Complex-Valued Kernel Methods for Regression

Rafael Boloix-Tortosa, Juan José Murillo-Fuentes, Irene Santos Velázquez, and Fernando Pérez-Cruz,

***Abstract*—Usually, complex-valued RKHS are presented as an straightforward application of the real-valued case. In this paper we prove that this procedure yields a limited solution for regression. We show that another kernel, here denoted as pseudo-kernel, is needed to learn any function in complex-valued fields. Accordingly, we derive a novel RKHS to include it, the widely RKHS (WRKHS). When the pseudo-kernel cancels, WRKHS reduces to complex-valued RKHS of previous approaches. We address the kernel and pseudo-kernel design, paying attention to the kernel and the pseudo-kernel being complex-valued. In the experiments included we report remarkable improvements in simple scenarios where real a imaginary parts have different similitude relations for given inputs or cases where real and imaginary parts are correlated. In the context of these novel results we revisit the problem of non-linear channel equalization, to show that the WRKHS helps to design more efficient solutions.**

***Index Terms*—Complex-valued RKHS, kernel methods, regression, non-linear channel equalization.**

I. Introduction

COMPLEX-VALUED signal processing is of fundamental interest. Its main benefit is the availability of processing the real and imaginary parts as a single signal. It finds application in a vast range of nowadays systems in science and engineering such as telecommunications, optics, electromagnetics, and acoustics among others. Signal processing for complex-valued signals has been widely studied in the linear case, see [1] and references therein. The non-linear processing of complex-valued signals has been addressed from the point of view of neural networks, [2] and, recently, using reproducing kernel Hilbert spaces (RKHS). Some complex kernel-based algorithms have been lately proposed for classification [3], regression [4], [5], [6] and mainly for kernel principal component analysis [7]. Regarding regression, in [5] the authors propose a complex-valued kernel based in the results in [3] and face the derivative of cost functions by using Wirtinger's derivatives. Same kernel is adopted in [4]. And in [8] the augmented version of the algorithm is proposed. In [6] the authors review the kernel design to improve the previous solutions. These previous approaches have been developed in the framework of kernel least mean square (KLMS).

Except for the method in [8] all these algorithms are straight-forward applications of real-valued RKHS. In [8] some additional considerations are developed for time adaptive estimation within the definition of the inner product in the feature space. These formulations, that are useful in the learning of many problems, are limited for learning in others. As we show in this paper they cannot learn any given complex-valued non-linear function.

In this paper we propose a novel RKHS for complex-valued fields with full representation capabilities. We show that to represent any complex-valued function we need to include an additional term, denoted as pseudo-kernel[1]. We refer to this new approach as widely RKHS (WRKHS) after *widely* linear complex valued solutions in linear systems [1]. The results in [4], [5] and [6] can be seen as a particular case of WRKHS in which the pseudo-kernel is considered zero. We denote these approaches as strictly complex-valued RKHS (SRKHS). The need for the WRKHS can be justified in cases where the real and imaginary parts are correlated and learning them independently is, at best, suboptimal. Besides there are some relations that cannot even be capture with SRKHS approach, while our WRKHS, relaying on the pseudo-kernel, is able to learn on those scenarios, as we illustrate in the experimental section.

One of the key issues with our WRKHS is the need to define kernels and pseudo-kernels. In this paper we describe valid kernels and pseudo-kernels. We also detail in which cases the WRKHS can be simplified to a SRKHS with complex or real-valued kernel.

Two experiments are included to illustrate the capabilities of WRKHS. First, we face a regression where clearly a different kernel for the real and imaginary parts benefits the learning. Then we learn a function using WRKHS with a real-valued kernel and a pure imaginary complex-valued pseudo-kernel. This solution allows modeling a dependence between real and imaginary part. Here, a WRKHS clearly improves the regression. We revisit the problem of non-channel equalization to conclude, from the results in this paper, that the best option is a SRKHS with real-valued kernel, even in the non-circular case. To compare to previous approaches we develop a version for the recursive case with sample selection [9] and compare it to the results in [5], [6] for non-linear channel equalization. The paper is organized as follows. In the next section we review some concepts needed on RKHS. We continue in Section III with the derivation of WRKHS. In Section IV the SRKHS is developed as a special case of WRKHS. Section V is devoted to the analysis of the kernel and pseudo-kernel and some new proposals. The performance of these approaches is illustrated in Section VI, where several scenarios are presented and the application of WRKHS is discussed. We end with Conclusions.

The notation used in the paper is as follows. For matrix $\mathbf{A}$, $[\mathbf{A}]_{l,q}$ is its (l, q) entry, $\mathbf{A}^\mathsf{T}$ is the transpose of $\mathbf{A}$, $\mathbf{A}^\mathsf{H}$ the

R. Boloix-Tortosa, Irene Santos-Velázquez, and Juan José Murillo-Fuentes are with the Department of Signal Theory and Communications, University of Seville, Spain. e-mail: rboloix@us.es. Fernando Pérez-Cruz is with Department of Signal Theory and Communications, University Carlos III de Madrid, Spain

Thanks to Spanish government (Ministerio de Educación y Ciencia, TEC2016-78434-C03-02) and European Union (FEDER) for funding.

[1]The pseudo-kernel plays a similar role of the pseudo-covariance of complex-valued random variables.

Hermitian transpose, $\mathbf{A}^*$ its complex conjugate and $\mathbf{A}^{-*}$ its inverse conjugate. $\mathbf{I}_n$ denotes the identity matrix of size n. For a vector $\mathbf{a}$, $[\mathbf{a}]_l$ denotes its l-th entry. To denote the i-th sample of a vector and signal we use, respectively, $\mathbf{a}(i)$ and $a(i)$. The real and imaginary parts are denoted by subindex r and j, respectively, i.e. $\mathbf{a} = \mathbf{a}_r + j\mathbf{a}_j$, with $j = \sqrt{-1}$. To denote the complex Gaussian distribution with mean vector $\boldsymbol{\mu}$, covariance matrix $\mathbf{K}$ and pseudo-covariance matrix $\tilde{\mathbf{K}}$ we use $\mathcal{N}(\boldsymbol{\mu}, \mathbf{K}, \tilde{\mathbf{K}})$. We write the inner or dot product as $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{b}^H \mathbf{a} = \mathbf{a}^T \mathbf{b}^*$.

II. RKHS

A RKHS is a Hilbert space of functions that can be defined by a reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ [10]. Given the reproducing kernel k, the RKHS $\mathcal{H}_k$ of real-valued functions on the set $\mathcal{X}$ is the Hilbert space containing $k(\mathbf{x}, \cdot)$ for every $\mathbf{x} \in \mathcal{X}$ and where k has the *reproducing property*

$$f(\mathbf{x}_\star) = \langle f, k(\mathbf{x}_\star, \cdot) \rangle_k \quad \forall f \in \mathcal{H}_k, \tag{1}$$

being $\langle \cdot, \cdot \rangle_k$ the inner product in $\mathcal{H}_k$. In particular, $\langle k(\mathbf{x}, \cdot), k(\mathbf{x}_\star, \cdot) \rangle_k = k(\mathbf{x}, \mathbf{x}_\star)$. In a RKHS, functions are in the closure of the linear combinations of the kernel at given points:

$$f(\mathbf{x}_\star) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_\star, \mathbf{x}(i)) = \mathbf{k}(\mathbf{x}_\star, \mathbf{X})\boldsymbol{\alpha}, \tag{2}$$

where f is in the class $\mathcal{F}$ of real functions forming a real Hilbert space, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$, and $\mathbf{k}(\mathbf{x}_\star, \mathbf{X}) = [k(\mathbf{x}_\star, \mathbf{x}(1)), k(\mathbf{x}_\star, \mathbf{x}(2)), \ldots, k(\mathbf{x}_\star, \mathbf{x}(n))]$.

In the complex-valued case, one might work with the real and imaginary parts stacked into a so denoted *composite* vector form. The definition of RKHS for vector valued functions parallels the one in the scalar, with the main difference that the reproducing kernel is now matrix valued [11],

$$\mathbf{f}_R(\mathbf{x}_\star) = \begin{bmatrix} f_r(\mathbf{x}_\star) \\ f_j(\mathbf{x}_\star) \end{bmatrix} = \begin{bmatrix} \mathbf{k}_{rr}(\mathbf{x}_\star, \mathbf{X}) & \mathbf{k}_{rj}(\mathbf{x}_\star, \mathbf{X}) \\ \mathbf{k}_{jr}(\mathbf{x}_\star, \mathbf{X}) & \mathbf{k}_{jj}(\mathbf{x}_\star, \mathbf{X}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_j \end{bmatrix} \tag{3}$$

that can be rewritten in compact form as $\mathbf{f}_R(\mathbf{x}_\star) = \mathbf{K}_R(\mathbf{x}_\star, \mathbf{X})\boldsymbol{\alpha}_R$. We have two-dimensional vector, and we can define an estimator by minimizing the regularized empirical error on the basis of a training set $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\} = \{(\mathbf{x}(1), y(1)), \ldots, (\mathbf{x}(n), y(n))\}$:

$$\frac{1}{n}\sum_{i=1}^{n} (f_r(\mathbf{x}(i)) - y_r(i))^2 + \frac{1}{n}\sum_{i=1}^{n} (f_j(\mathbf{x}(i)) - y_j(i))^2 + \frac{\lambda}{n}\|\mathbf{f}_R\|^2_{\mathbf{K}} \tag{4}$$

the coefficients yield

III. Widely Complex RKHS

Based on the *widely linear* concept [1] we propose the following RKHS for regression in complex-valued formulation.

Definition 3.1: Widely complex RKHS. We denote as widely complex-valued RKHS (WRKHS) the RKHS defined by the kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ and a *pseudo-kernel* $\tilde{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$,

$$f(\mathbf{x}_\star) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_\star, \mathbf{x}(i)) + \sum_{i=1}^{n} \alpha_i^* \tilde{k}(\mathbf{x}_\star, \mathbf{x}(i)) \tag{6}$$

where $\alpha_i \in \mathbb{C}$.

The *pseudo-kernel* is related to the feature map, $\boldsymbol{\varphi} : \mathcal{X} \to \mathbb{C}^q$, by $\tilde{k}(\mathbf{x}_\star, \mathbf{x}) = \langle \boldsymbol{\varphi}(\mathbf{x}_\star), \boldsymbol{\varphi}(\mathbf{x})^* \rangle$. We introduce the following definitions that we need in the next proposition.

Definition 3.2: Kernels of real-imaginary parts of the feature space. We define the kernels for the real to real, real to imaginary, imaginary to real and imaginary to imaginary parts of the feature space, respectively, as

$$\begin{aligned} \gamma_{rr}(\mathbf{x}_\star, \mathbf{x}) &= \langle \boldsymbol{\varphi}_r(\mathbf{x}_\star), \boldsymbol{\varphi}_r(\mathbf{x}) \rangle, \\ \gamma_{rj}(\mathbf{x}_\star, \mathbf{x}) &= \langle \boldsymbol{\varphi}_r(\mathbf{x}_\star), \boldsymbol{\varphi}_j(\mathbf{x}) \rangle, \\ \gamma_{jr}(\mathbf{x}_\star, \mathbf{x}) &= \langle \boldsymbol{\varphi}_j(\mathbf{x}_\star), \boldsymbol{\varphi}_r(\mathbf{x}) \rangle, \\ \gamma_{jj}(\mathbf{x}_\star, \mathbf{x}) &= \langle \boldsymbol{\varphi}_j(\mathbf{x}_\star), \boldsymbol{\varphi}_j(\mathbf{x}) \rangle, \end{aligned} \tag{7}$$

where $\gamma_{rj}(\mathbf{x}_\star, \mathbf{x}) = \gamma_{jr}(\mathbf{x}, \mathbf{x}_\star)$.

Proposition 3.1: WRKHS reproducing properties. The WRKHS can learn the real and the imaginary parts of the output as in (3).

Proof. The output (6) as a function of the feature space can be rewritten as

$$f(\mathbf{x}_\star) = \sum_{i=1}^{n} \alpha_i \langle \boldsymbol{\varphi}(\mathbf{x}_\star), \boldsymbol{\varphi}(\mathbf{x}(i)) \rangle + \sum_{i=1}^{n} \alpha_i^* \langle \boldsymbol{\varphi}(\mathbf{x}_\star), \boldsymbol{\varphi}(\mathbf{x}(i))^* \rangle .$$

In composite form it follows that

$$\begin{aligned} \begin{bmatrix} f_r(\mathbf{x}_\star) \\ f_j(\mathbf{x}_\star) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\varphi}_{r\star}^T & -\boldsymbol{\varphi}_{j\star}^T \\ \boldsymbol{\varphi}_{j\star}^T & \boldsymbol{\varphi}_{r\star}^T \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{\Phi}_r & \boldsymbol{\Phi}_j \\ -\boldsymbol{\Phi}_j & \boldsymbol{\Phi}_r \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi}_r & -\boldsymbol{\Phi}_j \\ \boldsymbol{\Phi}_j & \boldsymbol{\Phi}_r \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_j \end{bmatrix} \\ &= 2 \begin{bmatrix} \boldsymbol{\varphi}_{r\star}^T \boldsymbol{\Phi}_r & \boldsymbol{\varphi}_{r\star}^T \boldsymbol{\Phi}_j \\ \boldsymbol{\varphi}_{j\star}^T \boldsymbol{\Phi}_r & \boldsymbol{\varphi}_{j\star}^T \boldsymbol{\Phi}_j \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_j \end{bmatrix} \\ &= 2 \begin{bmatrix} \boldsymbol{\gamma}_{rr}(\mathbf{x}_\star, \mathbf{X}) & \boldsymbol{\gamma}_{rj}(\mathbf{x}_\star, \mathbf{X}) \\ \boldsymbol{\gamma}_{jr}(\mathbf{x}_\star, \mathbf{X}) & \boldsymbol{\gamma}_{jj}(\mathbf{x}_\star, \mathbf{X}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_j \end{bmatrix}. \end{aligned} \tag{8}$$

This corresponds to the approach in (3) where the entries of the matrix in $\mathbf{K}_R(\mathbf{x}_\star, \mathbf{X})$ can be easily identified. □

In the proof of Proposition 3.1 we added both terms in (6). An interesting conclusion can be drawn if we develop each term independently,

$$\begin{bmatrix} f_r(\mathbf{x}_\star) \\ f_j(\mathbf{x}_\star) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}_{r\star}^T \boldsymbol{\Phi}_r + \boldsymbol{\varphi}_{j\star}^T \boldsymbol{\Phi}_j & \boldsymbol{\varphi}_{r\star}^T \boldsymbol{\Phi}_j - \boldsymbol{\varphi}_{j\star}^T \boldsymbol{\Phi}_r \\ -\boldsymbol{\varphi}_{r\star}^T \boldsymbol{\Phi}_j + \boldsymbol{\varphi}_{j\star}^T \boldsymbol{\Phi}_r & \boldsymbol{\varphi}_{r\star}^T \boldsymbol{\Phi}_r + \boldsymbol{\varphi}_{j\star}^T \boldsymbol{\Phi}_j \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_j \end{bmatrix}$$

$$+ \begin{bmatrix} \boldsymbol{\varphi}_{r\star}^T \boldsymbol{\Phi}_r - \boldsymbol{\varphi}_{j\star}^T \boldsymbol{\Phi}_j & -\boldsymbol{\varphi}_{r\star}^T \boldsymbol{\Phi}_j - \boldsymbol{\varphi}_{j\star}^T \boldsymbol{\Phi}_r \\ \cdots \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_r \\ \cdots \end{bmatrix} = \boldsymbol{\alpha}_R \quad ($$

$$\mathbf{K}_{\mathrm{R}}(\mathbf{X},\mathbf{X}) + \lambda\mathbf{I}_{2n})^{-1}\,\mathbf{y}_{\mathrm{R}}, \quad (5)$$

where $\mathbf{y}_{\mathrm{R}} = [\mathbf{y}_{\mathrm{r}}^{\mathsf{T}}\mathbf{y}_{\mathrm{j}}^{\mathsf{T}}]^{\mathsf{T}}$, with $\mathbf{y}_{\mathrm{r}} = [y_{\mathrm{r}}(1), \ldots, y_{\mathrm{r}}(n)]^{\mathsf{T}}$ and $\mathbf{y}_{\mathrm{j}} = [y_{\mathrm{j}}(1), \ldots, y_{\mathrm{j}}(n)]^{\mathsf{T}}$.

$$+\boldsymbol{\varphi}_{\mathrm{r}}.\quad \boldsymbol{\Phi}_{\mathrm{j}} + \boldsymbol{\varphi}_{\mathrm{j}}.\quad \boldsymbol{\Phi}_{\mathrm{r}} \quad -\boldsymbol{\varphi}_{\mathrm{r}}.\quad \boldsymbol{\Phi}_{\mathrm{r}} + \boldsymbol{\varphi}_{\mathrm{j}}.\quad \boldsymbol{\Phi}_{\mathrm{j}} \quad -\boldsymbol{\alpha}_{\mathrm{j}} \quad (9)$$

The two matrices above resemble the covariance and the pseudo-covariance in complex-valued Gaussian distributions, respectively. The pseudo-covariance cancels for conditions similar to those of the proper case in complex-valued random variables: if $\boldsymbol{\varphi}_{r\cdot}^{T}\boldsymbol{\varphi}_{r} = \boldsymbol{\varphi}_{j\cdot}^{T}\boldsymbol{\varphi}_{j}$ and $\boldsymbol{\varphi}_{r\cdot}^{T}\boldsymbol{\varphi}_{j} = -\boldsymbol{\varphi}_{j\cdot}^{T}\boldsymbol{\varphi}_{r}$. From the result in (9), the WRKHS can be rewritten in complex-valued form as follows

$$f(\mathbf{x}_\cdot) = \mathbf{k}(\mathbf{x}_\cdot,\mathbf{X})\boldsymbol{\alpha} + \tilde{\mathbf{k}}(\mathbf{x}_\cdot,\mathbf{X})\boldsymbol{\alpha}^*, \tag{10}$$

and we next derive the value for $\boldsymbol{\alpha}$ by minimizing the regularized empirical error. We will make use of the *augmented* vector $\underline{\mathbf{f}}(\mathbf{x}_\cdot) = [f(\mathbf{x}_\cdot)\; f^*(\mathbf{x}_\cdot)]^{\mathsf{T}}$:

$$\underline{\mathbf{f}}(\mathbf{x}_\cdot) = \begin{bmatrix} \mathbf{k}(\mathbf{x}_\cdot,\mathbf{X}) & \tilde{\mathbf{k}}(\mathbf{x}_\cdot,\mathbf{X}) \\ \tilde{\mathbf{k}}^*(\mathbf{x}_\cdot,\mathbf{X}) & \mathbf{k}^*(\mathbf{x}_\cdot,\mathbf{X}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^* \end{bmatrix} = \underline{\mathbf{K}}(\mathbf{x}_\cdot,\mathbf{X})\underline{\boldsymbol{\alpha}}, \tag{11}$$

where matrix $\underline{\mathbf{K}}(\mathbf{x}_\cdot,\mathbf{X})$ is the *augmented kernel matrix*, and $\underline{\boldsymbol{\alpha}} = [\boldsymbol{\alpha}^{\mathsf{T}}\; \boldsymbol{\alpha}^{\mathsf{H}}]^{\mathsf{T}}$ is also an augmented vector. There exists a simple relation between the composite and the *augmented* vector, $\underline{\mathbf{f}}(\mathbf{x}_\cdot) = \mathbf{T}\mathbf{f}_{\mathbb{R}}(\mathbf{x}_\cdot)$, where

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_n & \mathrm{j}\mathbf{I}_n \\ \mathbf{I}_n & -\mathrm{j}\mathbf{I}_n \end{bmatrix} \in \mathbb{C}^{2n\times 2n}, \tag{12}$$

and $\mathbf{T}\mathbf{T}^{\mathsf{H}} = \mathbf{T}^{\mathsf{H}}\mathbf{T} = 2\mathbf{I}_{2n}$. This simple transformation allows us to calculate the augmented vector (11) from the real and imaginary parts (3):

$$\begin{aligned} \underline{\mathbf{f}}(\mathbf{x}_\cdot) &= \mathbf{T}\mathbf{K}_{\mathbb{R}}(\mathbf{x}_\cdot,\mathbf{X})\boldsymbol{\alpha}_{\mathbb{R}} = \mathbf{T}\mathbf{K}_{\mathbb{R}}(\mathbf{x}_\cdot,\mathbf{X})\left(\frac{1}{2}\mathbf{T}^{\mathsf{H}}\mathbf{T}\right)\boldsymbol{\alpha}_{\mathbb{R}} \\ &= \frac{1}{2}\mathbf{T}\mathbf{K}_{\mathbb{R}}(\mathbf{x}_\cdot,\mathbf{X})\mathbf{T}^{\mathsf{H}}\underline{\boldsymbol{\alpha}}. \end{aligned} \tag{13}$$

Hence, the augmented kernel matrix is related to $\mathbf{K}_{\mathbb{R}}(\mathbf{x}_\cdot,\mathbf{X})$ as

$$\underline{\mathbf{K}}(\mathbf{x}_\cdot,\mathbf{X}) = \frac{1}{2}\mathbf{T}\mathbf{K}_{\mathbb{R}}(\mathbf{x}_\cdot,\mathbf{X})\mathbf{T}^{\mathsf{H}}. \tag{14}$$

And the augmented vector $\underline{\boldsymbol{\alpha}}$ can be found from (5) as

$$\begin{aligned} \underline{\boldsymbol{\alpha}} &= \mathbf{T}\boldsymbol{\alpha}_{\mathbb{R}} = \mathbf{T}\left(\mathbf{K}_{\mathbb{R}}(\mathbf{X},\mathbf{X}) + \lambda\mathbf{I}_{2n}\right)^{-1}\mathbf{y}_{\mathbb{R}} \\ &= \left(\left(\mathbf{K}_{\mathbb{R}}(\mathbf{X},\mathbf{X}) + \lambda\mathbf{I}_{2n}\right)\mathbf{T}^{-1}\right)^{-1}\frac{1}{2}\mathbf{T}^{\mathsf{H}}\mathbf{T}\mathbf{y}_{\mathbb{R}} \\ &= \frac{1}{2}\left((\mathbf{T}^{\mathsf{H}})^{-1}\left(\mathbf{K}_{\mathbb{R}}(\mathbf{X},\mathbf{X}) + \lambda\mathbf{I}_{2n}\right)\mathbf{T}^{-1}\right)^{-1}\underline{\mathbf{y}} \\ &= \left(\frac{1}{2}\,\frac{1}{2}\mathbf{T}\left(\mathbf{K}_{\mathbb{R}}(\mathbf{X},\mathbf{X}) + \lambda\mathbf{I}_{2n}\right)\frac{1}{2}\mathbf{T}^{\mathsf{H}}\right)^{-1}\underline{\mathbf{y}} \\ &= \left(\underline{\mathbf{K}}(\mathbf{X},\mathbf{X}) + \lambda\mathbf{I}_{2n}\right)^{-1}\underline{\mathbf{y}}. \end{aligned} \tag{15}$$

Note that in the general complex case, two functions $k(\mathbf{x}_\cdot,\mathbf{x})$ and $\tilde{k}(\mathbf{x}_\cdot,\mathbf{x})$ must be defined. By identifying $\mathbf{K}_{\mathbb{R}}(\mathbf{x}_\cdot,\mathbf{X})$ in (8) and substituting it in (14),

$$\underline{\mathbf{K}}(\mathbf{x}_\cdot,\mathbf{X}) = \frac{1}{2}\mathbf{T}\left(2\begin{bmatrix} \gamma_{rr}(\mathbf{x}_\cdot,\mathbf{X}) & \gamma_{rj}(\mathbf{x}_\cdot,\mathbf{X}) \\ \gamma_{jr}(\mathbf{x}_\cdot,\mathbf{X}) & \gamma_{ii}(\mathbf{x}_\cdot,\mathbf{X}) \end{bmatrix}\right)\mathbf{T}^{\mathsf{H}}. \tag{16}$$

the kernel and pseudo-kernel can be identified:

$$\begin{aligned} k(\mathbf{x}_\cdot,\mathbf{x}(i)) &= \gamma_{rr}(\mathbf{x}_\cdot,\mathbf{x}(i)) + \gamma_{jj}(\mathbf{x}_\cdot,\mathbf{x}(i)) \\ &\quad + \mathrm{j}\left(\gamma_{jr}(\mathbf{x}_\cdot,\mathbf{x}(i)) - \gamma_{rj}(\mathbf{x}_\cdot,\mathbf{x}(i))\right), \end{aligned} \tag{17}$$

$$\begin{aligned} \tilde{k}(\mathbf{x}_\cdot,\mathbf{x}(i)) &= \gamma_{rr}(\mathbf{x}_\cdot,\mathbf{x}(i)) - \gamma_{jj}(\mathbf{x}_\cdot,\mathbf{x}(i)) \\ &\quad + \mathrm{j}\left(\gamma_{jr}(\mathbf{x}_\cdot,\mathbf{x}(i)) + \gamma_{rj}(\mathbf{x}_\cdot,\mathbf{x}(i))\right). \end{aligned} \tag{18}$$

Finally, by applying the matrix-inversion lemma to (15),

$$\left(\underline{\mathbf{K}}(\mathbf{X},\mathbf{X}) + \lambda\mathbf{I}_{2n}\right)^{-1} = \begin{bmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) + \lambda\mathbf{I}_n & \tilde{\mathbf{K}}(\mathbf{X},\mathbf{X}) \\ \tilde{\mathbf{K}}^*(\mathbf{X},\mathbf{X}) & \mathbf{K}^*(\mathbf{X},\mathbf{X}) + \lambda\mathbf{I}_n \end{bmatrix}^{-1} \tag{19}$$

and substitution in (11), it yields the prediction

$$\begin{aligned} f(\mathbf{x}_\cdot) &= \mathbf{k}(\mathbf{x}_\cdot,\mathbf{X})\left[\mathbf{P}^{-1}\mathbf{y} - \mathbf{C}^{-1}\tilde{\mathbf{K}}(\mathbf{X},\mathbf{X})\mathbf{P}^{-*}\mathbf{y}^*\right] \\ &\quad + \tilde{\mathbf{k}}(\mathbf{x}_\cdot,\mathbf{X})\left[\mathbf{P}^{-*}\mathbf{y}^* - \mathbf{C}^{-*}\tilde{\mathbf{K}}^*(\mathbf{X},\mathbf{X})\mathbf{P}^{-1}\mathbf{y}\right] \end{aligned} \tag{20}$$

where $\mathbf{C} = (\mathbf{K}(\mathbf{X},\mathbf{X}) + \lambda\mathbf{I}_n)$, and $\mathbf{P} = \mathbf{C} - \tilde{\mathbf{K}}(\mathbf{X},\mathbf{X})\mathbf{C}^{-*}\tilde{\mathbf{K}}^*(\mathbf{X},\mathbf{X})$. Now the pair $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^*$ are easily identifiable.

IV. Strictly Complex RKHS

By removing the last term in (6) we have a particular case of the WRKHS that we denote as strictly complex-valued RKHS.

Definition 4.1: Strictly complex-valued RKHS. We denote as strictly complex-valued RKHS (SRKHS) the RKHS defined by the kernel $k : \mathbb{F}\times\mathbb{F} \to \mathbb{C}$,

$$f(\mathbf{x}_\cdot) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_\cdot,\mathbf{x}(i)) = \sum_{i=1}^{n} \alpha_i \langle \boldsymbol{\varphi}(\mathbf{x}_\cdot), \boldsymbol{\varphi}(\mathbf{x}(i))\rangle \tag{21}$$

where $\alpha_i \in \mathbb{C}$ and the feature map is given by $\boldsymbol{\varphi} : \mathbb{F} \to \mathbb{C}^q$.

It can be proved to be a RKHS by using complex-valued Hilbert spaces, see [12]. This RKHS, that it is a straightforward application of the real-valued RKHS, is limited compared to the WRKHS as we show next.

Proposition 4.1: SRKHS is limited as RKHS. The SRKHS is limited to represent any given complex-valued function. In particular, it yields a subset of the functions that WRKHS can represent.

Proof. By rewriting the output (21) in composite form, i.e. real and imaginary parts stacked in vector form,

$$\begin{bmatrix} f_r(\mathbf{x}_\cdot) \\ f_j(\mathbf{x}_\cdot) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}_{r\cdot}^{T}\boldsymbol{\Phi}_r + \boldsymbol{\varphi}_{j\cdot}^{T}\boldsymbol{\Phi}_j & \boldsymbol{\varphi}_{r\cdot}^{T}\boldsymbol{\Phi}_j - \boldsymbol{\varphi}_{j\cdot}^{T}\boldsymbol{\Phi}_r \\ -\boldsymbol{\varphi}_{r\cdot}^{T}\boldsymbol{\Phi}_j + \boldsymbol{\varphi}_{i\cdot}^{T}\boldsymbol{\Phi}_r & \boldsymbol{\varphi}_{r\cdot}^{T}\boldsymbol{\Phi}_r + \boldsymbol{\varphi}_{i\cdot}^{T}\boldsymbol{\Phi}_j \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_j \end{bmatrix} \tag{22}$$

where $\boldsymbol{\varphi}_{r\cdot} = \boldsymbol{\varphi}_r(\mathbf{x}_\cdot)$, $\boldsymbol{\varphi}_{j\cdot} = \boldsymbol{\varphi}_j(\mathbf{x}_\cdot)$, $\boldsymbol{\Phi}_r = \boldsymbol{\Phi}_r(\mathbf{X})$, $\boldsymbol{\Phi}_j = \boldsymbol{\Phi}_j(\mathbf{X})$ and $\boldsymbol{\Phi}(\mathbf{X})$ is an $m \times n$ matrix whose i-th column is $\boldsymbol{\varphi}(\mathbf{x}(i))$. It can be observed that the diagonal blocks of the matrix above have the same value while the off-diagonal ones have opposite sign. Hence, it cannot provide the same solutions than the WKRHS in (6) where in the general case $\mathbf{k}_{rr}(\mathbf{x}_\cdot,\mathbf{X}) \neq \mathbf{k}_{jj}(\mathbf{x}_\cdot,\mathbf{X})$ and $\mathbf{k}_{rj}(\mathbf{x}_\cdot,\mathbf{X}) \neq -\mathbf{k}_{jr}(\mathbf{x}_\cdot,\mathbf{X})$. □

The previous proposition is a consequence of the fact that linear operations applied to a real vector formed with the real and imaginary parts are not generally translated into linear operations applied to its complex counterpart.

A. Kernel structure

We next study the structure of the kernel for the SRKHS.

Proposition 4.2: Kernel in SRKHS. The solution in (3) with $k_{rr}(\mathbf{x}_\cdot,\mathbf{x}(i)) = k_{jj}(\mathbf{x}_\cdot,\mathbf{x}(i)) = k_{rr}(\mathbf{x}(i),\mathbf{x}_\cdot)$ and $k_{rj}(\mathbf{x}_\cdot,\mathbf{x}(i)) = -k_{rj}(\mathbf{x}(i),\mathbf{x}_\cdot)$ yields the SRKHS in (21) with kernel

$$k(\mathbf{x}_\cdot,\mathbf{x}(i)) = k_{rr}(\mathbf{x}_\cdot,\mathbf{x}(i)) - \mathrm{j}k_{rj}(\mathbf{x}_\cdot,\mathbf{x}(i)). \tag{23}$$

Proof. First, we rewrite (21) in vector form as

$$f(\mathbf{x}_\cdot) = \boldsymbol{\varphi}(\mathbf{x}_\cdot)^{\mathsf{T}}\Phi(\mathbf{X})^*\boldsymbol{\alpha} = \mathbf{k}(\mathbf{x}_\cdot, \mathbf{X})\boldsymbol{\alpha} \tag{24}$$

and decompose it into real and imaginary parts as in (22). Define

$$\begin{aligned} \mathbf{k}_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{X}) &= \boldsymbol{\gamma}_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{X}) + \boldsymbol{\gamma}_{\mathrm{jj}}(\mathbf{x}_\cdot, \mathbf{X}), \\ \mathbf{k}_{\mathrm{rj}}(\mathbf{x}_\cdot, \mathbf{X}) &= \boldsymbol{\gamma}_{\mathrm{rj}}(\mathbf{x}_\cdot, \mathbf{X}) - \boldsymbol{\gamma}_{\mathrm{jr}}(\mathbf{x}_\cdot, \mathbf{X}), \end{aligned} \tag{25}$$

where $\boldsymbol{\gamma}_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{X}) = [\gamma_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{x}(1)), \cdots, \gamma_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{x}(n))]$, and we have analogous definitions for $\boldsymbol{\gamma}_{\mathrm{jj}}(\mathbf{x}_\cdot, \mathbf{X})$, $\boldsymbol{\gamma}_{\mathrm{rj}}(\mathbf{x}_\cdot, \mathbf{X})$ and $\boldsymbol{\gamma}_{\mathrm{jr}}(\mathbf{x}_\cdot, \mathbf{X})$. The terms in (25) can be identified in (22) as

$$\begin{pmatrix} f_{\mathrm{r}}(\mathbf{x}_\cdot) \\ f_{\mathrm{j}}(\mathbf{x}_\cdot) \end{pmatrix} = \begin{pmatrix} \mathbf{k}_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{X}) & \mathbf{k}_{\mathrm{rj}}(\mathbf{x}_\cdot, \mathbf{X}) \\ -\mathbf{k}_{\mathrm{rj}}(\mathbf{x}_\cdot, \mathbf{X}) & \mathbf{k}_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{X}) \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_{\mathrm{r}} \\ \boldsymbol{\alpha}_{\mathrm{j}} \end{pmatrix} \tag{26}$$

where it can be concluded that $k_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{x}(i)) = k_{\mathrm{jj}}(\mathbf{x}_\cdot, \mathbf{x}(i))$.

Going back to (21) it follows that $k(\mathbf{x}_\cdot, \mathbf{x}(i)) = k_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{x}(i)) - \mathrm{j}k_{\mathrm{rj}}(\mathbf{x}_\cdot, \mathbf{x}(i))$. Finally, from the definitions of $k_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{x}(i))$ and $\mathbf{k}_{\mathrm{rj}}(\mathbf{x}_\cdot, \mathbf{X})$ in (25) and definitions in (7), it is easy to check the symmetries $k_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{x}(i)) = k_{\mathrm{rr}}(\mathbf{x}(i), \mathbf{x}_\cdot)$ and $k_{\mathrm{rj}}(\mathbf{x}_\cdot, \mathbf{x}(i)) = -k_{\mathrm{rj}}(\mathbf{x}(i), \mathbf{x}_\cdot)$.

□

Note first that by minimizing the regularized empirical error $\boldsymbol{\alpha}$ in (24) we have

$$\boldsymbol{\alpha} = (\mathbf{K}(\mathbf{X}, \mathbf{X}) + \lambda\mathbf{I}_n)^{-1}\mathbf{y}, \tag{27}$$

where $[\mathbf{K}(\mathbf{X}, \mathbf{X})]_{r,s} = k(\mathbf{x}(r), \mathbf{x}(s))$. Also, it is important to remark that a SRKHS is a particular case of the solution in (3), even if the kernel is complex-valued. By analogy with covariances of complex-valued random variables, this formulation resembles the *proper* case. In the proper case real and imaginary parts exhibit the same covariance while the covariance of the real part to the imaginary part is minus the covariance of the imaginary part to the real one [1].

B. Connection to previous approaches

It is straightforward to show that previous approaches in [4], [5], [6] belong to the SRKHS type with kernels as described in the following section and, therefore, are limited compared to the WRKHS. An interesting singular case of this SRKHS formulation corresponds to the scenario where the real and imaginary parts are not related. In this case, $\mathbf{k}_{\mathrm{jr}} = -\mathbf{k}_{\mathrm{rj}} = 0$ and the formulation yields,

$$\begin{pmatrix} f_{\mathrm{r}}(\mathbf{x}_\cdot) \\ f_{\mathrm{j}}(\mathbf{x}_\cdot) \end{pmatrix} = \begin{pmatrix} \mathbf{k}_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{X})\boldsymbol{\alpha}_{\mathrm{r}} \\ \mathbf{k}_{\mathrm{rr}}(\mathbf{x}_\cdot, \mathbf{X})\boldsymbol{\alpha}_{\mathrm{j}} \end{pmatrix}, \tag{28}$$

where the kernel in (21) is real. This is the simple *complexification* described in [13], [12] based on building a complex Hilbert space considering functions of the form $f(\cdot) = f_r(\cdot) + \mathrm{j}f_j(\cdot)$ where f_{r} and f_{j} are in class $\mathcal{F}$. Note that f_{r} and f_{j} must be real for every input, and that being in the same class implies the same real-valued reproducing kernel for the real and the imaginary parts [13]. This procedure is limited in that it amounts to learning the real and imaginary parts independently but with the same kernel.

V. Kernel Design

The kernel is a key tool in RKHS: it encodes our assumptions about the function that we wish to learn. The kernel measures *similarity* between inputs. In this section we first analyze kernels, including previous proposals. Then we face the design of pseudo-kernels, for WRKHS.

A. Kernel

In [3], [5] a complex-valued Gaussian kernel approach is proposed. The kernel used was as an extension of the real Gaussian kernel:

$$k_{\mathbb{C}}(\mathbf{x}, \mathbf{x}') = \exp\left(-(\mathbf{x} - \mathbf{x}'^*)^{\mathsf{T}}(\mathbf{x} - \mathbf{x}'^*)/\gamma\right). \tag{29}$$

If we separate the real and imaginary parts, $\mathbf{x} = \mathbf{x}_{\mathrm{r}} + \mathrm{j}\mathbf{x}_{\mathrm{j}}$ and $\mathbf{x}' = \mathbf{x}'_{\mathrm{r}} + \mathrm{j}\mathbf{x}'_{\mathrm{j}}$, then it follows that

$$\begin{aligned} k_{\mathbb{C}}(\mathbf{x}, \mathbf{x}') &= \exp\left(-(|\mathbf{x}_{\mathrm{r}} - \mathbf{x}'_{\mathrm{r}}|^2 - |\mathbf{x}_{\mathrm{j}} + \mathbf{x}'_{\mathrm{j}}|^2)/\gamma\right) \\ &\quad\cdot \exp\left(-2\mathrm{j}(\mathbf{x}_{\mathrm{r}} - \mathbf{x}'_{\mathrm{r}})^{\mathsf{T}}(\mathbf{x}_{\mathrm{j}} + \mathbf{x}'_{\mathrm{j}})/\gamma\right) \\ &= \exp\left(-|\mathbf{x}_{\mathrm{r}} - \mathbf{x}'_{\mathrm{r}}|^2/\gamma\right)\exp\left(|\mathbf{x}_{\mathrm{j}} + \mathbf{x}'_{\mathrm{j}}|^2/\gamma\right) \\ &\quad\cdot \left(\cos(2(\mathbf{x}_{\mathrm{r}} - \mathbf{x}'_{\mathrm{r}})^{\mathsf{T}}(\mathbf{x}_{\mathrm{j}} + \mathbf{x}'_{\mathrm{j}})/\gamma)\right. \\ &\quad\left. - \mathrm{j}\sin(2(\mathbf{x}_{\mathrm{r}} - \mathbf{x}'_{\mathrm{r}})^{\mathsf{T}}(\mathbf{x}_{\mathrm{j}} + \mathbf{x}'_{\mathrm{j}})/\gamma)\right), \end{aligned} \tag{30}$$

where $|\cdot|$ is the l^2-norm.

Another complex-valued kernel was proposed in [6] also within a SRKHS. The authors of the proposal remarked that the kernel in (29) does not have the intuitive physical meaning of a measure of similarity of the samples and propose the so-called independent kernel:

$$\begin{aligned} k_{\mathrm{ind}}(\mathbf{x}, \mathbf{x}') &= \kappa_{\mathbb{R}}(\mathbf{x}_{\mathrm{r}}, \mathbf{x}'_{\mathrm{r}}) + \kappa_{\mathbb{R}}(\mathbf{x}_{\mathrm{j}}, \mathbf{x}'_{\mathrm{j}}) && (31) \\ &\quad + \mathrm{j}\left(\kappa_{\mathbb{R}}(\mathbf{x}_{\mathrm{r}}, \mathbf{x}'_{\mathrm{j}}) - \kappa_{\mathbb{R}}(\mathbf{x}_{\mathrm{j}}, \mathbf{x}'_{\mathrm{r}})\right), && (32) \end{aligned}$$

where $\kappa_{\mathbb{R}}$ is a real kernel of real inputs.

These two kernels, $k_{\mathbb{C}}$ and k_{ind}, were introduced in [5] and [6] as part of a machine of the type in (21). First conclusion, in the view of Proposition 4.1, is that these methods belong to SRKHS. For SRKHS $\mathbf{k}_{\mathrm{rr}}(\mathbf{x}, \mathbf{x}') = \mathbf{k}_{\mathrm{jj}}(\mathbf{x}, \mathbf{x}')$ the pseudo-kernel cancels and the kernel matrix is limited to have a particular symmetry. In the view of Proposition 4.2 both kernels $k_{\mathbb{C}}$ and k_{ind} are of the form given in (23) $k(\mathbf{x}, \mathbf{x}') = k_{\mathrm{rr}}(\mathbf{x}, \mathbf{x}') - \mathrm{j}k_{\mathrm{rj}}(\mathbf{x}, \mathbf{x}')$, with $k_{\mathrm{rr}}(\mathbf{x}, \mathbf{x}') = k_{\mathrm{rr}}(\mathbf{x}', \mathbf{x})$ and $k_{\mathrm{rj}}(\mathbf{x}, \mathbf{x}') = -k_{\mathrm{rj}}(\mathbf{x}', \mathbf{x})$. These symmetries for one-dimensional complex-valued inputs are illustrated in Fig. 1 and Fig. 3. In these figures it can be observed the particular way these kernels measure *similarity* between inputs. The kernel $k_{\mathbb{C}}$ in (29), measures similarities between real parts of the inputs with $|\mathbf{x}_{\mathrm{r}} - \mathbf{x}'_{\mathrm{r}}|^2$, while for the imaginary ones it uses $|\mathbf{x}_{\mathrm{j}} + \mathbf{x}'_{\mathrm{j}}|^2$. Also, it is not stationary and has an oscillatory behavior. We illustrate these features in the example in Fig. 1. The exponent in the kernel may easily grow large and positive as can be observed in the example depicted in Fig. 2. This might cause numerical problems in the learning algorithms. On the other hand, the kernel k_{ind} in (31) has a very particular structure since it follows the structure in (23) but it is not written as a function of the complex-valued inputs, but as a function of the real and imaginary parts of the inputs. This way of measuring the similarity between the inputs produces a particular *cross*-shape, as shown in the example in Fig. 3. Again, notice that because of the high constant values along the real and imaginary axis this kernel may be not useful for a wide range of systems.

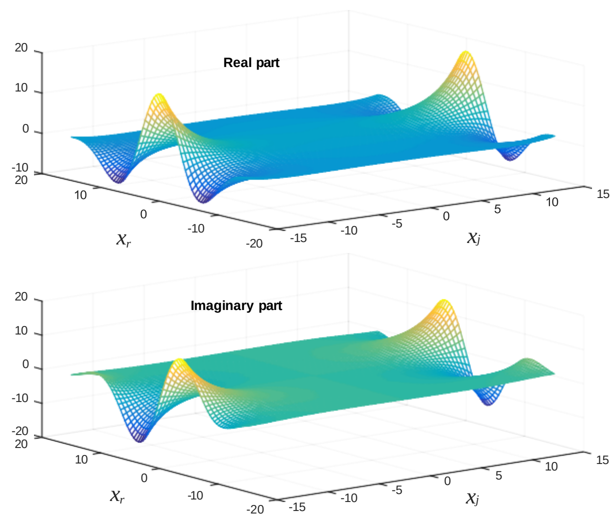

Fig. 1. Real and imaginary parts of $k_{\mathbb{C}}(x, x')$ when $x = 0 + \mathrm{j}0$ and x' with real and imaginary parts in $[-15, 15]$, $\gamma = 80$.

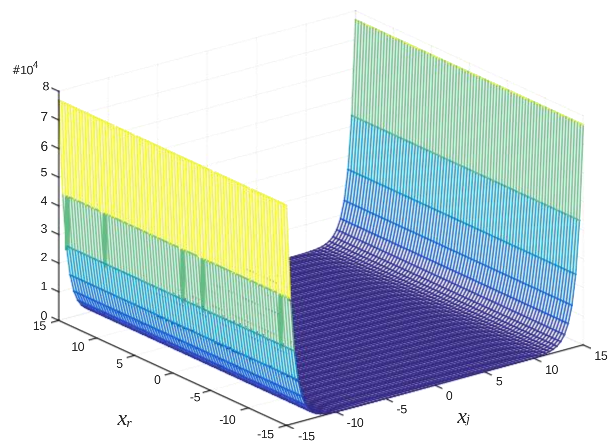

Fig. 2. $k_{\mathbb{C}}(x, x')$ when $x = x'$ with real and imaginary parts in $[-15, 15]$, $\gamma = 80$.

Our conclusion is that, in WRKHS with null pseudo-kernel, enforcing a complex-valued kernel is counterproductive unless you identify, for the particular problem at hand, a skew-symmetry of the kind $k_{rj}(\mathbf{x}, \mathbf{x}') = -k_{rj}(\mathbf{x}', \mathbf{x})$. Note that a null pseudo-kernel and a real-valued kernel yields the complexification case in (28). The way that similarity is measured and the structure of the kernel function are two important issues to take into account when designing the kernel. Regarding similarity, since we are working in a model with complex-valued inputs, we propose using the difference between complex-valued inputs, $\mathbf{d}_\mathbf{x} = (\mathbf{x} - \mathbf{x}')$. In addition, if an isotropic behavior is desired, functions $k_{rr}(\mathbf{x}, \mathbf{x}')$ and $k_{rj}(\mathbf{x}', \mathbf{x})$ in (23) could better rely on the inner product $\mathbf{d}_\mathbf{x}^H \mathbf{d}_\mathbf{x} = (\mathbf{x} - \mathbf{x}')^H(\mathbf{x} - \mathbf{x}')$ rather than on expressions of real and imaginary parts. We propose as

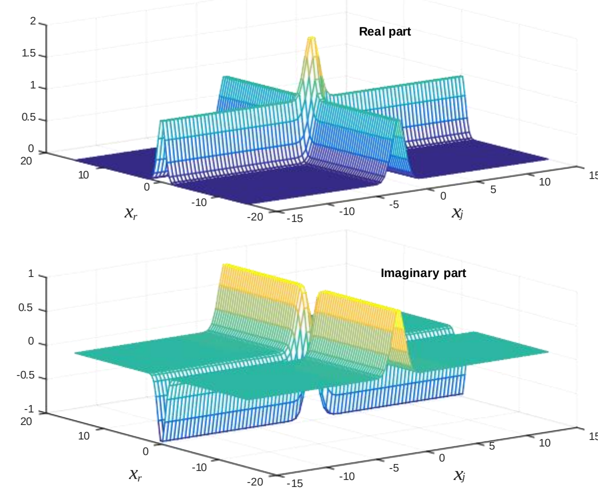

Fig. 3. Real and imaginary parts of $k_{\mathrm{ind}}(x, x')$ when $x = 0 + \mathrm{j}0$ and x' with real and imaginary parts in $[-15, 15]$, $\gamma = 0.8$.

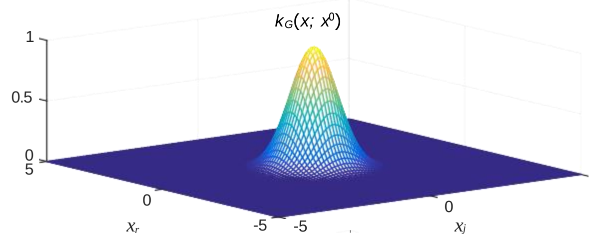

Fig. 4. Examples for $k_G(x, x')$ with null imaginary part in (33) when $x = 0 + \mathrm{j}0$ and x' with real and imaginary parts in $[-5, 5]$, $\gamma = 0.8$.

an isotropic stationary kernel the following adaptation of the real-valued Gaussian kernel depending on the inner product $\mathbf{d}_\mathbf{x}^H \mathbf{d}_\mathbf{x}$:

$$k_{rr}(\mathbf{x}, \mathbf{x}') = k_G(\mathbf{x}, \mathbf{x}') = \exp\left(-\mathbf{d}_\mathbf{x}^H \mathbf{d}_\mathbf{x}/\gamma\right). \qquad (33)$$

An example of this kernel is shown in Fig. 4. We will use this kernel for the equalization problem in the Experiments Section, where usually real and imaginary parts in a digital communication constellations are independent but exhibit similar properties.

B. Kernel and Pseudo-Kernel

In WRKHS we have both the kernel, $k(\mathbf{x}, \mathbf{x}')$, and the pseudo-kernel, $\tilde{k}(\mathbf{x}, \mathbf{x}')$. These kernels can be written as functions of the kernels of the real part, $\gamma_{rr}(\mathbf{x}, \mathbf{x}')$, the imaginary part, $\gamma_{jj}(\mathbf{x}, \mathbf{x}')$ and the real-imaginary parts $\gamma_{jr}(\mathbf{x}, \mathbf{x}')$ and $\gamma_{rj}(\mathbf{x}, \mathbf{x}')$, as in (17)-(18). Therefore the design is quite open. We bring here two particular but interesting cases. First one is the scenario where real and imaginary parts are independent but exhibit different properties and different kernels should be used. As a second case we design a kernel for the scenario where real and imaginary parts are not independent.

1) Different kernels for the real and imaginary parts: If the real and imaginary parts need different kernels we may use a real-valued kernel for the real part, $\gamma_{rr}(\mathbf{x}, \mathbf{x}')$, and another real-valued design for the imaginary one, $\gamma_{jj}(\mathbf{x}, \mathbf{x}')$, assuming independence between real and imaginary parts. The kernels in (17)-(18) yield

$$k(\mathbf{x}_\cdot, \mathbf{x}(i)) = \gamma_{rr}(\mathbf{x}_\cdot, \mathbf{x}(i)) + \gamma_{jj}(\mathbf{x}_\cdot, \mathbf{x}(i)),$$
$$\tilde{k}(\mathbf{x}_\cdot, \mathbf{x}(i)) = \gamma_{rr}(\mathbf{x}_\cdot, \mathbf{x}(i)) - \gamma_{jj}(\mathbf{x}_\cdot, \mathbf{x}(i)). \quad (34)$$

Note that this scenario is simple, but the SRKHS is not a valid framework to explain it in a complex-valued formalism and a pseudo-kernel is needed. Besides, the resulting kernels are real-valued.

2) Non-independent real and imaginary parts: This scenario can be easily handled by using the concept of *separable kernel* and *sum of separable* (SoS) kernels [11]. In the complex case a mixed effect regularizer (MER) translates into

$$k(\mathbf{x}_\cdot, \mathbf{x}(i)) = 2\sum_{q=1}^{Q} k^{(q)}(\mathbf{x}_\cdot, \mathbf{x}(i)), \quad (35)$$

$$\tilde{k}(\mathbf{x}_\cdot, \mathbf{x}(i)) = 2\mathrm{j}\sum_{q=1}^{Q} \omega^{(q)}\tilde{k}^{(q)}(\mathbf{x}_\cdot, \mathbf{x}(i)), \quad (36)$$

if we choose $k^{(q)}(\mathbf{x}_\cdot, \mathbf{x}(i))$ to be real-valued kernels of complex-valued inputs, where $0 < \omega^{(q)} < 1$.

VI. Experiments

A. Learning with WRKHS

To illustrate the learning with WRKHS we bring here two synthetic experiments. In the first one we learn with a different similarity measurement for the real and the imaginary parts. We use the WRKHS solution in (20) with the kernel and pseudo-kernel in (34). In the second scenario we exploit the relation between the real and imaginary parts of the output, using (35)-(36).

1) Real and imaginary parts: We propose to learn a non-linear function of the type $y(x) = y_r(x) + \mathrm{j}y_j(x)$, where $x = x_r + \mathrm{j}x_j$ and in this experiment

$$y_r(x) = \sum_{r=-1}^{1} \mathrm{Sinc}(1.2x_r + 2r) \cdot \mathrm{Sinc}(1.2x_j - 2r)$$
$$y_j(x) = \mathrm{Sinc}(0.2x_j - 1.5). \quad (37)$$

We generate $n = 200$ random training samples of $y(x)$ in the range $[-5, 5]$. In (34), $k_{rr}(x, x')$ and $k_{jj}(x, x')$ are $k_G(x, x')$ in (33) with $\gamma = 1$ and $\gamma = 3.5$, respectively. Since the imaginary part of the output has a softer behavior, the optimal hyperparameter of the kernel is larger than for the real part. The result is included in Fig. 5, where the training samples are plotted in red circles. Without a pseudo-kernel we can not use a different similarity measurement for the real and the imaginary parts. In Fig. 6 we include the result of the learning of the imaginary part if the same kernel with $\gamma = 1$ is used for both the real and the imaginary parts. In this case we observe that the learning of the imaginary part exhibits quite a larger error. The overall mean square error (MSE) with the WRKHS is -54.9 dB while WRKHS with null pseudo-kernel is -38.8 dB.

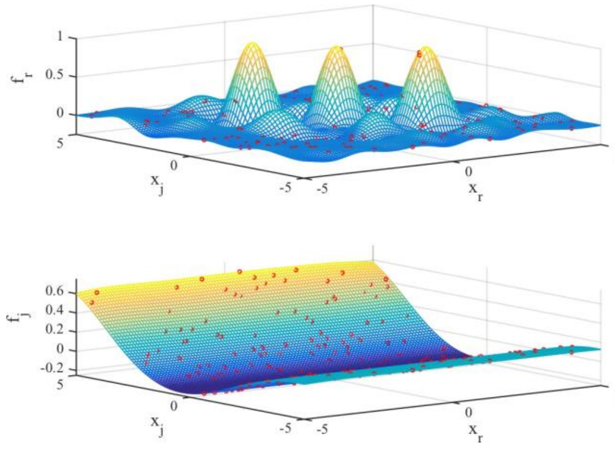

Fig. 5. Real (top) and imaginary (bottom) parts of the WRKHS estimation $f(\mathbf{x})$ versus the real and imaginary parts of the input. The training samples are depicted as red circles.

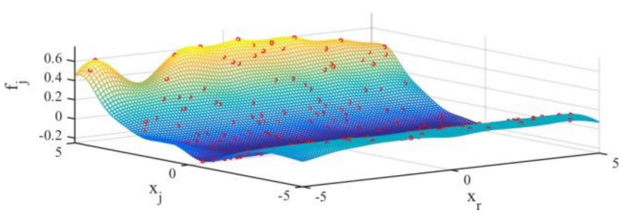

Fig. 6. Imaginary part of the WRKHS estimation $f(\mathbf{x})$ with null pseudo-kernel versus the real and imaginary parts of the input. Same kernel for the real and imaginary parts are used. The training samples are depicted as red circles.

2) Non-independentreal and imaginary parts: We propose to learn a non-linear function of the type $y(x) = y_r(x) + \mathrm{j}y_j(x)$, where $y_r(x) = z_r + \omega z_j$, $y_j(x) = z_j + \omega z_r$,

$$z_r(x) = \mathrm{Sinc}(0.5x_r) \cdot \mathrm{Sinc}(0.5x_j)$$
$$z_j(x) = 0.1 \cdot \mathrm{Sinc}(0.3x_j). \quad (38)$$

and $x = x_r + \mathrm{j}x_j$. We generate $n = 200$ random training samples of $y(x)$ in the range $[-5, 5]$. In (34) we use MER and SoS but with just one term, $Q = 1$, setting $\omega = \omega^{(1)} = 0.3$. The kernel $k^{(1)}(\mathbf{x}, \mathbf{x}')$ in (35)-(36) is the Gaussian, $k_G(x, x')$, in (33) with $\gamma = 2$. The result of the learning is depicted in Fig. 7. Note that the real and imaginary parts are similar but not equal. The MSE of this solution is -45.3 dB, while the solution for $\omega^{(1)} = 0$, that corresponds to SRKHS, is -41.8 dB.

B. Nonlinear channel equalization

The main advantage of the analysis of the WRKHS in this paper is that it greatly facilitates decisions to make through the design to select the simplest model. To illustrate this point we bring here the nonlinear channel equalization in [5] and [8]. We use the channel considered in [5] and [8]. It consists of a linear filter $t(n) = (-0.9 + 0.8\mathrm{j}) \cdot s(n) + (0.6 - 0.7\mathrm{j}) \cdot s(n-1)$

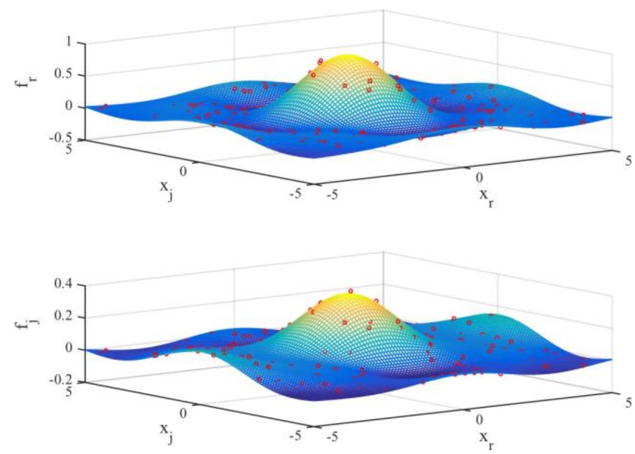

Fig. 7. Real (top) and imaginary (bottom) parts of the WRKHS estimation $f(\mathbf{x})$ versus the real and imaginary parts of the input. A separable kernel was used with $Q = 1$ and $\omega^{(1)} = 0.3$. The training samples are depicted as red circles.

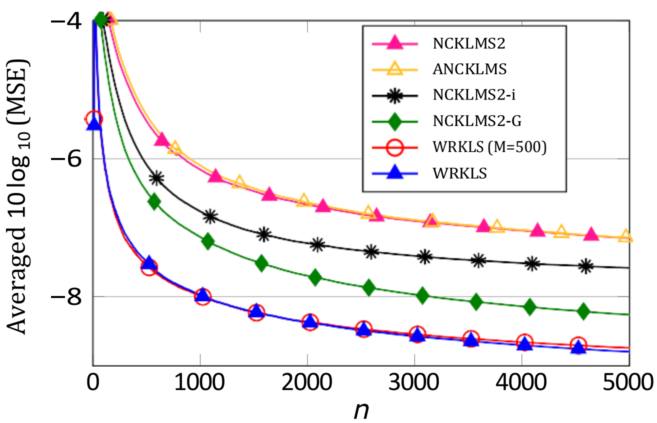

Fig. 8. Averaged MSE along n for NCKLMS2, ANCKLMS, NCKLMS2-i, NCKLMS2-G, the WRKLS and the WRKLS with M=500 basis for the strong nonlinear channel equalization problem and the circular input case.

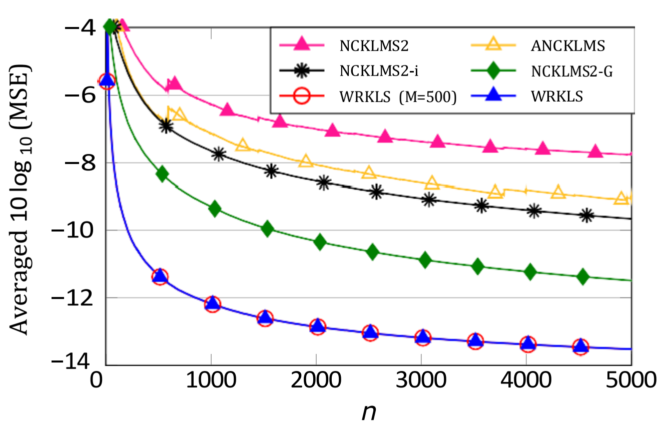

Fig. 9. Averaged MSE along n for NCKLMS2, ANCKLMS, NCKLMS2-i, NCKLMS2-G, the WRKLS and the WRKLS with M=500 basis for the strong nonlinear channel equalization and the noncircular input case ($\rho = 0.1$).

and a memoryless nonlinearity. The nonlinearity is $q(n) = t(n) + (0.2 + 0.25j) \cdot t^2(n) + (0.12 + 0.09j) \cdot t^3(n)$ and it is labeled as *strong nonlinear channel*. The input signals had the form $s(n) = 0.70(\sqrt{1-\rho^2}X(n) + j\rho Y(n))$, and $X(n)$ and $Y(n)$ were Gaussian random variables. Note that the real and the imaginary parts of the input signals were generated independently. The input signals are circular for $\rho = 1/\sqrt{2}$ and highly noncircular if ρ approaches 0 or 1. At the receiver end of the channel, the signal $q(n)$ was corrupted by additive white circular Gaussian noise with the SNR set to 16 dB, as in [5]. The aim of the channel equalization task is to construct an inverse filter, which acts on the received signal $r(t)$ and reproduces the original input signal $s(n)$ as close as possible. To this end, the inputs to the equalizer are the sets of samples $\mathbf{x} = [r(n+D), r(n+D-1), \cdots, r(n+D-L+1)]^\top$, where $L > 0$ is the filter length and D is the equalization time delay.

The authors in [5] first proposed a machine than can be easily proved to be within the SRKHS framework to later improve it adding more terms [8]. Kernel used was k_C in (29). The algorithms proposed, the NCKLMS2 and the ACKLMS, are of the kernel LMS type and they use novelty sparsification criterion to reduce the number of training samples used to compute the solution. We use the code available in [14] to run these algorithms. All the parameters required for both algorithms (γ in kernel and step update parameter) are set to the values described in [5] and [8], except for the *strong nonlinear channel* in the noncircular case, where in order to ensure convergence it is needed to increase γ to $\gamma = 20^2$. The parameters for the sparsification are $\delta_1 = 0.15$ and $\delta_2 = 0.2$. Also, $L = 5$ and $D = 2$, on a set of 5000 samples of the input signal considering both the circular and the noncircular ($\rho = 0.1$) cases and the described nonlinear channel. In all cases the results in [5] and [8] are averaged over 500 trials where the input signals $s(n)$ and noise are generated randomly.

1) Circular case, $\rho = 1/\sqrt{2}$: From the problem at hand we know that the output has independent real and imaginary parts. Since there is no relation between real and imaginary parts we have $k_{rj}(\mathbf{x}, \mathbf{x}') = 0$ and the kernel should be chosen real-valued. Hence, we avoid the design of a skew-symmetric imaginary part $k_{rj}(\mathbf{x}, \mathbf{x}') = -k_{rj}(\mathbf{x}', \mathbf{x})$. Besides, using the same kernel for the real and the imaginary parts is a suitable assumption in the circular case, since we expect both to behave similarly in terms of similarity. We conclude that the solution lies within a WRKHS with null pseudo-kernel, i.e. (21). The Gaussian kernel $k_{rr}(\mathbf{x}, \mathbf{x}') = k_G(\mathbf{x}, \mathbf{x}')$ in (33) with null imaginary part is a suitable solution in this equalization problem [15], [16]. The values for the hyperparameters are set to $\gamma = 8.92$ and $\lambda = 0.32$. To compare the solution in (21) with coefficients given by (27), we derive the kernel recursive least square (RKLS) [9] version of WRKHS with null pseudo-kernel, denoted as WRKLS. This approach adaptively computes the solution for each new input and retains M basis or samples to compute it, reducing the computational complexity. We first compare the performances of the NCKLMS2 [5], the ACKLMS [8] and the SRKLS. We include two versions of the WRKHS: with basis removal criterion ($M = 500$) and without basis removal criterion. Note that the number of bases used by the NCKLMS2 and the ACKLMS algorithms with the novelty sparsification criterion grows above 2000 in the experiments in this section, and therefore the choice of $M = 500$ is far below that number. In Figs. 8 we can observe the averaged MSE along the input samples for these four methods and the channels and inputs described. The MSE value depicted for each sample is the averaged MSE for all previous outputs, as in [5], [14].

2) Non-circular case, $\rho = 0.1$: In the non-circular case we have a different behavior for the real and imaginary part of the output, but just in the scaling. The similarity between inputs remains the same and we may use the same kernel for the real and imaginary part. The solution for the coefficients will change, but just in scale. The module of $\boldsymbol{\alpha}_r$ in (27) increases while the module of $\boldsymbol{\alpha}_j$ decreases. Therefore, we may apply exactly the same solution as in the circular case. In Figs. 9 we compare the performance of the WKRLS with no basis selection and 500 basis to the results of the mean square error for the NCKLMS2 and the ACKLMS with sparsification. The values for the hyperparameters are set to $\gamma = 10.4$ and $\lambda = 0.18$.

3) Discussion: It can be observed in the figures the remarkable good results of the WKRLS in all the cases, *strong* nonlinear channels and circular or noncircular signals. With only $M = 500$ bases used for the prediction, the averaged MSE is very close to the method using all samples. When comparing with the NCKLMS2 and ACKLMS, the SKRLS remarkably outperforms both algorithms that use above 2000 basis. The gains are not only due to the better capabilities of the recursive and basis removal approach used but to the model selection. From the results in this paper we conclude first that the best option is a WKRHS with real-valued kernel and null pseudo-kernel. Since real and imaginary parts are independent and they exhibit similar similitude measure up to a scaling. Hence, a good selection of kernel and (null) pseudo-kernel improves the final results. To further illustrate this point, we also include the NCKLMS2 algorithm with the kernel used in the WRKLS, in (33). This algorithm is labeled as NCKLMS2-G in the figures. The parameters for this algorithm are the same that were previously used for the NCKLMS2, and the novelty criterion is again used for the sparsification. Note that by using the proposed kernel we obtain a much better performance in all cases when comparing with the NCKLMS2 or ACKLMS algorithms. Finally, for the sake of completeness we also include in the comparison the method in [6]: the NCKLMS2 algorithm with the independent kernel (31) with κ_R being the real-valued Gaussian kernel. We labeled this algorithm as NCKLMS2-i in the figures. Although this kernel seems more suitable for the problem at hand than the complex Gaussian kernel (29), as shown in the figures, the performance is not as good as the NCKLMS2-G algorithm. The reason, again, is a sub-optimal model selection, where we have a kernel with non-null imaginary part and with the particular *cross*-shape shown in Fig. 3.

VII. Conclusions

Complex-valued kernel regression has been tackled by adapting the real-valued approaches in a straight forward manner [4], [5], [6]. As in strictly linear estimation, this is useful in many scenarios, it is not efficient in others. We develop a novel solution, WRKHS, to avoid this limitation. The solution is based on including a pseudo-kernel. The resulting structure of the regressor resembles that of the widely complex linear solutions, being capable of learning the real and imaginary parts of the output regardless of the relation between them. In the experiments we show how systems with independent and different real and imaginary parts are better learned. Regression for correlated real and imaginary parts is also improved. We introduce some proposals for the kernel and the pseudo-kernel. The complex-valued nature of the kernel and the pseudo-kernel is also discussed. When the pseudo-kernel cancels, special attention is to be paid to the imaginary part of the kernel. In this case the imaginary part must be skew symmetric or null. We apply these concepts to face the nonlinear channel equalization, minimizing the overall error. We believe the results in this paper are relevant to better face any complex-valued regression problem, using complex-valued formulation.

References

[1] P. J. Schreier and L. L. Scharf, *Statistical Signal Processing of Complex-Valued Data. The Theory of Improper and Noncircular Signals*. Cambridge, UK: Cambridge University Press, 2010.

[2] A. Hirose, *Complex-Valued Neural Networks: Advances and Applications*, ser. IEEE Press Series on Computational Intelligence. Wiley, 2013.

[3] I. Steinwart, D. Hush, and C. Scovel, "An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels," *IEEE Trans. Information Theory*, vol. 52, no. 10, pp. 4635–4643, Oct 2006.

[4] T. Ogunfunmi and T. K. Paul, "On the complex kernel-based adaptive filter." in *IEEE Int. Symp. on Circuits and Systems, ISCAS*, 2011, pp. 1263–1266.

[5] P. Bouboulis and S. Theodoridis, "Extension of wirtinger's calculus to reproducing kernel Hilbert spaces and the complex kernel LMS," *IEEE Trans. Signal Processing*, vol. 59, no. 3, pp. 964–978, 2011.

[6] F. Tobar, A. Kuh, and D. Mandic, "A novel augmented complex valued kernel LMS," in *IEEE 7th Sensor Array and Multichannel Signal Processing Workshop, SAM*, 2012, pp. 473–476.

[7] A. Papaioannou and S. Zafeiriou, "Principal component analysis with complex kernel: The widely linear model," *IEEE Trans. Neural Networks and Learning Systems*, vol. 25, no. 9, pp. 1719–1726, Sept 2014.

[8] P. Bouboulis, S. Theodoridis, and M. Mavroforakis, "The augmented complex kernel LMS," *IEEE Trans. Signal Processing*, vol. 60, no. 9, pp. 4962–4967, 2012.

[9] S. Van Vaerenbergh, M. Lazaro-Gredilla, and I. Santamar´ıa, "Kernel recursive least-squares tracker for time-varying regression," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 8, pp. 1313–1326, Aug 2012.

[10] B. Scho¨lkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, ser. Adaptive computation and machine learning. Cambridge, Massachusetts: MIT Press, 2002.

[11] M. A. A´ lvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Found. Trends Mach. Learn.*, vol. 4, no. 3, pp. 195–266, Mar. 2012.

[12] V. I. Paulsen, "An introduction to the theory of reproducing kernel Hilbert spaces," https://www.math.uh.edu/~vern/rkhs.pdf, 9 2009, [Online].

[13] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, pp. 337–404, 1950.

[14] P. Bouboulis, "The Augmented Complex Kernel LMS Matlab and C Code," http://bouboulis.mysch.gr/kernels.html, [Online].

[15] F. Pe´rez-Cruz, S. Van Vaerenbergh, J. Murillo-Fuentes, M. Lazaro-Gredilla, and I. Santamaria, "Gaussian processes for nonlinear signal processing: An overview of recent advances," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 40–50, July 2013.

[16] F. Pe´rez-Cruz, J. Murillo-Fuentes, and S. Caro, "Nonlinear channel equalization with Gaussian processes for regression," *IEEE Trans. Signal Processing*, vol. 56, no. 10, pp. 5283–5286, Oct 2008.