

The Augmented Complex Kernel LMS

Pantelis Bouboulis, *Member, IEEE*, and Sergios Theodoridis, *Fellow, IEEE* and Michael Mavroforakis

Abstract—Recently, a unified framework for adaptive kernel based signal processing of complex data was presented by the authors, which, besides offering techniques to map the input data to complex Reproducing Kernel Hilbert Spaces, developed a suitable Wirtinger-like Calculus for general Hilbert Spaces. In this short paper, the extended Wirtinger's calculus is adopted to derive complex kernel-based widely-linear estimation filters. Furthermore, we illuminate several important characteristics of the widely linear filters. We show that, although in many cases the gains from adopting widely linear estimation filters, as alternatives to ordinary linear ones, are rudimentary, for the case of kernel based widely linear filters significant performance improvements can be obtained.

I. INTRODUCTION

Kernel-based processing is gaining in popularity within the Signal Processing community [?], [?], [?], [?], [?], [?], [?], as it provides an efficient toolbox for treating non-linear problems. The main advantage of this procedure is that it transforms the original non-linear task, in a low dimensional space, into a linear one, that is performed in a higher dimensionality (possible infinite Hilbert) space \mathcal{H} . This is equivalent with solving a nonlinear problem in the original space. The space \mathcal{H} is implicitly chosen via a kernel function that defines the associated inner product.

However, most of the kernel-based techniques were designed to process real data. Until recently, no kernel-based methodology for treating complex signals had been developed, in spite of their potential interest in a number of applications. In [?], a framework based on complex RKHS was presented to solve this problem. Its main contributions are: a) the development of a wide framework that allows real-valued kernel algorithms to be extended to treat complex data efficiently, taking advantage of a technique called *complexification* of real RKHSs, b) the elevation from obscurity of the pure complex kernels (such as the complex Gaussian one) as a tool for kernel based adaptive processing of complex signals and c) the extension of *Wirtinger's Calculus* in complex RKHSs as a means for an elegant and efficient computation of the gradients, which are involved in the derivation of adaptive learning algorithms.

Complex-valued signals arise frequently in applications as diverse as communications, biomedicine, radar, etc. The complex domain not only provides a convenient and elegant representation for these signals, but also a natural way to preserve their characteristics and to handle transformations that need to be performed. In the more traditional setting, treating complex signals is often followed by (implicitly) assuming the *circularity* of the signal. Circularity is intimately related to the rotation in the geometric sense. A complex random variable Z is called circular, if for any angle ϕ both Z and $Ze^{i\phi}$ (i.e., the rotation of Z by angle ϕ) follow the same probability distribution [?], [?]. Naturally, this assumption limits the area for applications, since many practical signals exhibit non-circular

characteristics. Thus, following the ideas originated by Picinbono and Chevalier in [?], [?], on-going research is focusing on the *widely linear* filters (or *augmented* filters) in the complex domain (see for example [?], [?], [?], [?], [?], [?], [?], [?], [?], [?]). The main characteristic of such filters is that they exploit simultaneously both the original signal as well as its conjugate analogue. An important characteristic of widely linear estimation is that it captures the full second-order statistics of a given complex signal, especially in the case where the signal is non-circular, by considering both the covariance and the pseudo-covariance matrices [?], [?]. Thus, for a general complex signal, the optimal linear filter (i.e., widely linear) is linear both in z and z^* .

The main contribution of the present paper is twofold. First, we explore the main concepts of widely linear estimation from a new perspective, showing why widely linear estimation can potentially lead to improved performance compared to complex linear estimation. Moreover, we employ the framework of [?] to develop widely linear adaptive filters in complex RKHS, to solve nonlinear filtering tasks. Thus, we introduce, for the first time, the notion of widely linear filters in infinite dimensional Hilbert spaces. Our findings indicate that the “natural” choice for kernels, in the context of the widely linear filtering structure, are the pure complex kernels. In contrast, combining the widely linear structure with kernels that result from complexification of real kernels, does not enhance performance, compared to that obtained with the complexified kernel-based linear filters¹.

The problem of widely linear estimation using kernels has been also considered in [?], [?]. Although in these papers the authors claim that they solve the problem of augmented complex signal processing using kernels, a careful study reveals a lot of points that are not clearly described (possibly because they are addressed to a conference). Moreover (and this is more important) there are several significant inconsistencies with the traditional theory of Reproducing Kernel Hilbert spaces (RKHS), which is the main mathematical tool of any kernel based method. For example, in both papers the complex data are mapped to the so called feature space using a mapping $\Phi : \mathbb{C}^n \rightarrow \mathbb{C}^d$, instead of the traditional approach, which considers a feature map of the form $\Phi : \mathbb{C}^n \rightarrow \mathbb{H}$, $\Phi(z) = \kappa(\cdot, z)$, where \mathbb{H} is the RKHS induced by the kernel κ . It is evident, that under this context, Φ maps the data to an Euclidean space. Following this mapping of the data, the presented methods build some kernel matrices of the form $(\Phi(x_1), \dots, \Phi(x_M))$. However, the case of treating the problem in finite dimensional Euclidean spaces, is a rather trivial extension of previously known results. Moreover, in those papers, the notion of fully complex kernels is neither mentioned and even more treated. However, as we prove in the present paper, this is the case where the augmented structure has a meaning, in the sense that can offer advantages. Using real kernels, under the complexification theory, in the context of augmented filters, offers no advantages.

The paper is organized as follows. We start with a brief introduction to complex RKHSs and Wirtinger's Calculus in Section II. In Section III, we describe the concept of widely linear estimation and show why

¹This is because complexification implicitly adds a conjugate component to the adopted model.

this is better than complex linear estimation. Finally, widely linear kernel based adaptive filters are described in section IV. Experiments are provided in section V. Section VI concludes the paper.

II. PRELIMINARIES

Throughout the paper, we will denote the set of all integers, real and complex numbers by \mathbb{N} , \mathbb{R} and \mathbb{C} respectively. Vector or matrix valued quantities appear in boldfaced symbols. A RKHS [?] is a Hilbert space \mathcal{H} over a field \mathbb{F} for which there exists a positive definite function $\kappa : X \times X \rightarrow \mathbb{F}$ with the following two important properties: a) For every $x \in X$, $\kappa(\cdot, x)$ belongs to \mathcal{H} and b) κ has the so called *reproducing property*, i.e., $f(x) = \langle f, \kappa(\cdot, x) \rangle_{\mathcal{H}}$, for all $f \in \mathcal{H}$, in particular $\kappa(x, y) = \langle \kappa(\cdot, y), \kappa(\cdot, x) \rangle_{\mathcal{H}}$. The map $\Phi : X \rightarrow \mathcal{H} : \Phi(x) = \kappa(\cdot, x)$ is called the *feature map* of \mathcal{H} .

Although the underlying theory has been developed by the mathematicians for general complex reproducing kernels and their associated RKHSs, mostly the real kernels have been considered by the machine learning and signal processing communities [?], [?]. Some of the most widely used kernel in the literature are the *Gaussian RBF*, i.e., $\kappa_{\sigma, \mathbb{R}^d}(\mathbf{x}, \mathbf{y}) := \exp\left(-\frac{\sum_{k=1}^d (x_k - y_k)^2}{\sigma^2}\right)$, defined for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, where σ is a free positive parameter and the *polynomial kernel*: $\kappa_d(\mathbf{x}, \mathbf{y}) := (1 + \mathbf{x}^T \mathbf{y})^d$, for $d \in \mathbb{N}$. Many more can be found in [?], [?]. Complex reproducing kernels, that have been extensively studied by the mathematicians, are, among others, the *Szego kernels* and the *Bergman kernels*. Another important complex kernel is the *complex Gaussian kernel*, which is defined as: $\kappa_{\sigma, \mathbb{C}^d}(\mathbf{z}, \mathbf{w}) := \exp\left(-\frac{\sum_{k=1}^d (z_k - w_k^*)^2}{\sigma^2}\right)$, where $\mathbf{z}, \mathbf{w} \in \mathbb{C}^d$, z_k denotes the k -th component of the complex vector $\mathbf{z} \in \mathbb{C}^d$ and $\exp(\cdot)$ is the extended exponential function in the complex domain.

To generate kernel adaptive filtering algorithms on complex domains, according to [?], one can adopt two methodologies. A first straightforward approach is to use directly a complex RKHS, using one of the complex kernels given in section II and map the original data to the complex RKHS through the associated feature map $\Phi(x) = \kappa(\cdot, x)$. Another alternative technique is to use real kernels through a rationale that is called *complexification* of real RKHSs. This method has the advantage of allowing modeling in complex RKHSs using popular well-established and well understood, from a performance point of view, real kernels (e.g., gaussian, polynomial, e.t.c.). In the first case, we map the data directly to the complex RKHS, using the corresponding complex feature map $\Phi(\mathbf{z}) = \kappa_{\mathbb{C}}(\cdot, \mathbf{z})$, while in the complexification scenario we employ the map $\hat{\Phi}(\mathbf{z}) = \Phi(\mathbf{x}, \mathbf{y}) + i\Phi(\mathbf{x}, \mathbf{y})$, where $\mathbf{z} = \mathbf{x} + i\mathbf{y}$, and Φ is the feature map of the chosen real kernel $\kappa_{\mathbb{R}}$, i.e., $\Phi(\mathbf{x}, \mathbf{y}) = \kappa_{\mathbb{R}}(\cdot, (\mathbf{x}, \mathbf{y}))$.

In order to compute the gradients of real valued cost functions, that are defined on complex domains, we adopt the rationale of Wirtinger's calculus [?]. This was brought into light recently [?], [?], [?], [?], [?], as a means to compute, in an efficient and elegant way, gradients of real valued cost functions that are defined on complex domains (\mathbb{C}^{ν}). It is based on simple rules and principles, which bear a great resemblance to the rules of the standard complex derivative, and it greatly simplifies the calculations of the respective derivatives. The difficulty with real valued cost functions is that they do not obey the Cauchy-Riemann conditions and are not differentiable in the complex domain. The alternative to Wirtinger's calculus would be to consider the complex variables as pairs of two real ones and employ the common real partial derivatives. However, this approach, usually, is more time consuming and leads to more cumbersome expressions. In [?], the notion of Wirtinger's calculus was extended to general complex Hilbert spaces, providing the tool to compute the gradients that are needed to develop kernel-based algorithms for treating complex data. In the same paper, the aforementioned toolbox was

employed in the context of the complex LMS and two realizations of the complex kernel LMS algorithm were developed. The first one, which is denoted as NCKLMS1 adopts the complexification procedure and the second one, which is denoted as NCKLMS2, uses the complex gaussian kernel².

III. WIDELY LINEAR ESTIMATION FILTERS

As mentioned in the introduction, ongoing research in complex signal processing is mainly focused on the so called *widely linear* estimation filters, or *augmented* filters, as they are also known. These are filters that take into account both the original values of the signal data and their conjugates. For example, in a typical LMS task, we estimate the output as $\hat{d}(n) = \mathbf{w}^H \mathbf{z}$ and the step update as $\mathbf{w}(n) = \mathbf{w}(n-1) + \mu e^*(n) \mathbf{z}(n)$. In this case, $\hat{d}(n)$ is a linear estimation filter. However, the linearity property is taken with respect to the field of complex numbers. Picinbono and Chevalier, in [?], proposed an alternative approach. They estimated the filter's output as $\tilde{d}(n) = \mathbf{w}^H \mathbf{z} + \mathbf{v}^H \mathbf{z}^*$ and showed that it provides better results in terms of the mean square error. This, of course, is expected since $\tilde{d}(n)$ provides a more rich representation than $\hat{d}(n)$. On the other hand, $\tilde{d}(n)$ is no longer linear over the field \mathbb{C} . It is linear, however, over the real numbers \mathbb{R} . To emphasize this difference, in the relative literature, $\hat{d}(n)$ is often called \mathbb{C} -linear, while $\tilde{d}(n)$ is called \mathbb{R} -linear.

In [?], [?], [?], it is shown that the widely linear estimation filter is able to capture the second order statistical characteristics of the signal which are essential, for non-circular sources. Although in many relative works this is highlighted as the main reason for adopting widely linear techniques, in this paper, we will highlight a different perspective.

Our starting point will be the definition of linearity. Prior to it, it should be clarified, that complex processing is equivalent with processing two real signals in the respective Euclidean (Hilbert) spaces. The advantage of using complex algebra is that the algorithm and/or the solution may be described in a more compact form. Moreover, the complex algebra allows for a more intuitive understanding of the problem, as many geometric transformations can be easily described using complex algebra in an elegant way. Finally, the application of Wirtinger's calculus greatly simplifies the calculations needed for the gradients of real valued cost functions.

Having this in mind, we now turn our attention to a typical complex LMS task. Let $\mathbf{z}(n) \in \mathbb{C}^{\nu}$ and $d(n) \in \mathbb{C}$ be the input and the output of the original filter. We estimate the output of the filter using a \mathbb{C} -linear response $\hat{d}(n) = \mathbf{w}^H \mathbf{z}(n)$. The typical complex LMS task aims to compute $\mathbf{w} \in \mathbb{C}^{\nu}$, such that the error $E[|d(n) - \hat{d}(n)|^2]$ is minimized. If we set $\mathbf{w} = \mathbf{w}_r + i\mathbf{w}_i$ and $\mathbf{z} = \mathbf{x} + i\mathbf{y}$, we take that

$$\hat{d}(n) = \mathbf{w}_r^T \mathbf{x} + \mathbf{w}_i^T \mathbf{y} + i(\mathbf{w}_r^T \mathbf{y} - \mathbf{w}_i^T \mathbf{x}). \quad (1)$$

However, the real essence behind a complex filter operation is the following: Given two real vectors, $\mathbf{x}(n)$ and $\mathbf{y}(n)$, compute linear filters in order to estimate two real processes, $d_r(n)$ and $d_i(n)$, in an optimal way, that jointly cares for both $d_r(n)$ and $d_i(n)$. Let us express the problem in its multichannel formulation, using real variables only, i.e.,

$$\begin{pmatrix} \bar{d}_r(n) \\ \bar{d}_i(n) \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{1,1}^T & \mathbf{u}_{1,2}^T \\ \mathbf{u}_{2,1}^T & \mathbf{u}_{2,2}^T \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \equiv \mathbf{U} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$$

From a mathematical point of view, this is the definition of a linear operator from $\mathbb{R}^{2\nu} \rightarrow \mathbb{R}^2$. The elements of \mathbf{U} are computed such that both $|d_r(n) - \bar{d}_r(n)|^2$ and $|d_i(n) - \bar{d}_i(n)|^2$ are jointly minimized. This leads to the so called Dual Real Channel (DRC) formulation.

²Due to the fact that there are a lot of algorithms mentioned in the paper, Table I presents their acronyms for the reader's convenience.

Thus, we take the relations $\bar{d}_r(n) = \mathbf{u}_{1,1}^T \mathbf{x} + \mathbf{u}_{1,2}^T \mathbf{y}$ and $\bar{d}_i(n) = \mathbf{u}_{2,1}^T \mathbf{x} + \mathbf{u}_{2,2}^T \mathbf{y}$. In complex notation, if we consider that $\bar{d}(n) = \bar{d}_r(n) + i\bar{d}_i(n)$, we obtain the expression

$$\bar{d}(n) = \mathbf{u}_{1,1}^T \mathbf{x} + \mathbf{u}_{1,2}^T \mathbf{y} + i(\mathbf{u}_{2,1}^T \mathbf{x} + \mathbf{u}_{2,2}^T \mathbf{y}). \quad (2)$$

It is easy to see that the DRC approach expressed by relation (2) adopts a richer representation than that of the traditional LMS in (1). Moreover, it takes a few lines of elementary algebra to show that an equivalent expression of (2) is the widely linear estimation filter, i.e., $\bar{d}(n) = \tilde{d}(n)$. This will be our kick off point to define the task in a general Hilbert space. In general, we can prove the following:

Proposition 1. Consider a real³ Hilbert space \mathcal{H} , the real Hilbert space \mathcal{H}^2 and the complex Hilbert space $\mathbb{H} = \mathcal{H} + i\mathcal{H}$. Then any continuous linear function $\mathbf{T} : \mathcal{H}^2 \rightarrow \mathbb{R}^2$ can be expressed in complex notation as

$$\mathbf{T}(\mathbf{x}, \mathbf{y}) = \mathbf{T}(\mathbf{x} + i\mathbf{y}) = \mathbf{T}(\mathbf{z}) = \langle \mathbf{z}, \mathbf{w} \rangle_{\mathbb{H}} + \langle \mathbf{z}^*, \mathbf{v} \rangle_{\mathbb{H}}, \quad (3)$$

for some $\mathbf{w}, \mathbf{v} \in \mathbb{H}$, where $\langle \cdot, \cdot \rangle_{\mathbb{H}}$ is the respective inner product of \mathbb{H} .

Proof: Recall that any element of \mathcal{H}^2 or \mathbb{R}^2 can be expressed either as a vector or as a complex element (complex notation). Any function $\mathbf{T} : \mathcal{H}^2 \rightarrow \mathbb{R}^2$ takes the form $\mathbf{T}(\mathbf{x}, \mathbf{y}) = (T_1(\mathbf{x}, \mathbf{y}), T_2(\mathbf{x}, \mathbf{y}))^T$. As \mathbf{T} is linear, both T_1, T_2 are also linear. Thus, as a consequence of the Riesz's representation theorem, there is $\mathbf{u}_1 = (\mathbf{u}_{1,1}^T, \mathbf{u}_{1,2}^T)^T \in \mathcal{H}^2$, such that $T_1(\mathbf{x}, \mathbf{y}) = \langle (\mathbf{x}, \mathbf{y})^T, (\mathbf{u}_{1,1}^T, \mathbf{u}_{1,2}^T)^T \rangle_{\mathcal{H}^2}$. Similarly, $T_2(\mathbf{x}, \mathbf{y}) = \langle (\mathbf{x}, \mathbf{y})^T, (\mathbf{u}_{2,1}^T, \mathbf{u}_{2,2}^T)^T \rangle_{\mathcal{H}^2}$, for some $\mathbf{u}_2 = (\mathbf{u}_{2,1}^T, \mathbf{u}_{2,2}^T)^T \in \mathcal{H}^2$.

Thus, in complex notation, \mathbf{T} can be written as

$$\begin{aligned} \mathbf{T}(\mathbf{x}, \mathbf{y}) &= T_1(\mathbf{x}, \mathbf{y}) + iT_2(\mathbf{x}, \mathbf{y}) \\ &= \langle \mathbf{x}, \mathbf{u}_{1,1} \rangle_{\mathcal{H}} + \langle \mathbf{y}, \mathbf{u}_{1,2} \rangle_{\mathcal{H}} + i(\langle \mathbf{x}, \mathbf{u}_{2,1} \rangle_{\mathcal{H}} + \langle \mathbf{y}, \mathbf{u}_{2,2} \rangle_{\mathcal{H}}). \end{aligned} \quad (4)$$

We define $\mathbf{w}_1, \mathbf{w}_2, \mathbf{v}_1, \mathbf{v}_2 \in \mathcal{H}$ as follows: $\mathbf{w}_1 = \frac{\mathbf{u}_{1,1} + \mathbf{u}_{2,2}}{2}$, $\mathbf{w}_2 = \frac{\mathbf{u}_{1,2} - \mathbf{u}_{2,1}}{2}$, $\mathbf{v}_1 = \frac{\mathbf{u}_{1,1} - \mathbf{u}_{2,2}}{2}$, $\mathbf{v}_2 = -\frac{\mathbf{u}_{2,1} + \mathbf{u}_{1,2}}{2}$. Then, $\mathbf{u}_{1,1} = \mathbf{w}_1 + \mathbf{v}_1$, $\mathbf{u}_{1,2} = \mathbf{w}_2 - \mathbf{v}_2$, $\mathbf{u}_{2,2} = \mathbf{w}_1 - \mathbf{v}_1$ and $\mathbf{u}_{2,1} = -\mathbf{w}_2 - \mathbf{v}_2$. Substituting in (4) we take:

$$\begin{aligned} \mathbf{T}(\mathbf{z}) &= \mathbf{T}(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{w}_1 \rangle_{\mathcal{H}} + \langle \mathbf{x}, \mathbf{v}_1 \rangle_{\mathcal{H}} + \langle \mathbf{y}, \mathbf{w}_2 \rangle_{\mathcal{H}} - \langle \mathbf{y}, \mathbf{v}_2 \rangle_{\mathcal{H}} \\ &\quad + i(-\langle \mathbf{x}, \mathbf{w}_2 \rangle_{\mathcal{H}} - \langle \mathbf{x}, \mathbf{v}_2 \rangle_{\mathcal{H}} + \langle \mathbf{y}, \mathbf{w}_1 \rangle_{\mathcal{H}} - \langle \mathbf{y}, \mathbf{v}_1 \rangle_{\mathcal{H}}) \\ &= \langle \mathbf{x} + i\mathbf{y}, \mathbf{w}_1 + i\mathbf{w}_2 \rangle_{\mathbb{H}} + \langle \mathbf{x} - i\mathbf{y}, \mathbf{v}_1 + i\mathbf{v}_2 \rangle_{\mathbb{H}} \\ &= \langle \mathbf{z}, \mathbf{w} \rangle_{\mathbb{H}} + \langle \mathbf{z}^*, \mathbf{v} \rangle_{\mathbb{H}}, \end{aligned}$$

where $\mathbf{z} = \mathbf{x} + i\mathbf{y}$, $\mathbf{w} = \mathbf{w}_1 + i\mathbf{w}_2$, $\mathbf{v} = \mathbf{v}_1 + i\mathbf{v}_2$. \square

Remark 1. In view of proposition 1, one understands that the original formulation of the complex LMS was rather “unorthodox”, as it excludes a large class of linear functions from being considered in the estimation process. It is evident, that the linearity with respect to the field of complex numbers is restricted, compared to the linearity that underlies the DRC approach, which is more natural. Thus, the correct complex linear estimation is $\mathbf{T}(\mathbf{z}) = \langle \mathbf{z}, \mathbf{w} \rangle_{\mathbb{H}} + \langle \mathbf{z}^*, \mathbf{v} \rangle_{\mathbb{H}}$ rather than $\mathbf{T}(\mathbf{z}) = \langle \mathbf{z}, \mathbf{w} \rangle_{\mathbb{H}}$.

Remark 2. Any \mathbb{R} -linear function can be expressed as in (3).

Remark 3. Let $e(n) = d(n) - \tilde{d}(n)$, be the error of the widely linear estimation, where $d(n)$ is the desired response. If we assume that the error follows the complex normal distribution of van den Boss [?],

³By the term real (complex) Hilbert space, we mean a Hilbert space over the field of real numbers \mathbb{R} (complex numbers \mathbb{C}).

with covariance matrix $V = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, we can easily derive that the maximum likelihood estimator, for \mathbf{w}, \mathbf{v} is equivalent with minimizing the square error, as it is the case in widely linear LMS.

IV. WIDELY LINEAR ESTIMATION IN COMPLEX RKHS

In this section, we will develop realizations of the *Augmented Complex Kernel LMS* (ACKLMS) algorithm using either pure complex kernels, or real kernels under the complexification trick. We show that ACKLMS offers substantial improvements versus complex kernel LMS (CKLMS), when the complex gaussian kernel is employed. On the other hand, the ACKLMS, which is developed under the complexification trick, degenerates to the standard CKLMS.

Consider the sequence of examples $(\mathbf{z}(1), d(1)), (\mathbf{z}(2), d(2)), \dots, (\mathbf{z}(N), d(N))$, where $d(n) \in \mathbb{C}$, $\mathbf{z}(n) \in \mathbb{C}^\nu$, $\mathbf{z}(n) = \mathbf{x}(n) + i\mathbf{y}(n)$, $\mathbf{x}(n), \mathbf{y}(n) \in \mathbb{R}^\nu$, for $n = 1, \dots, N$. Consider, also, a real RKHS \mathcal{H} , the real Hilbert space \mathcal{H}^2 and the complex Hilbert space $\mathbb{H} = \mathcal{H} + i\mathcal{H}$. We model the widely linear estimation filter in \mathbb{H} as:

$$\tilde{d}(n) = \langle \Phi(\mathbf{z}(n)), \mathbf{w}(n-1) \rangle_{\mathbb{H}} + \langle \Phi^*(\mathbf{z}(n)), \mathbf{v}(n-1) \rangle_{\mathbb{H}}, \quad (5)$$

where Φ is an appropriate function that maps the input data to the feature space \mathbb{H} . This is equivalent with transforming the data to a complex RKHS and applying a widely linear complex LMS to the transformed data. The objective of the ACKLMS is to estimate, \mathbf{w} and \mathbf{v} , so that to minimize $E[\mathcal{L}_n(\mathbf{w})]$, where $\mathcal{L}_n(\mathbf{w}) = |e(n)|^2 = |d(n) - \tilde{d}(n)|^2$, at each time instance n .

Applying the rules of Wirtinger's calculus in complex RKHS, we can easily deduce that $\frac{\partial \mathcal{L}_n(\mathbf{w})}{\partial \mathbf{w}^*} = -\Phi(\mathbf{z}(n)) \cdot e^*(n)$, $\frac{\partial \mathcal{L}_n(\mathbf{w})}{\partial \mathbf{v}^*} = -\Phi^*(\mathbf{z}(n)) \cdot e^*(n)$. Thus, the step updates of the ACKLMS are $\mathbf{w}(n) = \mathbf{w}(n-1) + \mu \Phi(\mathbf{z}(n))e^*(n)$, $\mathbf{v}(n) = \mathbf{v}(n-1) + \mu \Phi^*(\mathbf{z}(n))e^*(n)$. Assuming that $\mathbf{w}(0) = \mathbf{v}(0) = \mathbf{0}$, the repeated application of the weight-update equations gives:

$$\mathbf{w}(n-1) = \mu \sum_{k=0}^{n-1} \Phi(\mathbf{z}(k))e^*(k), \quad (6)$$

$$\mathbf{v}(n-1) = \mu \sum_{k=0}^{n-1} \Phi^*(\mathbf{z}(k))e^*(k). \quad (7)$$

Combining (5), (6) and (7) leads us to conclude that the filter's output, at iteration n , becomes:

$$\begin{aligned} \tilde{d}(n) &= \mu \sum_{k=1}^{n-1} e(k) \langle \Phi(\mathbf{z}(n)), \Phi(\mathbf{z}(k)) \rangle_{\mathbb{H}} \\ &\quad + \mu \sum_{k=1}^{n-1} e(k) \langle \Phi^*(\mathbf{z}(n)), \Phi^*(\mathbf{z}(k)) \rangle_{\mathbb{H}}. \end{aligned}$$

Recall that in the complexification trick the associated function that maps the data to \mathbb{H} is given by $\tilde{\Phi}(\mathbf{z}) = \Phi(\mathbf{x}, \mathbf{y}) + i\Phi(\mathbf{x}, \mathbf{y}) = \kappa_{\mathbb{R}}(\cdot, (\mathbf{x}, \mathbf{y})) + i\kappa_{\mathbb{R}}(\cdot, (\mathbf{x}, \mathbf{y}))$, where Φ is the feature map of \mathcal{H} and $\kappa_{\mathbb{R}}$ its respective real kernel. Under this condition, the filter output at iteration n takes the form

$$\tilde{d}(n) = 4\mu \sum_{k=1}^{n-1} e(k) \cdot \kappa_{\mathbb{R}}((\mathbf{x}(k), \mathbf{y}(k)), (\mathbf{x}(n), \mathbf{y}(n))). \quad (8)$$

This is exactly the same formula, which was obtained in the case of the complexified complex kernel LMS (except a rescaling). Thus, we deduce that, in this case, the standard complexified NCKLMS1 presented in [?] and the complexified augmented CKLMS are identical.

Name	Description
NCLMS	Normalized Complex LMS
NACLMS	Normalized Augmented Complex LMS (Widely-Linear)
NCKLMS1	Normalized Complex Kernel LMS (complexified real kernels)
NCKLMS2	Normalized Complex Kernel LMS (pure complex kernels)
NACKLMS	Normalized Augmented Complex Kernel LMS (pure complex kernels)
CNGD	Complex Non-linear Gradient Descent [?]
MLP	Multi Layer Perceptron (50 nodes in the hidden layer)

TABLE I
ACRONYMS OF THE ALGORITHMS USED IN THE PAPER.

On the other hand, for the case of a pure complex kernel $\kappa_{\mathbb{C}}$, the filter output becomes

$$\tilde{d}(n) = \mu \sum_{k=1}^{n-1} e(k) \kappa_{\mathbb{C}}(\mathbf{z}(k), \mathbf{z}(n)) + \mu \sum_{k=1}^{n-1} e(k) \kappa_{\mathbb{C}}^*(\mathbf{z}(k), \mathbf{z}(n)), \quad (9)$$

as $\langle \Phi(\mathbf{z}), \Phi(\mathbf{c}) \rangle_{\mathbb{H}} = \kappa_{\mathbb{C}}(\mathbf{z}, \mathbf{c})$, for any $\mathbf{z}, \mathbf{c} \in \mathbb{C}^{\nu}$. In this case, it is evident that the augmented CKLMS (ACKLMS) will result to a different solution compared to that of NCKLMS2, as it exploits a richer representation. From (9), we deduce that the complexity of the normalised ACKLMS is of the same order as the complexity of NCKLMS2 and NCKLMS1.

V. EXPERIMENTS

The performance of the normalized augmented CKLMS (NACKLMS) has been tested in the context of a nonlinear channel equalization task. As in [?], two nonlinear channels have been considered. The first channel (labeled as *soft nonlinear channel* in the figures) 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 $q(n) = t(n) + (0.1 + 0.15i) \cdot t^2(n) + (0.06 + 0.05i) \cdot t^3(n)$. The second one (labeled as *strong nonlinear channel* in the figures) comprises of the consists of the linear filter: $t(n) = (-0.9 + 0.8i) \cdot s(n) + (0.6 - 0.7i) \cdot s(n-1) + (-0.4 + 0.3i) \cdot s(n-2) + (0.3 - 0.2i) \cdot s(n-3) + (-0.1i - 0.2i) \cdot s(n-4)$ and the nonlinearity: $q(n) = t(n) + (0.2 + 0.25i) \cdot t^2(n) + (0.08 + 0.09i) \cdot t^3(n)$. At the receiver end of the channels, the signal is corrupted by white Gaussian noise and then observed as $r(n)$. The level of the noise was set to 15dB. The input signal that was fed to the channels had the form $s(n) = 0.70 \left(\sqrt{1 - \rho^2} X(n) + i\rho Y(n) \right)$, 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 [?]. To solve the channel equalization task, we apply the NACKLMS algorithm to the set of samples $((r(n+D), r(n+D-1), \dots, r(n+D-L+1)), s(n))$, where $L > 0$ is the filter length and D the equalization time delay.

Experiments were conducted on 100 sets of 5000 samples of the input signal considering both the circular and the non-circular cases. The results are compared with the NCLMS and the NACLMS (i.e., Normalized Augmented CLMS, or widely linear NCLMS as it is sometimes called) algorithms as well as two adaptive nonlinear algorithms: a) the CNGD algorithm [?] and a Multi Layer Perceptron (MLP) with 50 nodes in the hidden layer [?]. In both cases, the complex tanh activation function was employed. Figure 1 shows the learning curves of the pure complex kernel LMS (labeled as NCKLMS2 in the figures) [?] and the proposed NACKLMS, using the complex Gaussian kernel (with $\sigma = 10$), together with those

obtained from the NCLMS and the WL-NCLMS algorithms. Figure 2 shows the learning curves of NCKLMS2 and NACKLMS, using the complex Gaussian kernel (with $\sigma = 15$), versus the CNGD and the L-50-1 MLP for the hard non-linear channel.

The novelty criterion (see [?], [?]) was used for the sparsification of the NCKLMS2 and NACKLMS with $\delta_1 = 0.1$ and $\delta_2 = 0.2$. In both examples, NACKLMS considerably outperforms the linear, widely linear (i.e., NCLMS and WL-NCLMS) and nonlinear (CNGD and MLP) algorithms (see figures 1, 2). The NACKLMS also exhibits improved performance compared to the NCKLMS2 for non-circular input sources. Moreover, observe that while the gain of the NACLMS against NCLMS is rather rudimentary (smaller than 0.2 dB), the gains of NACKLMS over NCKLMS2 are significant (approximately 2dB). For circular signals, the two models (NCKLMS2 and NACKLMS) lead to almost identical results, as expected [?], [?].

VI. CONCLUSIONS

In this paper we proposed a method for widely linear estimation in complex RKHS, based on the framework presented in [?]. The developed rationale was applied to derive the augmented complex kernel LMS (NACKLMS). Moreover, some important properties of widely linear estimation were discussed to provide a further understanding of the underlying toolbox. Experiments of the developed NACKLMS, for both circular and non-circular input data, showed a significant decrease in the steady state mean square error, compared with other known linear, widely linear and nonlinear techniques. Both theoretical and experimental results showed the importance of employing augmented filters versus linear ones in the complex domain.

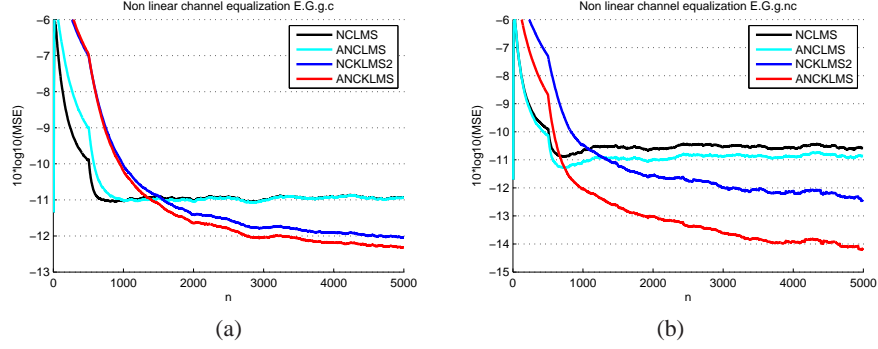


Fig. 1. Learning curves for NCKLMS2 ($\mu = 1/8$), NACKLMS, ($\mu = 1/8$, $\sigma = 10$), CLMS ($\mu = 1/16$) and widely linear CLMS ($\mu = 1/16$) (filter length $L = 5$, delay $D = 2$) for the soft nonlinear channel equalization problem, for (a) the circular input case, (b) the non-circular input case ($\rho = 0.1$).

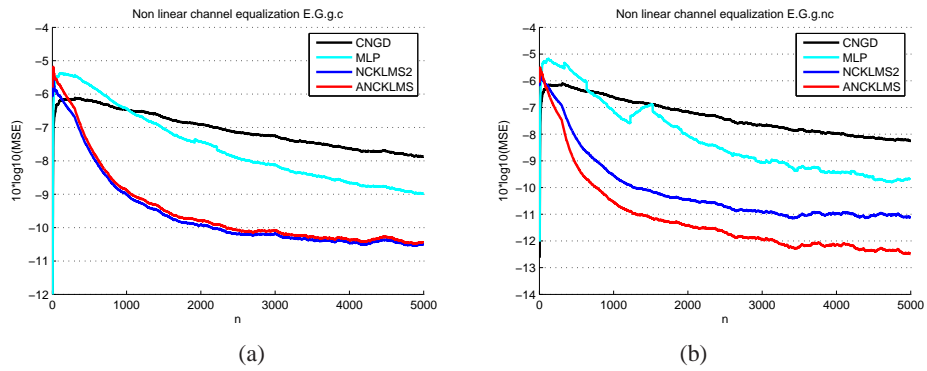


Fig. 2. Learning curves for NCKLMS2 ($\mu = 1/8$), NACKLMS, ($\mu = 1/8$, $\sigma = 15$), MLP ($\mu = 0.0003$) and CNGD ($\mu = 0.0005$) (filter length $L = 5$, delay $D = 2$) for the hard nonlinear channel equalization problem, for (a) the circular input case, (b) the non-circular input case ($\rho = 0.1$).