Kernel-Based APA Adaptive Filters for Complex Data

Tokunbo Ogunfunmi, *Senior Member IEEE* and Thomas Paul, *Member IEEE*
Santa Clara University, Santa Clara, CA 95053, USA
E-mail: togunfunmi@scu.edu, tpaul@scu.edu

***Abstract*— Kernel-based adaptive filters present a new opportunity to re-cast nonlinear optimization problems over a RKHS and transform the original nonlinear task into a linear one, where one may employ linear and well-known adaptive algorithms. It also allows different types of nonlinearities to be treated in a unifying way. This method has been used in the machine learning community for some time. Now, its use in adaptive filters elevates the subject of adaptive filtering theory to a new level, presenting a new designing methodology of nonlinear adaptive filters.**

We have recently applied the complex Gaussian kernel to develop Affine Projection (AP)-based complex algorithms for Kernel Adaptive Filtering. In this paper, we now consider the performance of these algorithms for different input signal distributions, specifically the circularity of the complex input. Recent work has shown the use of widely-linear filtering leads to use of complete second-order statistical data when performing mean square estimation for complex data. We also extend recently developed complex kernel-based algorithms using the idea of widely-linear (WL) estimation. Simulations are used to verify our theoretical results.

I. Introduction

For applications such as communications, image processing and biological systems, nonlinear adaptive filters can be seen to well describe the physically occurring phenomena. The performance of linear systems for these cases can be highly suboptimal, illustrating the need for understanding nonlinear filtering algorithms. For these cases, many new methods have proven useful, some of which are discussed in recent references such as [1-4], along with several recent papers.

In particular, nonlinear methods known as kernel learning algorithms have gained considerable interest. The recent book Kernel Adaptive Filters [5] describes a comprehensive, unifying introduction to online learning algorithms using Reproducing Kernel Hilbert Spaces (RKHS), introducing a new design methodology for nonlinear adaptive filters. The book serves as a useful guide to understand the practical implementation of nonlinear adaptive filters.

Recently, the complex Kernel Affine Projection Adaptive (CKAPA) algorithm [6] was developed as an extension of the Complex Kernel Least Mean Square (CKLMS) algorithm. It was derived with the new Wirtinger calculus for RKHS of [7], [8], which allowed for determining the gradient of the APA-based cost function which, although real-valued, was defined on a complex RKHS.

For both complex algorithms (CKLMS, CKAPA), the MSE performance can vary substantially, depending on the circularity of the complex input. This was observed when using the complex Gaussian kernel from [7] to implement these algorithms. The circularity of complex data is a known statistical characteristic, which essentially considers whether the input signal probability distribution (pdf) is invariant under rotation or not. An input signal whose pdf is rotation invariant is known as circular. Otherwise, the input signal is considered noncircular.

In this work, we extend the kernel-based adaptive algorithms previously developed [6], [7] considering widely-linear (WL) estimation [9], [10]. The benefits of this approach when using complex input data of various circular distributions is demonstrated using simulations. We first give a very brief overview of the new Kernel-based adaptive filter algorithms for complex-valued data recently obtained. These filters are based on the theory of Reproducing Kernel Hilbert Space (RKHS). We then describe the widely-linear (WL) estimator, and the benefits for estimation with complex data. Afterwards, the kernel-based methods are extended for WL estimation. Finally, we evaluate the performance of the algorithms developed using practical simulation applications. A discussion of the performance benefits is provided (based on the polynomial kernel), and simulations are used to verify the results.

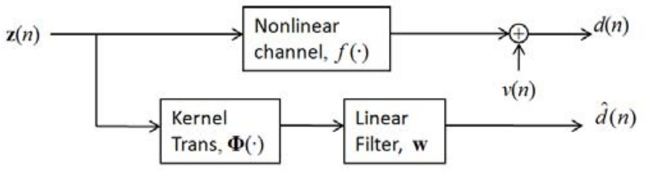
This paper is organized as follows: In Section II, we give a brief overview of the complex kernel-based adaptive algorithms. Section III describes background of widely-linear estimation. In Section IV, we extend the complex kernel-based algorithms considering WL estimation. This section also provide an analysis of the channel modeling capability of the extended algorithms considering the polynomial kernel. Simulation results for the new algorithms are provided in Section V. Conclusions and summary of results are presented in Section VI.

II. Complex Kernel-based LMS and APA

Here, we briefly summarize the Complex Kernel LMS (CKLMS) [7], and the Complex Kernel APA (CKAPA) [6] algorithms.

Both algorithms make use of a complex kernel transform to map complex input to a nonlinear space (known as a Reproducing Kernel Hilbert Space, or RKHS) where a complex nonlinear channel can be learned linearly, i.e. using an LMS or APA-based approach. Since a complex kernel is used, the associated RKHS is referred to as a complex RKHS.

A block diagram of the system (for channel estimation) is shown in Figure 1.



$\mathbf{z}(n)$
Nonlinear channel, $f(\cdot)$
$d(n)$
$v(n)$
Kernel Trans, $\mathbf{\Phi}(\cdot)$
Linear Filter, $\mathbf{w}$
$\hat{d}(n)$

Figure 1: Channel Estimation using Kernel-based Methods

The goal for both algorithms is to find a filter $\mathbf{w}$ to estimate the desired response, $d(n)$, using: $\hat{d}(n) = <\mathbf{\Phi}(\mathbf{z}(n)), \mathbf{w}>_S$, where $\mathbf{z}(n) = [z(n)...z(n\text{-}M\text{+}1)]^T$ is the complex input vector (length M) at iteration n, $\mathbf{\Phi}(.)$ is a complex kernel transform mapping, and $< . >_S$ denotes an inner product in the complex RKHS. However, the cost function to be minimized for the CKLMS, CKAPA algorithms differ and are described later.

From [6-8], notations $\mathbf{\Phi}(.)$ and $< . >_S$ may also refer to a complexified kernel transform map and complexified RKHS, respectively. However, for the analysis here we only consider the complex case. The performance of the complexified case will be considered later in the simulations section.

For the CKLMS, the cost function to minimize is the instantaneous square error, i.e. $|e(n)|^2$. The algorithm is summarized here (see [7], [8] for details).

The weights for the CKLMS algorithm are:

$$\mathbf{w}(n) = \mu \sum_{k=1}^{n} e(k)^* \cdot \mathbf{\Phi}(\mathbf{z}(k)) \tag{1}$$

and the filter output (i.e. estimate of channel output):

$$\begin{aligned}\hat{d}(n) &= <\mathbf{\Phi}(\mathbf{z}(n)), \mathbf{w}(n-1)>_S \\ &= \mu \sum_{k=1}^{n-1} e(k) <\mathbf{\Phi}(\mathbf{z}(n)), \mathbf{\Phi}(\mathbf{z}(k))>_S\end{aligned} \tag{2}$$

where $e(n) = d(n) - \hat{d}(n)$ is the estimation error, and μ is the filter update step-size.

For the CKAPA algorithhm, the cost function is based on the principle of minimum disturbance [16]. The cost function is formed with the goal of finding the minimum increment for the weights forcing the estimation error for the past K input vectors to zero. This can be written as follows:

Keeping the K most recent input vectors and observations:

$$\begin{aligned}\mathbf{X}_{ap}(n) &= \left[\mathbf{\Phi}(\mathbf{z}(n-K+1))\ \mathbf{\Phi}(\mathbf{z}(n-K+2))\ ...\ \mathbf{\Phi}(\mathbf{z}(n))\right] \\ \mathbf{d}_{ap}(n) &= \left[d(n-K+1)\ d(n-K+2)\ ...\ d(n)\right]^T\end{aligned} \tag{3}$$

The goal of the CKAPA can be written as:

$$\begin{aligned}&\mathit{Minimize}: \ \frac{1}{2}\left\|\mathbf{w}(n) - \mathbf{w}(n-1)\right\|^2 \\ &\mathit{Subject\ to}: \ \mathbf{d}_{ap}(n) - \mathbf{X}_{ap}^T(n)\mathbf{w}^*(n) = 0\end{aligned} \tag{4}$$

To obtain the required cost function, the method of Lagrange multipliers is used to convert equation (4) into an unconstrained form, which becomes:

$$\begin{aligned}F[\mathbf{w}(n)] = &\frac{1}{2}[\mathbf{w}(n) - \mathbf{w}(n-1)]^H[\mathbf{w}(n) - \mathbf{w}(n-1)] \\ &+ \mathrm{Re}\left\{\boldsymbol{\lambda}_{ap}^T(n)[\mathbf{d}_{ap}(n) - \mathbf{X}_{ap}^T(n)\mathbf{w}^*(n)]\right\}\end{aligned} \tag{5}$$

The resulting weight update algorithm, based on finding the cost function gradient, can be shown to be as follows (see [6] for details).:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu \mathbf{X}_{ap}(n)\left(\mathbf{X}_{ap}^H(n)\mathbf{X}_{ap}(n) + \varepsilon \mathbf{I}\right)^{-1} \mathbf{e}_{ap}^*(n) \cdot \tag{6}$$

where $\mathbf{e}_{ap}(n) = \mathbf{d}_{ap}(n) - \hat{\mathbf{d}}_{ap}(n)$ is the estimation error.

While the weights at iteration n (assuming w(0) = 0), can be shown to be of the following form, in general:

$$\mathbf{w}(n-1) = \sum_{j=1}^{n-1} a_j(n-1)\mathbf{\Phi}(\mathbf{z}(j)) \tag{7}$$

where $a_j(n\text{-}1)$, for j = 1...(n-1), are coefficients applied to the transformed input vectors, $\mathbf{\Phi}(\mathbf{z}(j))$ to find the weights. These coefficients are determined based on (6) (see Appendix A).

The filter output for CKAPA is:

$$\hat{\mathbf{d}}_{ap}(n) = \mathbf{X}_{ap}^T(n)\mathbf{w}^*(n-1) = \begin{bmatrix} \sum_{j=1}^{n-1} a_j^*(n-1) <\mathbf{\Phi}(\mathbf{z}(n-K+1)), \mathbf{\Phi}(\mathbf{z}(j))>_S \\ \vdots \\ \sum_{j=1}^{n-1} a_j^*(n-1) <\mathbf{\Phi}(\mathbf{z}(n-1)), \mathbf{\Phi}(\mathbf{z}(j))>_S \\ \sum_{j=1}^{n-1} a_j^*(n-1) <\mathbf{\Phi}(\mathbf{z}(n)), \mathbf{\Phi}(\mathbf{z}(j))>_S \end{bmatrix} \tag{8}$$

where $\hat{\mathbf{d}}_{ap}(n)$ is a vector of the K most recent output estimates.

For both algorithms, it was necessary to express the equations entirely in terms of inner product, $< . >_S$. This form is visible with equations (2) and (8).

Expressing these equations entirely using inner products allows for the use of kernel functions to evaluate the estimate (note complex kernel function $\kappa(\cdot,\cdot)$ defines the inner product $< . >_S$ for a complex RKHS [5-8]). This, in turn, allows for the finite computation of (2), (8), even for kernels associated with an infinite-dimension RKHS since the kernel evaluation is still of finite complexity in this case. This is known as the kernel trick [5-8]. Example kernel functions, $\kappa(\cdot,\cdot)$, will be shown later.

Next we describe the idea of widely-linear estimation, which will be used to extend these algorithms.

III. Widely-Linear Estimation

Widely-linear (WL) estimation, and its benefits for adaptive filtering, are considered in references [9-11]. Here we briefly describe the approach.

WL estimation allows for improved mean-square estimation (MS) performance with complex data. For mean-square estimation, the problem can be described as estimating some scalar random variable y, in terms of an observed random vector, $\mathbf{x}$. The estimate, $\hat{y}$, minimizing mean-square error (MSE) is the regression based on conditional expectation, $E[y|\mathbf{x}]$.

When $\mathbf{x}$ and y are jointly Gaussian and real-valued, the regression is linear in $\mathbf{x}$. However with complex-valued data, the regression is linear in $\mathbf{x}$ as well as $\mathbf{x}^*$, and is referred to as widely-linear (WL) [9].

For linear MSE (LMSE), the estimate has the form:

$$y = \mathbf{h}^H \mathbf{x} \tag{9}$$

where $\mathbf{h}$ is the optimum linear weight vector for estimation, and ^{H} denotes the Hermitian transpose.

And for the widely-linear MSE (WLMSE):

$$y' = \mathbf{h}^H \mathbf{x} + \mathbf{g}^H \mathbf{x}^* . \tag{10}$$

Equation (10) is the general form for the regression with complex data. From (10), we see estimate y' is not a linear function of $\mathbf{x}$. However the order k moments for y' are completely defined from moments of order k of $\mathbf{x}$ and $\mathbf{x}^*$, forming a type of linearity known as wide sense linear [9].

This approach can be seen to utilize the full second-order statistics of complex input $\mathbf{x}$, which can be described from two covariance matrices. The covariance matrix:

$$\mathbf{C} = E[\mathbf{x}\mathbf{x}^H] \tag{11}$$

and the pseudo-covariance matrix (also referred to as the complementary covariance matrix) [10]:

$$\mathbf{P} = E[\mathbf{x}\mathbf{x}^T] . \tag{12}$$

These statistical quantities can be seen to vary based on the circularity of the complex input.

In general, the circularity of a complex random vector, $\mathbf{x}$, is based on whether its pdf is rotation invariant (i.e. if $\mathbf{x}$ and $\mathbf{x}e^{j\theta}$ have same pdf, for all θ). In this case, the vector $\mathbf{x}$ is considered circular in the strict sense [10], or simply circular.

However, if complex vector $\mathbf{x}$ is zero-mean and $E[\mathbf{x}\mathbf{x}^T] = \mathbf{0}$, it is said to be second-order circular (or proper). Otherwise, $\mathbf{x}$ is considered noncircular (i.e. if $E[\mathbf{x}\mathbf{x}^T] \neq \mathbf{0}$).

The circularity of the complex input for MS estimation is known to affect the performance for LMSE [9], [10]. The WLMSE, however, improves the MSE performance based on use of complete second-order input statistics. Here we consider the use of WL estimation with the CKLMS, CKAPA algorithms.

IV. Extension of CKLMS and CKAPA Based On Widely-Linear Estimation

In order to combine WL estimation with the CKLMS, CKAPA algorithms, we replace the previous estimation form:

$$\hat{d}(n) = <\mathbf{\Phi}(\mathbf{z}(n)), \mathbf{w}>_S \tag{13}$$

with the estimate:

$$\hat{d}_{WL}(n) = <\mathbf{\Phi}(\mathbf{z}_{WL}(n)), \mathbf{w}_{WL}>_S \tag{14}$$

where:

$$\mathbf{z}_{WL}(n) = \begin{bmatrix} \mathbf{z}(n) \\ \mathbf{z}^*(n) \end{bmatrix}$$

and $\mathbf{w}_{WL}$ are the optimum weights for the widely linear based method. For analysis convenience (observed later) we define the function, $\xi(.)$:

$$\xi(\mathbf{z}(n)) = \mathbf{z}_{WL}(n) = [\mathbf{z}(n)\ \mathbf{z}^*(n)]^T \tag{15}$$

which will be used in place of the $\mathbf{z}_{WL}(n)$ notation.

To evaluate the weights for estimate (14) using an LMS-based approach (i.e. WL-CKLMS), we use the following weight update equation:

$$\mathbf{w}_{WL}(n) = \mathbf{w}_{WL}(n-1) + \mu e(n)\mathbf{\Phi}(\xi(\mathbf{z}(n))) \tag{16}$$

where $e(n) = d(n) - \hat{d}_{WL}(n)$ is the estimation error.

Assuming initial weights, $\mathbf{w}_{WL}(0) = 0$, at iteration n we get :

$$\mathbf{w}_{WL}(n) = \mu \sum_{k=1}^{n} e(k)\mathbf{\Phi}(\xi(\mathbf{z}(k))) \cdot \tag{17}$$

And the filter output is:

$$\begin{aligned} \hat{d}(n) &= <\mathbf{\Phi}(\xi(\mathbf{z}(n))), \mathbf{w}_{WL}(n-1)>_S \\ &= \mu \sum_{k=1}^{n-1} e(k) <\mathbf{\Phi}(\xi(\mathbf{z}(n))), \mathbf{\Phi}(\xi(\mathbf{z}(k)))>_S \end{aligned} \tag{18}$$

Equation (18) forms the WL-CKLMS algorithm.

In a similar way, considering [6], an APA-based approach (WL-CKAPA) can be seen to have the following form. Based on the K most recent input vectors and observations:

$$\mathbf{X}_{\xi,ap}(n) = \left[\mathbf{\Phi}(\xi(\mathbf{z}(n-K+1)))\ \mathbf{\Phi}(\xi(\mathbf{z}(n-K+2)))\ ...\ \mathbf{\Phi}(\xi(\mathbf{z}(n)))\right]$$

$$\mathbf{d}_{ap}(n) = \left[d(n-K+1)\ d(n-K+2)\ ...\ d(n)\right]^T$$

The weight update for WL-CKAPA may be expressed as:

$$\begin{aligned} \mathbf{w}_{WL}(n) &= \mathbf{w}_{WL}(n-1) + \\ &\mu \mathbf{X}_{\xi,ap}(n)\left(\mathbf{X}^H_{\xi,ap}(n)\mathbf{X}_{\xi,ap}(n) + \varepsilon\mathbf{I}\right)^{-1} \mathbf{e}^*_{ap}(n) \end{aligned} \tag{19}$$

The weights at iteration n, assuming w(0) = 0, can be shown to be of the following form, in general:

$$\mathbf{w}_{WL}(n-1) = \sum_{j=1}^{n-1} a_j(n-1)\mathbf{\Phi}(\xi(\mathbf{z}(j))) \tag{20}$$

where $a_j(n\text{-}1)$, for j = 1...(n-1) are coefficients determined based on (19) (see Appendix B).

The filter output for WL-CKAPA is:

$$\hat{\mathbf{d}}_{WL,ap}(n) = \mathbf{X}_{\xi,ap}^{T}(n)\mathbf{w}_{WL}^{*}(n-1) = \begin{bmatrix} \sum_{j=1}^{n-1} a_j^*(n-1) < \mathbf{\Phi}(\xi(\mathbf{z}(n-K+1))), \mathbf{\Phi}(\xi(\mathbf{z}(j))) >_S \\ \vdots \\ \sum_{j=1}^{n-1} a_j^*(n-1) < \mathbf{\Phi}(\xi(\mathbf{z}(n-1))), \mathbf{\Phi}(\xi(\mathbf{z}(j))) >_S \\ \sum_{j=1}^{n-1} a_j^*(n-1) < \mathbf{\Phi}(\xi(\mathbf{z}(n))), \mathbf{\Phi}(\xi(\mathbf{z}(j))) >_S \end{bmatrix} \tag{21}$$

where $\hat{\mathbf{d}}_{WL,ap}(n)$ is a vector of the K most recent channel output estimates.

For both methods, it can be observed that the difference between the original algorithms and the WL estimation-based algorithms is only in the kernel evaluation, which changes from:

$$\kappa(\mathbf{u},\mathbf{v}) = < \mathbf{\Phi}(\mathbf{u}), \mathbf{\Phi}(\mathbf{v}) >_S$$

to the form:

$$\kappa_{WL}(\mathbf{u},\mathbf{v}) = < \mathbf{\Phi}(\xi(\mathbf{u})), \mathbf{\Phi}(\xi(\mathbf{v})) >_S . \tag{22}$$

This is straightforward to see, considering the input vector transformation of (15).

In order to convert the CKLMS, CKAPA algorithms (or any other complex kernel-based methods) to consider widely-linear estimation, we may simply replace the complex kernel evaluation with a form considering WL estimation.

Here we illustrate with a few examples. Considering the standard Hermitian inner product, i.e.:

$$< \mathbf{\Phi}(\mathbf{u}), \mathbf{\Phi}(\mathbf{v}) >_S = \mathbf{v}^H \mathbf{u} \tag{23}$$

The WL estimation-based form is:

$$\begin{aligned} < \mathbf{\Phi}(\xi(\mathbf{u})), \mathbf{\Phi}(\xi(\mathbf{v})) >_S &= \begin{bmatrix} \mathbf{v} \\ \mathbf{v}^* \end{bmatrix}^H \begin{bmatrix} \mathbf{u} \\ \mathbf{u}^* \end{bmatrix} \\ &= [\mathbf{v}^H \ \mathbf{v}^T] \begin{bmatrix} \mathbf{u} \\ \mathbf{u}^* \end{bmatrix} \\ &= \mathbf{v}^H\mathbf{u} + \mathbf{v}^T\mathbf{u}^* \\ &= 2\,\mathrm{Re}\{\mathbf{v}^H\mathbf{u}\} \end{aligned} \tag{24}$$

Use of equation (23) in (2) can be seen to reduce the CKLMS to an algorithm equivalent to the Complex LMS (since the kernel function used is the standard Hermitian inner product). Meanwhile, use of (24) in (18) reduces the WL-CKLMS to the Widely-Linear LMS algorithm of [10]. Although, for both cases here, these algorithms are now cast entirely in terms of inner products. A similar claim can be made considering (23) in the CKAPA, and (24) in the WL-CKAPA, i.e. they become the Complex APA, and WL form of Complex APA, respectively.

We can also similarly consider other complex kernels. For example, with the complex Gaussian kernel of [7]:

$$< \mathbf{\Phi}(\mathbf{u}), \mathbf{\Phi}(\mathbf{v}) >_S = \exp\left(-\frac{\sum_{i=1}^{d} (u_i - v_i^*)^2}{\sigma^2} \right) \tag{25}$$

The WL estimation-based form is:

$$\begin{aligned} &< \mathbf{\Phi}(\xi(\mathbf{u})), \mathbf{\Phi}(\xi(\mathbf{v})) >_S \\ &= \exp\left(-\frac{\sum_{i=1}^{d} (u_i - v_i^*)^2 + \sum_{i=1}^{d} (u_i^* - v_i)^2}{\sigma^2} \right) \\ &= \exp\left(-\frac{\sum_{i=1}^{d} 2\,\mathrm{Re}\{(u_i - v_i^*)^2\}}{\sigma^2} \right) \end{aligned} \tag{26}$$

The kernel function (26) represents the widely-linear version of the complex Gaussian kernel (25). We will also consider the polynomial kernel later in this section.

An important consideration, however, is whether the function $\kappa_{WL}(\cdot,\cdot)$ of (22) remains a kernel function when $\kappa(\cdot,\cdot)$ is a kernel. This is considered in Appendix C, where requirements for a function to be a kernel are outlined, and a proof that $\kappa_{WL}(\cdot,\cdot)$ is a kernel function is described.

To understand the benefits of the widely-linear methods described for learning with complex nonlinear channels, we consider an example using the polynomial kernel.

The polynomial kernel may be written as [5]:

$$K(\mathbf{u},\mathbf{v}) = (1 + \mathbf{v}^T\mathbf{u})^q .$$

Using the Hermitian inner product (since **u**, **v** are complex):

$$K(\mathbf{u},\mathbf{v}) = (1 + \mathbf{v}^H\mathbf{u})^q . \tag{27}$$

Note we have not yet considered use of WL estimation.

To simplify the analysis, we consider the $q = 2$ case only. The results can be seen to extend for larger values of q. Also, we will consider **u**, **v** as vectors of length $M = 2$ (again to simply the analysis). The length, M, of **u**, **v** corresponds to the input vector length used in estimating the nonlinear channel (same as $\mathbf{z}(n)$ in (2)). The effect of this length will be considered in the analysis to follow.

For $q = 2$, equation (27) can be expanded as:

$$\begin{aligned} K(\mathbf{u},\mathbf{v}) &= (1 + \mathbf{v}^H\mathbf{u})^2 \\ &= 1 + 2\mathbf{v}^H\mathbf{u} + (\mathbf{v}^H\mathbf{u})^2 \\ &= 1 + 2(v_1^*u_1 + v_2^*u_2) + (v_1^*u_1 + v_2^*u_2)^2 \\ &= 1 + 2v_1^*u_1 + 2v_2^*u_2 + v_1^{*2}u_1^2 + 2v_1^*v_2^*u_1u_2 + v_2^{*2}u_2^2 \end{aligned} \tag{28}$$

which can be expressed using the form:

$$K(\mathbf{u},\mathbf{v}) = \varphi(\mathbf{v})^H \varphi(\mathbf{u}) \tag{29}$$

where:

$$\varphi(\mathbf{u}) = [1 \ \sqrt{2}\,u_1 \ \sqrt{2}\,u_2 \ u_1^2 \ \sqrt{2}\,u_1 u_2 \ u_2^2]^T \qquad (30)$$
$$\varphi(\mathbf{v}) = [1 \ \sqrt{2}\,v_1 \ \sqrt{2}\,v_2 \ v_1^2 \ \sqrt{2}\,v_1 v_2 \ v_2^2]^T .$$

Note that any function (in this case equation (27)) that can be written equivalently with the form $\varphi(\mathbf{v})^H\varphi(\mathbf{u})$ can be shown to be a kernel function, and is referred to as a Mercer kernel [5].

Thus, the complex polynomial kernel form of (27) can be seen as defining the inner product for a nonlinear transform space (i.e. complex RKHS of a complex kernel), where $\varphi(\cdot)$ is the transform mapping function.

Considering (30), the 2-dimensional input is nonlinearly mapped to a transform space with dimension size 6. The transform space can be considered as follows. Based on Fig. 1 and equations (1), (2), we can see that function $\varphi(\cdot)$ of (30) may be viewed as the definition of function $\mathbf{\Phi}(\cdot)$ when equation (27) is the kernel being considered.

And, if we evaluate $\mathbf{\Phi}(\mathbf{z}(n))$:

$$\begin{aligned}\mathbf{\Phi}(\mathbf{z}(n)) &= \varphi(\mathbf{z}(n)) \\ &= [1 \ \sqrt{2}\,z(n) \ \sqrt{2}\,z(n-1) \\ &\qquad z^2(n) \ \sqrt{2}\,z(n)z(n-1) \ \ z^2(n-1)]^T.\end{aligned} \qquad (31)$$

Thus, the input is transformed to a space where all the various correlations of the elements of $\mathbf{z}(n)$, up to order q=2, are the dimensions of the transform space for mapping $\mathbf{\Phi}(\cdot)$ (i.e. dimensions are 1, $z(n)$, $z(n$-1), $z^2(n)$, $z(n)\cdot z(n$-1), $z^2(n$-1)). Thus linear weights, $\mathbf{w}$, applied to transformed input, $\mathbf{\Phi}(\mathbf{z}(n))$, can learn nonlinear channels up to order q=2.

However, to improve the nonlinear MS estimation performance with complex data, we consider an approach which allows the moments of estimate $\hat{d}(n)$ to be described by moments of order k of inputs $z(n)$, $z(n$-1) and their conjugates, similar to the widely-linear analysis of Section III. This clearly is not the case using (31), since the conjugate forms of $z(n)$, $z(n$-1) are not present.

Now, we consider the widely-linear form of (27):

$$\begin{aligned}K(\xi(\mathbf{u}),\xi(\mathbf{v})) \\ &= \left(1+\begin{bmatrix}\mathbf{v}\\ \mathbf{v}^*\end{bmatrix}^H\begin{bmatrix}\mathbf{u}\\ \mathbf{u}^*\end{bmatrix}\right)^q \\ &= \left(1+\mathbf{v}^H\mathbf{u}+\mathbf{v}^T\mathbf{u}^*\right)^q \\ &= \left(1+2\,\mathrm{Re}\{\mathbf{v}^H\mathbf{u}\}\right)^q\end{aligned} \qquad (32)$$

Using (32), we again consider the q=2, M=2 case:

$$\begin{aligned}K(\mathbf{u},\mathbf{v}) &= (1+\mathbf{v}^H\mathbf{u}+(\mathbf{v}^H\mathbf{u})^*)^2 \\ &= 1+2((\mathbf{v}^H\mathbf{u})+(\mathbf{v}^H\mathbf{u})^*)+((\mathbf{v}^H\mathbf{u})+(\mathbf{v}^H\mathbf{u})^*)^2 \\ &= 1+2(v_1^*u_1+v_2^*u_2+v_1u_1^*+v_2u_2^*) \\ &\qquad +(v_1^*u_1+v_2^*u_2+v_1u_1^*+v_2u_2^*)^2 \\ &= 1+2v_1^*u_1+2v_2^*u_2+2v_1u_1^*+2v_2u_2^* \\ &\qquad +v_1^{*2}u_1^2+v_1^*v_2^*u_1u_2+v_1^*v_1u_1u_1^*+v_1^*v_2u_1u_2^* \\ &\qquad +v_1^*v_2^*u_1u_2+v_2^{*2}u_2^2+v_2^*v_1u_2u_1^*+v_2^*v_2u_2u_2^* \\ &\qquad +v_1^*v_1u_1u_1^*+v_2^*v_1u_2u_1^*+v_1^2u_1^{*2}+v_1v_2u_1^*u_2^* \\ &\qquad +v_1^*v_2u_1u_2^*+v_2^*v_2u_2u_2^*+v_1v_2u_1^*u_2^*+v_2^2u_2^{*2}\end{aligned} \qquad (33)$$

Which can be written in the form:

$$K(\xi(\mathbf{u}),\xi(\mathbf{v})) = \varphi_\xi(\mathbf{v})^H\varphi_\xi(\mathbf{u}) \qquad (34)$$

where:

$$\begin{aligned}\varphi_\xi(\mathbf{v}) = \big[\,1 \ \ &\sqrt{2}\cdot v_1 \ \ \sqrt{2}\cdot v_2 \ \ \sqrt{2}\cdot v_1^* \ \ \sqrt{2}\cdot v_2^* \\ &\sqrt{2}\cdot v_1v_2 \ \ \sqrt{2}\cdot v_1v_1^* \ \ \sqrt{2}\cdot v_1v_2^* \\ &\sqrt{2}\cdot v_2v_1^* \ \ \sqrt{2}\cdot v_2v_2^* \ \ \sqrt{2}\cdot v_1^*v_2^* \\ &v_1^2 \ \ \ v_2^2 \ \ \ v_1^{*2} \ \ \ v_2^{*2}\,\big]^T \\ \varphi_\xi(\mathbf{u}) = \big[\,1 \ \ &\sqrt{2}\cdot u_1 \ \ \sqrt{2}\cdot u_2 \ \ \sqrt{2}\cdot u_1^* \ \ \sqrt{2}\cdot u_2^* \\ &\sqrt{2}\cdot u_1u_2 \ \ \sqrt{2}\cdot u_1u_1^* \ \ \sqrt{2}\cdot u_1u_2^* \\ &\sqrt{2}\cdot u_2u_1^* \ \ \sqrt{2}\cdot u_2u_2^* \ \ \sqrt{2}\cdot u_1^*u_2^* \\ &u_1^2 \ \ \ u_2^2 \ \ \ u_1^{*2} \ \ \ u_2^{*2}\,\big]^T.\end{aligned} \qquad (35)$$

Using $\varphi_\xi(\cdot)$ of equation (35), the 2-dimensional input vector is nonlinearly mapped to a transform space of dimension size 15.

Considering the transform space of $\varphi_\xi(\cdot)$ in a similar fashion to the transform space of $\varphi(\cdot)$, the dimensions of the transform space for mapping $\mathbf{\Phi}(\xi(\mathbf{z}(n)))$ can be seen to consist of:

1, $z(n)$, $z(n$-1), $z^*(n)$, $z^*(n$-1),
$z(n)\cdot z(n$-1), $z(n)\cdot z^*(n)$, $z(n)\cdot z^*(n$-1),
$z(n$-1)$\cdot z^*(n)$, $z(n$-1)$\cdot z^*(n$-1), $z^*(n)\cdot z^*(n$-1),
$z^2(n)$, $z^2(n$-1), $z^{*2}(n)$, $z^{*2}(n$-1)

Thus, the input is now transformed into a space where all the various correlations of the elements of $\mathbf{z}(n)$ along with their conjugates, up to order q=2, form the dimensions of the transform space for the mapping.

Use of kernel (33) (i.e. transform mapping (35)), should allow for improved nonlinear MS estimation performance with complex data, since the moments of the estimate can be completely described using moments of order k of the input samples, $z(n)$, $z(n$-1) and their conjugates. The result obtained can also be seen to extend for larger values of q and M.

Thus, the use of widely-linear estimation can clearly be seen to provide benefit using the polynomial kernel (27) and the standard Hermitian inner product (23). A proof of the benefits for the complex Gaussian kernel (25) is not as straightforward to determine via analysis, however, and simulations are used here to verify the benefit.

Simulation results using complex input data (both circular, noncircular) are used to verify the MSE performance benefits of the WL estimation-based forms. A comparison with the complexified form of the real Gaussian kernel, from [8] which is an approach for extending the real Gaussian kernel for complex data is also considered.

It may be observed that the use of the WL estimation form can affect convergence behavior, since the dimension size of the resulting transform space increases substantially (i.e. from 6 to 15 for the polynomial example with q=2, M=2). Thus, the choice of μ, in addition to kernel parameter σ, should be carefully considered. However, the convergence behavior of kernel-based methods, specifically the KLMS and Complex KLMS algorithms, are currently not as well-studied as their non kernel-based forms. An analysis of the convergence behavior of the algorithms described here may be considered for future work.

V. Simulation Results

Here, we compare the performance of the WL-CKLMS and WL-CKAPA with their CKLMS, CKAPA counterparts, for a nonlinear channel estimation task. The testing is performed using the complex Gaussian kernel, and circular, noncircular complex data.

The nonlinear channel model used consisted of the widely-linear filter:

$$\begin{aligned} t(n) &= (-0.9+0.8i)\cdot s(n)+(0.6-0.7i)\cdot s(n-1) \\ &+ (0.4+0.7i)\cdot s^*(n)+(-0.3-0.6i)\cdot s^*(n-1) \end{aligned} \quad (36)$$

followed by memoryless nonlinearity:

$$q(n)=t(n)+(0.1-0.15i)\cdot t^2(n)+(0.06+0.05i)\cdot t^3(n) \quad (37)$$

The approach is based on a standard model for nonlinear channel estimation tasks (from [1-3]), though modified considering widely-linear estimation. The signal, $r(n)$, at the receiver is corrupted using white Gaussian noise, i.e. $r(n) = q(n) + v(n)$. The noise, $v(n)$, is added such that the SNR is 16dB at the receiver.

The complex input data was generated using:

$$s(n)=0.7(\sqrt{1-\rho^2}X(n)+i\cdot\rho Y(n))$$

where $X(n)$, $Y(n)$ are Gaussian random variables with unity variance, and ρ is a variable controlling circularity. The value ρ=0.7071 generates circular input, while values close to either 0 or 1 generate highly noncircular input [7]. Here we use ρ=0.7071 for tests with circular input, and ρ=0.1 for tests with noncircular input.

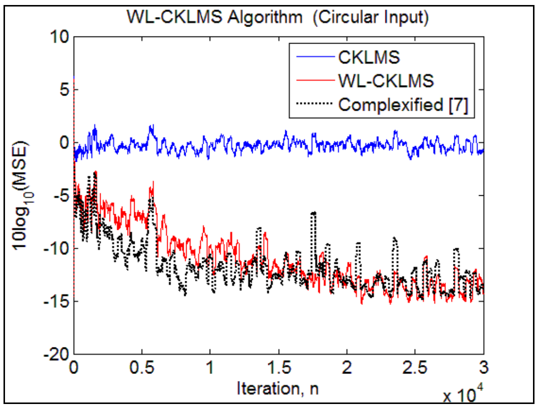


Fig. 2: WL-CKLMS, CKLMS with Circular Input

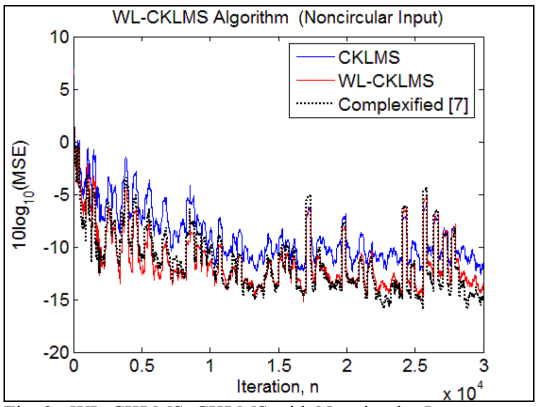


Fig. 3: WL-CKLMS, CKLMS with Noncircular Input

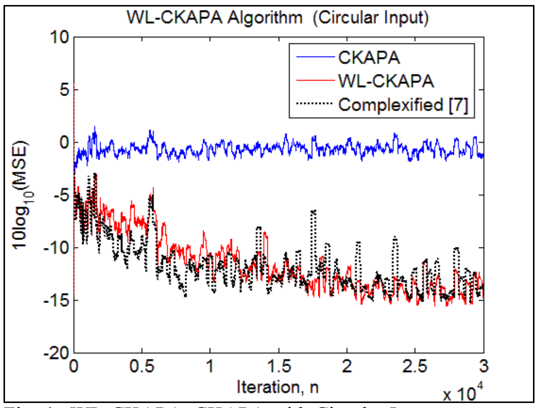


Fig. 4: WL-CKAPA, CKAPA with Circular Input

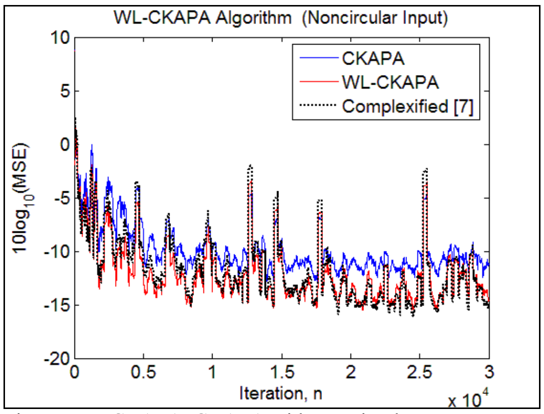


Fig. 5: WL-CKAPA, CKAPA with Noncircular Input

The following tests are shown here:

Figure 2: WL-CKLMS, CKLMS with Circular Input
Figure 3: WL-CKLMS, CKLMS with Noncircular Input
Figure 4: WL-CKAPA, CKAPA with Circular Input
Figure 5: WL-CKAPA, CKAPA with Noncircular Input

Each of the tests includes simulation results with the corresponding algorithm based on the complexified form of the real Gaussian kernel described in reference [8]. Also K=3 was used for WL-CKAPA and CKAPA.

For both Figures 2, 3, we see WL-CKLMS has improved MSE performance compared to CKLMS (a 15dB difference for circular input, 3-4dB for noncircular input). The substantial difference for the circular input case is similar to the results in [10] for the same scenario (i.e. use of circular data for a channel with a nonlinear form modeled well by WL estimation), and a detailed reason for the MSE difference is described therein. However, the MSE performance for the complexified kernel method (from [8]) is essentially similar to the WL-CKLMS in both cases.

Both methods appear near-optimal with both circular data, as well as noncircular data. Any remaining suboptimality may be due to the choice of kernel parameter, σ, for (25), which affects the nonlinear mapping. The parameter was chosen, in both cases, through extensive trial-and-error testing, and the optimal performance for both may vary.

For Figures 4, 5, we see similar test results based on the WL-CKAPA and CKAPA algorithms. The MSE performance difference between WL-CKAPA and CKAPA in steady-state again differ by 15dB for circular data and 3-4dB for noncircular data, respectively. And the MSE performance of WL-CKAPA compared to the complexified approach is similar, with suboptimalities likely due to choice of kernel parameter, σ, similar to before.

However, the convergence benefit for the APA version compared to LMS is not clearly visible from Figures 2-5. APA-based algorithms are known to provide a convergence benefit, however typically for colored input signals [15], [16].

A simulation result illustrating the convergence rate benefit for WL-CKAPA compared to WL-CKLMS can be seen in Figure 6, where a non-linear equalization test is performed (instead of channel estimation) using the same nonlinear channel used for Figures 1-4 (i.e. equations (36), (37)). The system model is illustrated in Figure 7. The benefit of faster initial convergence is demonstrated here for WL-CKAPA. However, channel equalization can often lead to enhancement of noise, affecting misadjustment performance.

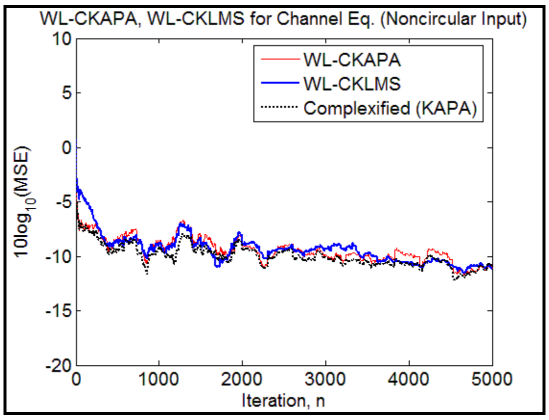


Fig. 6: WL-CKAPA,WL-CKLMS with Colored Noncircular Input

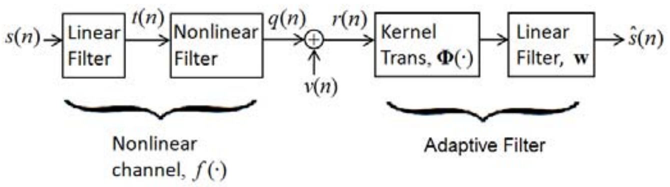


Fig. 7: System Model for Channel Equalization (used for Figure 6)

Finally, we note the complexified kernel method from [8] has similar performance to the WL estimation-based methods described here. However the work here shows an approach for improving the performance of complex kernel-based methods (described in both [7] and [8]), allowing for improved MSE performance and broadening the understanding of the complex kernel method. A detailed comparison between the complexified methods and WL estimation-based complex kernel methods are left for future work.

VI. CONCLUSIONS

In this work, we extended the complex kernel-based methods of CKLMS and CKAPA based on the widely-linear estimation approach. The approach used involved deriving widely-linear versions of complex kernel functions, and may easily be applied to other complex kernel methods, not just CKLMS, CKAPA.

Also, an analysis of the estimation benefit using the widely-linear approach (based on the polynomial kernel) was illustrated, indicating the full statistical information needed for MS estimation with complex data is available using this approach.

Simulation results also demonstrate the benefits compared to the original complex kernel methods, and similarity with the complexified method from [8]. Future work may involve a more detailed comparison of the widely-linear and complexified approaches.

APPENDIX A

Here we consider the update for coefficients, $a_j(n)$, of:

$$\mathbf{w}(n-1)=\sum_{j=1}^{n-1}a_j(n-1)\mathbf{\Phi}(\mathbf{z}(j)) \tag{A.1}$$

based on the weight update:

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\mu\mathbf{X}_{ap}(n)\left(\mathbf{X}_{ap}^{H}(n)\mathbf{X}_{ap}(n)+\varepsilon\mathbf{I}\right)^{-1}\mathbf{e}_{ap}^{*}(n)\cdot \tag{A.2}$$

Initially, we rewrite (A.2) as:

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\mu\mathbf{X}_{ap}(n)\tilde{\mathbf{e}}(n) \tag{A.3}$$

where:

$$\tilde{\mathbf{e}}(n)=\left(\mathbf{X}_{ap}^{H}(n)\mathbf{X}_{ap}(n)+\varepsilon\mathbf{I}\right)^{-1}\mathbf{e}_{ap}^{*}(n)\cdot$$

Combining (A.1) and (A.3), we get:

$$\mathbf{w}(n)=\sum_{j=1}^{n-1}a_j(n-1)\mathbf{\Phi}(\mathbf{z}(j))+\mu\mathbf{X}_{ap}(n)\tilde{\mathbf{e}}(n)\cdot \tag{A.4}$$

Which can be expanded as:

$$\mathbf{w}(n)=\sum_{j=1}^{n-1}a_j(n-1)\mathbf{\Phi}(\mathbf{z}(j))+\mu\sum_{k=1}^{K}\tilde{\mathbf{e}}_k\mathbf{\Phi}(\mathbf{z}(n-K+k))\cdot \tag{A.5}$$

where $\tilde{\mathbf{e}}_k(n)$ is the k-th element of vector $\tilde{\mathbf{e}}(n)$.

From equation (A.5), we can see that only the coefficients a_{n-k+1} to a_{n-1} are modified from iteration n-1 to n.

Thus, the update:

$$a_k(i)=\begin{cases}\mu\tilde{\mathbf{e}}_K(i), & k=i\\ a_k(i-1)+\mu\tilde{\mathbf{e}}_{K+k-i}(i), & i-K+1\le k\le i-1\\ a_k(i-1), & 1\le k<i-K+1\end{cases} \tag{A.6}$$

can be used to update the $a_j(n)$ coefficients.

APPENDIX B

Similar to Appendix A, we consider here the update for the coefficients, $a_j(n)$, of:

$$\mathbf{w}(n-1)=\sum_{j=1}^{n-1}a_j(n-1)\mathbf{\Phi}(\xi(\mathbf{z}(j))) \tag{B.1}$$

based on weight update:

$$\mathbf{w}_{WL}(n)=\mathbf{w}_{WL}(n-1)+\mu\mathbf{X}_{\xi,ap}(n)\left(\mathbf{X}_{\xi,ap}^{H}(n)\mathbf{X}_{\xi,ap}(n)+\varepsilon\mathbf{I}\right)^{-1}\mathbf{e}_{ap}^{*}(n)\,. \tag{B.2}$$

Rewriting (B.2) as:

$$\mathbf{w}_{WL}(n)=\mathbf{w}_{WL}(n-1)+\mu\mathbf{X}_{\xi,ap}(n)\tilde{\mathbf{e}}(n) \tag{B.3}$$

where:

$$\tilde{\mathbf{e}}(n)=\left(\mathbf{X}_{\xi,ap}^{H}(n)\mathbf{X}_{\xi,ap}(n)+\varepsilon\mathbf{I}\right)^{-1}\mathbf{e}_{ap}^{*}(n)\cdot$$

And combining (B.1), (B.3):

$$\mathbf{w}_{WL}(n)=\sum_{j=1}^{n-1}a_j(n-1)\mathbf{\Phi}(\xi(\mathbf{z}(j)))+\mu\sum_{k=1}^{K}\tilde{\mathbf{e}}_k\mathbf{\Phi}(\xi(\mathbf{z}(n-K+k))) \tag{B.4}$$

We obtain the same update form for updating the $a_j(n)$ coefficients as Appendix A, i.e.:

$$a_k(i)=\begin{cases}\mu\tilde{\mathbf{e}}_K(i), & k=i\\ a_k(i-1)+\mu\tilde{\mathbf{e}}_{K+k-i}(i), & i-K+1\le k\le i-1\\ a_k(i-1), & 1\le k<i-K+1\end{cases} \tag{B.5}$$

However $\tilde{\mathbf{e}}(n)$ in this case is now defined from (B.3).

APPENDIX C

Briefly, a function $\kappa(\cdot,\cdot)$ is considered a kernel function if it meets the following conditions (from [8]).

For a function: $\kappa:\boldsymbol{X}\times\boldsymbol{X}\to\mathbb{F}$, where x_1, ..., $x_N\in\boldsymbol{X}$ defines the set of possible inputs, and $\mathbb{F}$ is a general field (which may be $\mathbb{R}$ (real), or $\mathbb{C}$ (complex)), define the N x N matrix, $\mathbf{K}$, with elements $\mathbf{K}_{i,j}=\kappa(x_i,x_j)$, for i, j = 1, ..., N. The resulting matrix $\mathbf{K}$ is referred to as the Gram (or Kernel) matrix of κ with respect to x_1, ..., x_N.

If matrix $\mathbf{K}$ is Hermitian and positive definite, i.e.

$$\mathbf{c}^H\mathbf{K}\mathbf{c}=\sum_{i=1,j=1}^{N,N}c_i^*c_jK_{i,j}\ge 0$$

for all $c_i\in\mathbb{F}$, i = 1, ..., N (where * denotes conjugation), then the function $\kappa(x_i,x_j)$ is referred to as a positive definite kernel, which is often referred to simply as a kernel.

For every positive definite kernel, κ, it has been shown that there exists a single class, $\mathcal{H}$, of functions (i.e. functions f defined on the set $\boldsymbol{X}$) where $\mathcal{H}$ is a Hilbert space with a unique inner product, and where κ can reproduce the entire space of functions, $\mathcal{H}$ [8]. Due to this, the function κ is also referred to as a reproducing kernel, and $\mathcal{H}$ as its associated Reproducing Kernel Hilbert Space (or RKHS) [8]. Note, for the complex kernel case, $\mathcal{H}$ is a class of complex-valued functions, and is referred to as a complex RKHS.

Thus we have described the conditions for a function κ to be a kernel function, and its associated RKHS. However, to determine if the function $\kappa_{WL}(\cdot,\cdot)$ of (22) is also a kernel function, consider that the inputs are transformed by $\xi(\cdot)$ before applying the kernel $\kappa(\cdot,\cdot)$ in (22). This scenario can be described as follows. With the original kernel function: $K:X\times X\to\mathbb{C}$, and a function $\xi:S\to X$, we can then define the composite function: $K\circ\xi:S\times S\to\mathbb{C}$. Thus: $(K\circ\xi)(\mathbf{x}_1,\mathbf{x}_2)=K(\xi(\mathbf{x}_1),\xi(\mathbf{x}_2))$.

From [17] (see Proposition 5.13), the composite function $K\circ\xi$ is a kernel function on S for any arbitrary function, ξ. The associated Hilbert space $\mathcal{H}(K\circ\xi)$ is referred to as the pull-back of $\mathcal{H}(K)$ along ξ (Definition 5.15), and can be written as: $\mathcal{H}(K\circ\xi)=\{f\circ\xi : f\in\mathcal{H}(K)\}$ (Theorem 5.14).

Thus the function $\kappa_{WL}(\cdot,\cdot)$ of (22) may be considered a kernel function when the original function, $\kappa(\cdot,\cdot)$, is a kernel.

REFERENCES

[1] Ogunfunmi, Tokunbo, *Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches*, published by Springer Publishers, 2007.

[2] Mathews, John and Sicuranza, Giovanni, *Polynomial Signal Processing*, published by Wiley Publishers, 2000.

[3] Westwick, David and Kearney, Robert, *Identification of Nonlinear Physiological Systems*, IEEE Press, John Wiley InterScience, 2003.

[4] Theodoridis, Sergios, Slavakis, Konstantinos, and Yamada, Isao, "Adaptive learning in a world of projections: A unifying framework for linear and nonlinear classification and regression tasks," *IEEE Signal Processing Magazine*, pp.97-123, vol. 28, no. 1, Jan. 2011.

[5] Liu, Weifeng, Principe, Jose, and Haykin, Simon, *Kernel Adaptive Filtering : A Comprehensive Introduction*, published by Wiley Publishers, 2010.

[6] Ogunfunmi, Tokunbo and Paul, Thomas, "On the Complex Kernel-based Adaptive Filter", *Proceedings of the IEEE Int. Symp. on Circuits and Systems*, pp. 1263-1266, May 2011

[7] Bouboulis, Pantelis and Theodoridis, Sergios, “The Complex Gaussian Kernel LMS Algorithm”, submitted to the *Int. Conf. on Artificial Neural Networks*, 2010.

[8] Bouboulis, Pantelis and Theodoridis, Sergios, “Extension of Wirtinger’s Calculus to Reproducing Kernel Hilbert Spaces and the Complex Kernel LMS”, *IEEE Transactions on Signal Processing*, vol. 59, no. 3, March 2011.

[9] Picinbono, Bernard and Chevalier, Pascal, “Widely linear estimation with complex data,” *IEEE Trans. Signal Process.*, vol. 43, no. 8, pp. 2030–2033, 1995

[10] Adali, Tülay, Li, Hualiang, and Aloyious, Ronald, "On Properties of the Widely Linear MSE Filter and its LMS Implementation", in *Proc. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, Mar. 2009

[11] Adali, Tülay and Haykin, Simon, *Complex-Valued Adaptive Signal Processing*, ser. Adaptive Signal Processing: Next Generation Solutions, Hoboken, NJ: Wiley, 2010

[12] Liu, Weifeng, and Principe, Jose C., "Kernel Affine Projection Algorithms", *EURASIP Journal on Advances in Signal Processing*, 2008. URL http://www.hindawi.com/GetArticle.aspx?doi=10.1155 /2008/784292

[13] Richard, Cedric, Bermudez, Jose Carlos M. and Honeine, Paul, "Online Prediction of time series data with Kernels", *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058-1066, 2009.

[14] Engel, Yaakov, Mannor, Shie and Meir, Ron, "Kernel recursive least squares", *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275-2285, 2004.

[15] Diniz, Paulo S. R. "Adaptive Filtering: Algorithms and Practical Implementation", 3rd Ed., published by Springer Publishers, 2008.

[16] Haykin, Simon, "Adaptive Signal Processing", 4th ed., published by Prentice-Hall, 2002

[17] V. I. Paulsen, An Introduction to the Theory of Reproducing Kernel Hilbert Spaces. Available: http://www.math.uh.edu/vern/rkhs.pdf