n) = \sum_{k=1}^{n} e(k)^* \varPhi(\boldsymbol{z}(k)). \tag{8}$$

Thus, the filter output at iteration n becomes:

$$\begin{aligned}\hat{d}(n) =& \langle \Phi(\boldsymbol{z}(n)), \boldsymbol{w}(n-1)\rangle_{\mathbb{H}} = \mu \sum_{k=1}^{n-1} e(k) \langle \Phi(\boldsymbol{z}(n)), \Phi(\boldsymbol{z}(k))\rangle_{\mathbb{H}} \\ =& \mu \sum_{k=1}^{n-1} e(k) \kappa_{\sigma,\mathbb{C}^\nu}(\boldsymbol{z}(n), \boldsymbol{z}(k)).\end{aligned}$$

It can readily be shown that, since the CKLMS is the complex LMS in RKHS, the important properties of the LMS (convergence in the mean, misadjustment, e.t.c.) carry over to CKLMS. Furthermore, note that using the complex gaussian kernel the algorithm is automatically normalized. The CKLMS algorithm is summarized in Algorithm 1. Although it is developed in the context of the complex gaussian kernel, it may be used with any other complex reproducing kernel.

Algorithm 1 Normalized Complex Kernel LMS

```
INPUT: $(\boldsymbol{z}(1), d(1)), \dots, (\boldsymbol{z}(N), d(N))$
OUTPUT: The expansion
$\boldsymbol{w} = \sum_{k=1}^{N} a(k)\kappa(\boldsymbol{z}(k), \cdot)$.

Initialization: Set $\boldsymbol{a} = \{\}$, $\boldsymbol{Z} = \{\}$ (i.e., $\boldsymbol{w} = \boldsymbol{0}$). Select the step parameter $\mu$ and the parameter $\sigma$ of the complex gaussian kernel.
  for n=1:N do
    Compute the filter output: $\hat{d}(n) = \sum_{k=1}^{n-1} a(k) \cdot \kappa_{\sigma,\mathbb{C}^\nu}(\boldsymbol{z}(n), \boldsymbol{z}(k))$.
    Compute the error: $e(n) = d(n) - \hat{d}(n)$.
    $a(n) = \mu e(n)$.
    Add the new center $\boldsymbol{z}(n)$ to the list of centers, i.e., add $\boldsymbol{z}(n)$ to the list $\boldsymbol{Z}$, add $a(n)$ to the list $\boldsymbol{a}$.
  end for
```

In CKLMS, we start from an empty set (usually called the *dictionary*) and gradually add new samples to that set, to form a summation similar to the one shown in equation (8). This results to an increasing memory and computational requirements, as time evolves. To cope with this problem and to produce sparse solutions, we employ the well known *novelty criterion* [26], [12]. In novelty criterion online sparsification, whenever a new data pair $(\Phi(\boldsymbol{z}_n), d_n)$ is considered, a decision is immediately made of whether to add the new center $\Phi(\boldsymbol{z}_n)$ to the dictionary of centers $\mathcal{C}$. The decision is reached following two simple rules. First, the distance of the new center $\Phi(\boldsymbol{z}_n)$ from the current dictionary is evaluated: $dis = \min_{\boldsymbol{c}_k \in \mathcal{C}}\{\|\Phi(\boldsymbol{z}_n) - \boldsymbol{c}_k\|_{\mathbb{H}}\}$. If this distance is smaller than a given threshold δ_1 (i.e., the new center is close to the existing dictionary), then the center is not added to $\mathcal{C}$. Otherwise, we compute the prediction error $e_n = d_n - \hat{d}_n$. If $|e_n|$ is smaller than a predefined threshold δ_2, then the new center is discarded. Only if $|e_n| \geq \delta_2$ the new center $\Phi(\boldsymbol{z}_n)$ is added to the dictionary.

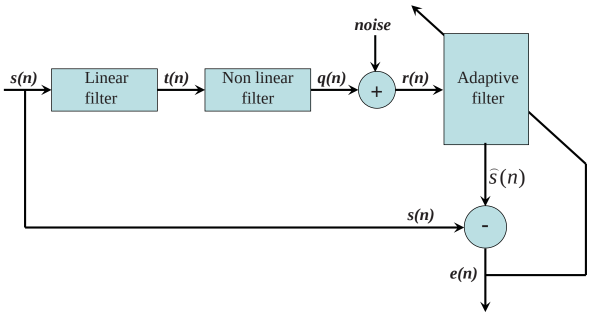


Fig. 1. The equalization problem.

5 Experiments

We tested the CKLMS using a simple nonlinear channel equalization problem (see figure 1). The nonlinear channel consists of a linear filter: $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1)$ and a memoryless nonlinearity $q(n) = t(n) + (0.1 + 0.15i) \cdot t^2(n) + (0.06 + 0.05i) \cdot t^3(n)$. At the receiver end of the channel, the signal is corrupted by white Gaussian noise and then observed as $r(n)$. The input signal that was fed to the channel had the form

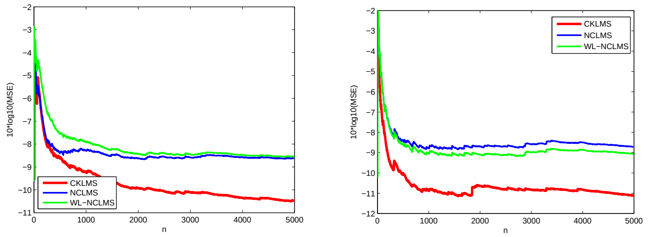
$$s(n) = 0.70(\sqrt{1-\rho^2}X(n) + i\rho Y(n)), \tag{9}$$

where $X(n)$ and $Y(n)$ are gaussian random variables. This input is circular for $\rho = \sqrt{2}/2$ and highly non-circular if ρ approaches 0 or 1 [15]. The aim of channel equalization is to construct an inverse filter which taking the output $r(n)$, reproduces the original input signal with as low an error rate as possible. To this end we apply the NCKLMS algorithm to the set of samples

$$\left((r(n+D), r(n+D-1), \dots, r(n+D-L))^T, s(n)\right),$$

where $L > 0$ is the filter length and D the equalization time delay.

Experiments were conducted on a set of 5000 samples of the input signal (9) considering both the circular and the non-circular case. The results are compared with the NCLMS and the WL-NCLMS algorithms. In all algorithms the step update parameter μ is tuned for best possible results. Time delay D was also set for optimality. Figure 2 shows the learning curves of the NCKLMS with $\sigma = 5$, compared with the NCLMS and the WL-NCLMS algorithms. Novelty criterion was applied to the CKLMS for sparsification with $\delta_1 = 0.1$ and $\delta_2 = 0.2$. In both examples, CKLMS considerably outperforms both the NCLMS and the WL-NCLMS algorithms. However, this enhanced behavior comes at a price in computational complexity, since the CKLMS requires the evaluation of the kernel function on a growing number of training examples.



(a) (b)

Fig. 2. Learning curves for KCLMS, ($\mu = 1$) CLMS ($\mu = 1/16$) and WL-CLMS ($\mu = 1/16$) (filter length $L = 5$, delay $D = 2$) in the nonlinear channel equalization, for the (a) circular input case and (b) the non-circular input case.

References

1. B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, 2002.
2. S. Theodoridis and K. Koutroumbas, *Pattern Recognition, 4th edition*, Academic Press, 2009.
3. W. Liu, P. Pokharel, and J. C. Principe, "The kernel least-mean-square algorithm," *IEEE Trans. Sign. Proc.*, vol. 56, no. 2, pp. 543–554, 2008.
4. J. Kivinen, A. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Sign. Proc.*, vol. 52, no. 8, pp. 2165–2176, 2004.
5. Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Sign. Proc.*, vol. 52, no. 8, 2004.
6. K. Slavakis, S. Theodoridis, and I. Yamada, "On line classification using kernels and projection based adaptive algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2781–2797, 2008.
7. K. Slavakis, S. Theodoridis, and I. Yamada, "Adaptive constrained learning in reproducing kernel hilbert spaces: The robust beamforming case," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4744–4764, 2009.
8. K. Kim, M. O. Franz, and B. Scholkopf, "Iterative kernel principal component analysis for image modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1351–1366, 2005.
9. P. Bouboulis, K. Slavakis, and S. Theodoridis, "Adaptive kernel-based image denoising employing semi-parametric regularization," *IEEE Trans. Image Process.*, (to appear).
10. A. J. Smola B. Schölkopf and K. R. Muller, "Kernel principal component analysis," *Lecture notes in computer science*, vol. 1327, pp. 583–588, 1997.
11. M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering. pattern recognition," *Pattern Recognition*, vol. 41, no. 1, pp. 176–190, 2008.
12. W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering*, Wiley, 2010.
13. W. Wirtinger, "Zur formalen theorie der functionen von mehr complexen veränderlichen," *Math. Ann.*, vol. 97, pp. 357–375, 1927.
14. B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Trans. Signal Process.*, vol. 43, no. 8, pp. 2030–2033, 1995.

15. T. Adali and H. Li, *Complex-valued adaptive signal processing*, Adaptive Signal Processing: Next Generation Solutions, T. Adali and S. Haykin, editors. Hoboken, NJ, Wiley, 2010.
16. T. Adali, H. Li, M. Novey, and J.F. Cardoso, "Complex ICA using nonlinear functions," *IEEE Trans. Signal Process.*, vol. 56, no. 9, pp. 4536–4544, 2008.
17. D. Mattera, L. Paura, and F. Sterle, "Widely linear decision-feedback equalizer for time-dispersive linear MIMO channels," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2525–2536, 2005.
18. J. Navarro-Moreno, "ARMA prediction of widely linear systems by using the innovations algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3061–3068, 2008.
19. S. Saitoh, *Integral Transforms, Reproducing Kernels and their applications*, Longman Scientific & Technical, Harlow, 1997.
20. V. I. Paulsen, "An introduction to the theory of reproducing kernel hilbert spaces," `http://www.math.uh.edu/~vern/rkhs.pdf`.
21. N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.
22. Ingo Steinwart, Don Hush, and Clint Scovel, "An explicit description of the reproducing kernel hilbert spaces of gaussian rbf kernels," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4635–4643, 2006.
23. D. Mandic and V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters*, Wiley, 2009.
24. A. S. Cacciapuoti, G. Gelli, L. Paura, and F. Verde, "Widely linear versus linear blind multiuser detection with subspace-based channel estimation: Finite sample-size effects," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1426–1443, 2009.
25. K. Kreutz-Delgado, "The complex gradient operator and the $\mathbb{CR}$-calculus," `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.6515&rep=rep1&type=pdf`.
26. J. Platt, "A resourse allocating network for function interpolation," *Newral Computation*, vol. 3, no. 2, pp. 213–225, 1991.