Complex Gaussian Processes for Regression

Rafael Boloix-Tortosa*, Juan José Murillo-Fuentes*, F. Javier Payán-Somet*, and Fernando Pérez-Cruz†

* Dep. Signal Theory and Communications, Universidad de Sevilla, Spain.
† Swiss Data Science Center, 8006 Zurich, Switzerland and Department of Signal Theory and Communications, University Carlos III de Madrid, Spain.

This is the accepted version of the paper. Reference:

Final Title: ”Complex Gaussian Processes for Regression”

Journal: IEEE Transactions on Neural Networks and Learning Systems

DOI: 10.1109/TNNLS.2018.2805019

This article has been accepted for publication in the journal IEEE Transactions on Neural Networks and Learning Systems, but has not been fully edited. Content may change prior to final publication. Citation information: DOI: 10.1109/TNNLS.2018.2805019, ©2018 IEEE.

Complex Gaussian Processes for Regression

Rafael Boloix-Tortosa*,Juan José Murillo-Fuentes, *Senior Member, IEEE,* F. Javier Payán-Somet, and Fernando Pérez-Cruz, *Senior Member, IEEE,*

***Abstract*—In this paper we propose a novel Bayesian solution for nonlinear regression in complex fields. Previous solutions for kernels methods usually assume a *complexification* approach, where the real-valued kernel is replaced by a complex-valued one. This approach is limited. Based on results in complex-valued linear theory and Gaussian random processes we show that a *pseudo-kernel* must be included. This is the starting point to develop the new complex-valued formulation for Gaussian process for regression (CGPR). We face the design of the covariance and pseudo-covariance based on a convolution approach and for several scenarios. Just in the particular case where the outputs are proper, the pseudo-kernel cancels. Also, the hyperparameters of the covariance can be learnt maximizing the marginal likelihood using Wirtinger's calculus and patterned complex-valued matrix derivatives. In the experiments included, we show how CGPR successfully solve systems where real and imaginary parts are correlated. Besides, we successfully solve the nonlinear channel equalization problem by developing a recursive solution with basis removal. We report remarkable improvements compared to previous solutions: a 2-4 dB reduction of the MSE with just a quarter of the training samples used by previous approaches.**

***Index Terms*—Gaussian processes, regression, complex-valued processes, kernel methods.**

I. Introduction

NOWADAYS complex-valued signals model a vast range of nowadays systems in science and engineering such as telecommunications, optics and acoustics among others. Complex-valued signal processing allows to natively process complex-valued sequences, like electromagnetic signals. Hence, complex-valued signal processing is of fundamental interest. Signal processing for complex-valued signals has been widely studied in the linear case, see [1] and references therein. The nonlinear processing of complex-valued signals has been addressed from different points of view, such as complex-valued nonlinear adaptive filtering [2], neural networks [3], [4] and, recently, using reproducing kernel Hilbert spaces (RKHS) [5]. Some complex kernel-based algorithms have been lately proposed for classification [6], regression [7], [8], [9] and mainly for kernel principal component analysis [10]. Regarding regression, in [11] the authors propose a complex-valued kernel based on the results in [6] and face the derivative of cost functions by using Wirtinger's derivatives. Same kernel is adopted in [7]. In [9] 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). In the framework of kriging some complex-valued scenarios have also been addressed [12], [13], [14].

The methods above proposed for regression deal with complex-valued inputs and outputs by either 1) learning the real and imaginary parts independently; 2) using a straightforward adaptation of a real-valued approach; or, 3) learning a vector with the real and imaginary parts stacked in an augmented vector. The first approach is suboptimal for systems where the real and imaginary parts are not independent. The straightforward adaptation of real-valued versions is limited to proper systems, as *strictly* linear approaches [1]. *Complexification* of real RKHSs [15] lies within this group. For non-proper systems *widely* linear solutions are needed. The last option fits any scenario, but the complex valued formulation is lost, which limits the native interpretation of the complex sequence and applicability of this procedure. Furthermore, the design of the kernel between the real and the imaginary parts remains an open problem. To the best of our knowledge, except for [8], where an augmented version is discussed for the KLMS, there is no general complex-valued formulation of a nonlinear regression algorithm based on kernels or covariances. In this paper, we propose a new complex-valued algorithm working both for proper and non-proper systems.

The Gaussian process (GP) [16] is a Bayesian nonparametric framework for inference. It has has attracted increasing attention from the machine learning community for its many applications, such as in regression, classification [17], [18], adaptive control [19], multitask learning [20], or data association [21], among others. Gaussian processes for regression (GPR) [16], [22], [23] are kernel methods that provide a full conditional statistical description for the predicted variable. The covariance matrix of the GPR plays the role of the kernel. In [24] we developed complex-valued GPR for proper systems. A proper complex random signal is uncorrelated with its complex conjugate [25], and hence the pseudo-covariance cancels. The solution in [24] can be described as a straightforward adaptation of a real-valued approach as in [5], [7], [8], [9]. In this paper we include the pseudo-covariance of a Gaussian process into GPR to develop a novel approach for complex-valued systems, hereafter denoted as complex GPR (CGPR). With this result we prove that another kernel matrix is needed to properly model any given system, including all non-proper ones. We also tackle the maximization of the marginal likelihood in the complex-valued scenario. Since it depends on a complex Hermitian matrix, generalized complex-valued derivatives are used [26].

The design of a good covariance and pseudo-covariance

R. Boloix-Tortosa, Juan José Murillo-Fuentes, and F. Javier Payán-Somet are with the Departamento Teoría de la Señal y Comunicaciones, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Camino de los Descubrimientos sn, 41092 Sevilla, Spain. e-mail: {rboloix,murillo,jpayan}@us.es. Fernando Pérez-Cruz is with the Computer Science Department, Stevens Institute of Technology, 1 Castle Point Terrace, Hoboken, NJ 07030 USA. e-mail: fperezcr@stevens.edu.

Thanks to Spanish government (Ministerio de Economía y Competitividad, TEC2016-78434-C3-02-R) and European Union (FEDER) for funding.

function is crucial for CGPRs to provide accurate nonlinear solutions. Under the Gaussian process regression point of view, the covariance function measures *similarity* between inputs [16]. The construction of the imaginary part quite depends on the system model. We propose to apply the convolution approach [27], [28] to generate a covariance and pseudo-covariance that explain the case where real and imaginary parts are correlated, even when shifted or displaced. This includes the proper case where the cross covariance between the real and the imaginary parts is either null or skew-symmetric.

As benchmark we propose the nonlinear channel equalization problem in digital communications. The authors in [11], [8] propose to build KLMS adaptive filtering of complex signals. We face this problem from the CGPR point of view. The statistical properties of the to-be-learned outputs in the channel equalization problem are taken into consideration in the selection of the model and a recursive version with basis removal criterion is adapted [29]. Compared to the solutions in [11], [8], [9] the CGPR exhibits a 2-4 dB gain with only a quarter of the training samples used by state-of-the-art approaches. Other experiments are also included to analyze other scenarios.

The paper is organized as follows. Next section includes the definition of the CGPR, including the derivation of the proper case for CGPR in Subsection II-C. Section III is devoted to the analysis of the structure of the complex-valued covariance and pseudo-covariance. We develop in Section IV the optimization procedure to set the kernel hyperparameters applying Wirtinger's calculus and patterned complex-valued matrix derivatives. We show the experimental results in Section V and conclude the paper in Section VI.

The notations used in the paper are as follows. For matrix $\mathbf{A}$, $\det \mathbf{A}$ is its determinant, $\mathrm{Tr}(\mathbf{A})$ is its trace, $[\mathbf{A}]_{l,q}$ is its (l,q) entry, $\mathbf{A}^\top$ represents the transpose, $\mathbf{A}^\mathrm{H}$ the Hermitian transpose, $\mathbf{A}^*$ represents the complex conjugate of its entries, and $\mathbf{A}^{-*} = (\mathbf{A}^*)^{-1}$. To denote the i-th sample of a vector we use $\mathbf{a}_i$. The real and imaginary parts are denoted by subindices r and j, respectively, i.e. $\mathbf{a} = \mathbf{a}_\mathrm{r} + \mathrm{j}\mathbf{a}_\mathrm{j}$, with $\mathrm{j} = \sqrt{-1}$. $\mathbb{E}[\cdot]$ refers to expectation. To denote the complex Gaussian distribution with mean vector $\boldsymbol{\mu}$, covariance matrix $\mathbf{K}$ and pseudo-covariance matrix $\tilde{\mathbf{K}}$ we use $\mathcal{N}\left(\boldsymbol{\mu}, \mathbf{K}, \tilde{\mathbf{K}}\right)$. The augmented covariance matrix, $\underline{\mathbf{K}}$, is given by

$$\underline{\mathbf{K}} = \begin{bmatrix} \mathbf{K} & \tilde{\mathbf{K}} \\ \tilde{\mathbf{K}}^* & \mathbf{K}^* \end{bmatrix}. \tag{1}$$

II. Complex Gaussian Process Regression

A. Complex-valued Gaussian process

GPR can be presented as a nonlinear regressor that expresses the input-output relation through function $f(\mathbf{x})$, known as latent function. This latent function follows a GP and underlies the regression problem

$$y = f(\mathbf{x}) + \epsilon, \tag{2}$$

where the error, ϵ, in the estimation of a real-valued output, y, is modeled as additive zero-mean Gaussian noise. In this paper, we consider that both inputs and outputs are complex-valued. The simpler real-valued input and complex-valued output case can be easily solved from the results herein. Each input at time i is a complex-valued column vector of dimension d, $\mathbf{x}_i \in \mathbb{C}^d$. For any input set $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ the latent function in (2) provides a multidimensional Gaussian complex-valued random vector $\mathbf{f} = [f(\mathbf{x}_1), ..., f(\mathbf{x}_n)]^\top$, where $f(\mathbf{x}_i) \in \mathbb{C}$. A complex random Gaussian vector $\mathbf{f}$ is characterized not only by its mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{K} = \mathbb{E}[(\mathbf{f}-\boldsymbol{\mu})(\mathbf{f}-\boldsymbol{\mu})^\mathrm{H}]$, but also by its complementary covariance or pseudo-covariance matrix $\tilde{\mathbf{K}} = \mathbb{E}[(\mathbf{f}-\boldsymbol{\mu})(\mathbf{f}-\boldsymbol{\mu})^\top]$, [1]. These matrices can be defined by kernels, $[\mathbf{K}]_{l,q} = k(\mathbf{x}_l, \mathbf{x}_q)$and $[\tilde{\mathbf{K}}]_{l,q} = \tilde{k}(\mathbf{x}_l, \mathbf{x}_q)$, respectively. The Gaussian process prior becomes

$$\begin{aligned} p(\mathbf{f}|\mathbf{X}) &= \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{K}, \tilde{\mathbf{K}}\right) \\ &= \frac{1}{\pi^n \sqrt{\det \underline{\mathbf{K}}}} \exp\left(-\frac{1}{2}(\underline{\mathbf{f}}-\underline{\boldsymbol{\mu}})^\mathrm{H} \underline{\mathbf{K}}^{-1} (\underline{\mathbf{f}}-\underline{\boldsymbol{\mu}})\right), \end{aligned} \tag{3}$$

where $\underline{\mathbf{f}} = [\mathbf{f}^\top \ \mathbf{f}^\mathrm{H}]^\top$ is the augmented vector for $\mathbf{f}$, $\underline{\boldsymbol{\mu}} = [\boldsymbol{\mu}^\top \ \boldsymbol{\mu}^\mathrm{H}]^\top$ is the augmented mean vector, and $\underline{\mathbf{K}}$ is the augmented covariance matrix (1). Without loss of generality, we consider zero-mean processes, $\mu(\mathbf{x}) = 0$.

B. Complex GP for Regression

In the learning process we condition the output of the GPR for some new observation $\mathbf{x}'$, given the training set $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, where the outputs $\mathbf{y} = [y_1, ..., y_n]^\top$ for a given set of observations $\mathbf{X}$ are known. First, we compute the joint distribution as follows. We assume that the additive noise ϵ in (2) follows an i.i.d. complex Gaussian distribution with zero mean, variance σ_ϵ^2 and pseudo-covariance $\rho\sigma_\epsilon^2$, with ρ being a complex number. The samples in the training set are i.i.d., hence the likelihood for the latent function at the training set is given by the factorized model

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f(\mathbf{x}_i)), \tag{4}$$

where $p(y_i|f(\mathbf{x}_i)) = \mathcal{N}(f(\mathbf{x}_i), \sigma_\epsilon^2, \rho\sigma_\epsilon^2)$[1]. Therefore, the likelihood is a complex multidimensional Gaussian $p(\mathbf{y}_n|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma_\epsilon^2\mathbf{I}_n, \rho\sigma_\epsilon^2\mathbf{I}_n)$. This likelihood and the prior in (3) yield the marginal likelihood or evidence

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{f})\, p(\mathbf{f}|\mathbf{X}) d\mathbf{f} = \mathcal{N}\left(\mathbf{0}, \mathbf{C}, \tilde{\mathbf{C}}\right), \tag{5}$$

where $\mathbf{C} = \mathbf{K} + \sigma_\epsilon^2\mathbf{I}_n$ and $\tilde{\mathbf{C}} = \tilde{\mathbf{K}} + \rho\sigma_\epsilon^2\mathbf{I}_n$. Given a test input vector $\mathbf{x}'$, the joint distribution of the training outputs $\mathbf{y}$ and $f' = f(\mathbf{x}'_1)$ is

$$\begin{bmatrix} \mathbf{y} \\ f' \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}, \tilde{\boldsymbol{\Lambda}}\right), \tag{6}$$

with

$$\boldsymbol{\Lambda} = \begin{bmatrix} \mathbf{C} & \mathbf{k}(\mathbf{X}, \mathbf{x}') \\ \mathbf{k}^\mathrm{H}(\mathbf{X}, \mathbf{x}') & k(\mathbf{x}', \mathbf{x}') \end{bmatrix}, \tag{7}$$

$$\tilde{\boldsymbol{\Lambda}} = \begin{bmatrix} \tilde{\mathbf{C}} & \tilde{\mathbf{k}}(\mathbf{X}, \mathbf{x}') \\ \tilde{\mathbf{k}}^\top(\mathbf{X}, \mathbf{x}') & \tilde{k}(\mathbf{x}', \mathbf{x}') \end{bmatrix} \tag{8}$$

[1]If the likelihood were not Gaussian, we can resort to Wrapped Gaussian processes [30], [31].

where $\mathbf{k}(\mathbf{X},\mathbf{x}') = [k(\mathbf{x}(1),\mathbf{x}'),\cdots,k(\mathbf{x}(n),\mathbf{x}')]^\top$ and $\tilde{\mathbf{k}}(\mathbf{X},\mathbf{x}') = [\tilde{k}(\mathbf{x}(1),\mathbf{x}'),\cdots,\tilde{k}(\mathbf{x}(n),\mathbf{x}')]^\top$.

The conditional distribution of f' given $\mathbf{y}$, i.e. the estimated output, yields the Gaussian distribution $p(f'|\mathbf{x}',\mathbf{X},\mathbf{y}) = \mathcal{N}(\mu_{f'},\sigma_{f'},\tilde{\sigma}_{f'})$, where

$$\underline{\mu}_{f'} = \begin{bmatrix}\mu_{f'}\\ \mu^*_{f'}\end{bmatrix} = \underline{\mathbf{K}}^{\mathrm{H}}(\mathbf{X},\mathbf{x}')\underline{\mathbf{C}}^{-1}\underline{\mathbf{y}}, \tag{9}$$

$$\underline{\mathbf{\Sigma}}_{f'} = \begin{bmatrix}\sigma_{f'} & \tilde{\sigma}_{f'}\\ \tilde{\sigma}^*_{f'} & \sigma_{f'}\end{bmatrix} = \underline{\mathbf{K}}(\mathbf{x}',\mathbf{x}') - \underline{\mathbf{K}}^{\mathrm{H}}(\mathbf{X},\mathbf{x}')\underline{\mathbf{C}}^{-1}\underline{\mathbf{K}}(\mathbf{X},\mathbf{x}'), \tag{10}$$

and

$$\underline{\mathbf{K}}(\mathbf{x}',\mathbf{x}') = \begin{bmatrix}k(\mathbf{x}',\mathbf{x}') & \tilde{k}(\mathbf{x}',\mathbf{x}')\\ \tilde{k}^*(\mathbf{x}',\mathbf{x}') & k^*(\mathbf{x}',\mathbf{x}')\end{bmatrix}, \tag{11}$$

$$\underline{\mathbf{K}}(\mathbf{X},\mathbf{x}') = \begin{bmatrix}\mathbf{k}(\mathbf{X},\mathbf{x}') & \tilde{\mathbf{k}}(\mathbf{X},\mathbf{x}')\\ \tilde{\mathbf{k}}^*(\mathbf{X},\mathbf{x}') & \mathbf{k}^*(\mathbf{X},\mathbf{x}')\end{bmatrix}. \tag{12}$$

$\underline{\mathbf{C}}$ is the augmented covariance matrix of the augmented observations $\underline{\mathbf{y}} = [\mathbf{y}^\top \ \mathbf{y}^{\mathrm{H}}]^\top$. By using the matrix-inversion lemma

$$\underline{\mathbf{C}}^{-1} = \begin{bmatrix}\mathbf{C} & \tilde{\mathbf{C}}\\ \tilde{\mathbf{C}}^* & \mathbf{C}^*\end{bmatrix}^{-1} = \begin{bmatrix}\mathbf{P}^{-1} & -\mathbf{C}^{-1}\tilde{\mathbf{C}}\mathbf{P}^{-*}\\ -\mathbf{C}^{-*}\tilde{\mathbf{C}}^*\mathbf{P}^{-1} & \mathbf{P}^{-*}\end{bmatrix}, \tag{13}$$

where $\mathbf{P} = \mathbf{C} - \tilde{\mathbf{C}}\mathbf{C}^{-*}\tilde{\mathbf{C}}^*$. Therefore, the mean, covariance and pseudo-covariance of the prediction yield, respectively,

$$\begin{aligned}\mu_{f'} &= \left[\mathbf{k}^{\mathrm{H}}(\mathbf{X},\mathbf{x}') - \tilde{\mathbf{k}}^\top(\mathbf{X},\mathbf{x}')\mathbf{C}^{-*}\tilde{\mathbf{C}}^*\right]\mathbf{P}^{-1}\mathbf{y}\\ &\quad + \left[\tilde{\mathbf{k}}^\top(\mathbf{X},\mathbf{x}') - \mathbf{k}^{\mathrm{H}}(\mathbf{X},\mathbf{x}')\mathbf{C}^{-1}\tilde{\mathbf{C}}\right]\mathbf{P}^{-*}\mathbf{y}^*, \qquad (14)\\ \sigma_{f'} &= k(\mathbf{x}',\mathbf{x}')\\ &\quad - \left[\mathbf{k}^{\mathrm{H}}(\mathbf{X},\mathbf{x}') - \tilde{\mathbf{k}}^\top(\mathbf{X},\mathbf{x}')\mathbf{C}^{-*}\tilde{\mathbf{C}}^*\right]\mathbf{P}^{-1}\mathbf{k}(\mathbf{X},\mathbf{x}')\\ &\quad - \left[\tilde{\mathbf{k}}^\top(\mathbf{X},\mathbf{x}') - \mathbf{k}^{\mathrm{H}}(\mathbf{X},\mathbf{x}')\mathbf{C}^{-1}\tilde{\mathbf{C}}\right]\mathbf{P}^{-*}\tilde{\mathbf{k}}^*(\mathbf{X},\mathbf{x}'), \qquad (15)\\ \tilde{\sigma}_{f'} &= \tilde{k}(\mathbf{x}',\mathbf{x}')\\ &\quad - \left[\mathbf{k}^{\mathrm{H}}(\mathbf{X},\mathbf{x}') - \tilde{\mathbf{k}}^\top(\mathbf{X},\mathbf{x}')\mathbf{C}^{-*}\tilde{\mathbf{C}}^*\right]\mathbf{P}^{-1}\tilde{\mathbf{k}}(\mathbf{X},\mathbf{x}')\\ &\quad - \left[\tilde{\mathbf{k}}^\top(\mathbf{X},\mathbf{x}') - \mathbf{k}^{\mathrm{H}}(\mathbf{X},\mathbf{x}')\mathbf{C}^{-1}\tilde{\mathbf{C}}\right]\mathbf{P}^{-*}\mathbf{k}^*(\mathbf{X},\mathbf{x}'). \qquad (16)\end{aligned}$$

C. Proper Complex GPR

When the pseudo-covariance matrix cancels, a complex Gaussian random vector is regarded as *proper* [25], [32]. In the zero-mean proper case, the prior (3) simplifies to

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{0},\mathbf{K},\mathbf{0}) = \frac{1}{\pi^n \det\mathbf{K}}\exp\left(-\mathbf{f}^{\mathrm{H}}\mathbf{K}^{-1}\mathbf{f}\right), \tag{17}$$

and the marginal likelihood (5) to $p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{0},\mathbf{C},\mathbf{0})$, so that $\mathbf{y}$ is also proper Gaussian. Furthermore, $\mathbf{y}$ and $\mathbf{f}$ are cross-proper, i.e., the complementary cross-covariance matrix $\mathbb{E}[\mathbf{y}\mathbf{f}^\top] = \mathbf{0}$. Hence, $\mathbf{y}$ and $\mathbf{f}$ are jointly proper [1], i.e., the composite complex random vector $[\mathbf{y}^\top,\mathbf{f}^\top]^\top$ is proper Gaussian. Now, given a test input vector $\mathbf{x}'$, the joint distribution of the training outputs $\mathbf{y}$ and f' is:

$$\begin{bmatrix}\mathbf{y}\\ f'\end{bmatrix} \sim \mathcal{N}(\mathbf{0},\mathbf{\Lambda},\mathbf{0}). \tag{18}$$

The estimated probabilistic output is the conditional distribution of f' given $\mathbf{y}$:

$$p(f'|\mathbf{x}',\mathbf{X},\mathbf{y}) = \mathcal{N}(\mu_{f'},\sigma_{f'},\mathbf{0}), \tag{19}$$

which is the conditional proper complex Gaussian distribution described with the following mean vector and covariance matrix

$$\mu_{f'} = \mathbf{k}^{\mathrm{H}}(\mathbf{X},\mathbf{x}')\mathbf{C}^{-1}\mathbf{y}, \tag{20}$$

$$\sigma_{f'} = k(\mathbf{x}',\mathbf{x}') - \mathbf{k}^{\mathrm{H}}(\mathbf{X},\mathbf{x}')\mathbf{C}^{-1}\mathbf{k}(\mathbf{X},\mathbf{x}'). \tag{21}$$

Notice that this result is the straightforward adaptation of the real-valued GPR to complex-valued signals, where transpose is changed to Hermitan transpose.

III. Complex Covariance Functions

Under the GPR point of view, the covariance function should measure *similarity* between inputs [16], [23]. A usual option is to consider that training points that are near to a test point are informative about the prediction at that point. In other kernels, e.g., polynomial kernels, similarity is measured in a different way. Covariance matrices should also be semi-definite positive. We next develop these issues for the complex-valued case, where we have the covariance and pseudo-covariance matrices. Given a zero-mean complex Gaussian vector $\mathbf{f} = \mathbf{f}_{\mathrm{r}} + \mathrm{j}\mathbf{f}_{\mathrm{j}}$, with $\mathbf{f}_{\mathrm{r}}$ its real part and $\mathbf{f}_{\mathrm{j}}$ its imaginary part:

$$\mathbf{K} = \mathbb{E}\left[\mathbf{f}\mathbf{f}^{\mathrm{H}}\right] = \mathbf{K}_{\mathrm{rr}} + \mathbf{K}_{\mathrm{jj}} + \mathrm{j}\left(\mathbf{K}_{\mathrm{jr}} - \mathbf{K}_{\mathrm{rj}}\right), \tag{22}$$

$$\tilde{\mathbf{K}} = \mathbb{E}\left[\mathbf{f}\mathbf{f}^\top\right] = \mathbf{K}_{\mathrm{rr}} - \mathbf{K}_{\mathrm{jj}} + \mathrm{j}\left(\mathbf{K}_{\mathrm{jr}} + \mathbf{K}_{\mathrm{rj}}\right), \tag{23}$$

where $\mathbf{K}_{\mathrm{rr}} = \mathbb{E}[\mathbf{f}_{\mathrm{r}}\mathbf{f}_{\mathrm{r}}^\top] \in \mathbb{R}_+^{n\times n}$ and $\mathbf{K}_{\mathrm{jj}} = \mathbb{E}[\mathbf{f}_{\mathrm{j}}\mathbf{f}_{\mathrm{j}}^\top] \in \mathbb{R}_+^{n\times n}$ are the covariance matrices of the real and imaginary parts of $\mathbf{f}$, respectively, and $\mathbf{K}_{\mathrm{rj}} = \mathbb{E}[\mathbf{f}_{\mathrm{r}}\mathbf{f}_{\mathrm{j}}^\top] = \mathbf{K}_{\mathrm{jr}}^\top \in \mathbb{R}^{n\times n}$ is the cross-covariance matrix of the real and imaginary parts. Matrix $\mathbf{K}$ must be Hermitian positive semidefinite while $\tilde{\mathbf{K}}$ must be symmetric. From the augmented point of view, the Schur complement of the augmented covariance matrix $\underline{\mathbf{K}}$ must be positive semidefinite [1].

In the design of these matrices we may proceed as follows. On the one hand, we can directly construct complex-valued functions that produce matrices as (22) and (23) with the properties described above, i.e., $\mathbf{K}$ must be Hermitian positive semidefinite and $\tilde{\mathbf{K}}$ must be symmetric. Those complex-valued functions should be carefully selected in order to fairly represent the covariance and pseudo-covariance properties of the complex-valued process being modeled. On the other hand, we may try to design their real and imaginary parts. In this second method, we can resort to three real functions $k_{\mathrm{rr}}(\mathbf{x},\mathbf{x}')$, $k_{\mathrm{jj}}(\mathbf{x},\mathbf{x}')$ and $k_{\mathrm{rj}}(\mathbf{x}',\mathbf{x})$ of the complex inputs $\mathbf{x}$, that are used to write out the three real matrices $\mathbf{K}_{\mathrm{rr}}$, $\mathbf{K}_{\mathrm{jj}}$ and $\mathbf{K}_{\mathrm{jr}} = \mathbf{K}_{\mathrm{rj}}^\top$. Again, the resulting covariance matrix $\mathbf{K}$ must be Hermitian positive semidefinite and $\tilde{\mathbf{K}}$ must be symmetric, and meeting these conditions from the design of their parts is not straightforward. However, this second option provides one important advantage: known correlation properties of the real and imaginary parts can be translated directly into the covariance and pseudo-covariance functions.

One important example is when it is known that the real and imaginary parts are uncorrelated and have null cross-covariance matrix. In such a case we should set $k_{\text{rj}}(\mathbf{x}', \mathbf{x}) = 0$ and the covariance and pseudo-covariance functions yield real functions. Also, any information about stationarity, periodicity, etc. of the real part can be modeled in $k_{\text{rr}}(\mathbf{x}, \mathbf{x}')$, and the same can be said about the imaginary part and $k_{\text{jj}}(\mathbf{x}, \mathbf{x}')$. Furthermore, in the particular case of a proper complex Gaussian vector the pseudo-covariance matrix (23) nulls and we can resort to *proper* complex GPR. Hence, $\mathbf{K}_{\text{rr}} = \mathbf{K}_{\text{jj}}$ and $\mathbf{K}_{\text{jr}} = \mathbf{K}_{\text{rj}}^\top = -\mathbf{K}_{\text{rj}}$, i.e., $\mathbf{K}_{\text{rj}}$ is a skew-symmetric cross-covariance matrix. In this case, the covariance matrix (22) simplifies to $\mathbf{K} = 2\mathbf{K}_{\text{rr}} - 2\text{j}\mathbf{K}_{\text{rj}}$, and the following properties for the three proposed real functions hold: $k_{\text{rr}}(\mathbf{x}, \mathbf{x}') = k_{\text{jj}}(\mathbf{x}, \mathbf{x}')$ and $k_{\text{rj}}(\mathbf{x}, \mathbf{x}') = -k_{\text{rj}}(\mathbf{x}', \mathbf{x})$. Also, as the covariance matrix must be Hermitian positive semi-definite, it follows that $k_{\text{rr}}(\mathbf{x}, \mathbf{x}')$ and $k_{\text{rj}}(\mathbf{x}', \mathbf{x})$ are interrelated.

Finally, the way that similarity is measured in the covariance and pseudo-covariance functions is another important issue to take into account when selecting them for complex-valued GPR and the similarity must be measured in the complex field.

A. Examples of complex-valued kernels and covariances functions

We first recall some examples of complex-valued kernels functions found in the literature.

1) Complex-valued Gaussian kernel: The first example is the complex-valued Gaussian kernel [6], [11]. It is an extension of the real Gaussian kernel defined as

$$k_{\mathbb{C}}(\mathbf{x}, \mathbf{x}') = \exp\left(-(\mathbf{x} - \mathbf{x}'^*)^\top (\mathbf{x} - \mathbf{x}'^*)/\gamma\right), \tag{24}$$

with kernel hyperparameter γ. If we separate the real and imaginary parts of the kernel

$$\begin{aligned} k_{\mathbb{C}}(\mathbf{x}, \mathbf{x}') = &\exp\left(-|\mathbf{x}_{\text{r}} - \mathbf{x}'_{\text{r}}|^2/\gamma\right) \exp\left(|\mathbf{x}_{\text{j}} + \mathbf{x}'_{\text{j}}|^2/\gamma\right) \\ &\cdot \left(\cos(2(\mathbf{x}_{\text{r}} - \mathbf{x}'_{\text{r}})^\top (\mathbf{x}_{\text{j}} + \mathbf{x}'_{\text{j}})/\gamma)\right. \\ &\left. -\text{j}\sin(2(\mathbf{x}_{\text{r}} - \mathbf{x}'_{\text{r}})^\top (\mathbf{x}_{\text{j}} + \mathbf{x}'_{\text{j}})/\gamma)\right), \end{aligned} \tag{25}$$

where $|\cdot|$ is the ℓ^2-norm. Note that this kernel gives rise to a covariance matrix with skew-symmetric cross-covariance matrix $\mathbf{K}_{\text{rj}}$. Hence, it fits in the proper case with a null pseudo-covariance. This kernel does not provide its maximum when $\mathbf{x} = \mathbf{x}'$, but when $\mathbf{x} = \mathbf{x}'^*$, i.e., it measures similarities between the real parts of the inputs, while measures dissimilarity between imaginary ones. The value it provides when $\mathbf{x} = \mathbf{x}'$ is not constant but depends on the imaginary part of $\mathbf{x}$ as $\exp(|2\mathbf{x}_{\text{j}}|^2/\gamma)$. Also, it is not stationary, it has an oscillatory behavior and may also cause serious numerical problems in the learning algorithms, as is later discussed in the Experiments section.

2) Independent kernel: In [9] the following kernel was proposed:

$$\begin{aligned} k_{ind}(\mathbf{x}, \mathbf{x}') = &\kappa_{\mathbb{R}}\left(\mathbf{x}_{\text{r}}, \mathbf{x}'_{\text{r}}\right) + \kappa_{\mathbb{R}}\left(\mathbf{x}_{\text{j}}, \mathbf{x}'_{\text{j}}\right) \\ &+ \text{j}\left(\kappa_{\mathbb{R}}\left(\mathbf{x}_{\text{r}}, \mathbf{x}'_{\text{j}}\right) - \kappa_{\mathbb{R}}\left(\mathbf{x}_{\text{j}}, \mathbf{x}'_{\text{r}}\right)\right), \end{aligned} \tag{26}$$

where $\kappa_{\mathbb{R}}$ is a real kernel of real inputs, in particular, they propose the real Gaussian kernel. Notice that this is an example of a design using real-valued functions, $k_{\text{rr}}(\mathbf{x}, \mathbf{x}')$, $k_{\text{jj}}(\mathbf{x}, \mathbf{x}')$ and $k_{\text{rj}}(\mathbf{x}', \mathbf{x})$, but with two simplifications. First, the three functions are the same real function $\kappa_{\mathbb{R}}$. Second, the inputs of the function are not complex, but real, i.e., the real part or the imaginary part of $\mathbf{x}$. Because of this simplifications, the independent kernel provides a high value when the inputs have equal real parts, $\mathbf{x}_{\text{r}} = \mathbf{x}'_{\text{r}}$, although the imaginary parts are very different. We have the same behavior for the imaginary part. Also, note that this kernel gives rise to a covariance matrix with skew-symmetric cross-covariance matrix $\mathbf{K}_{\text{rj}}$. Hence, it also could be used in the proper case as covariance.

3) Spectral kernels: Finally, there have also been some proposals to create an imaginary part from the real part of the covariance function in kriging [12], [13] for a multiple output learning framework. However, these proposals are for stationary random fields of just real inputs, i.e., $k(\mathbf{x} - \mathbf{x}')$, with $\mathbf{x} \in \mathbb{R}^d$, and do not provide a pseudo-covariance function. In [12], they propose to obtain a covariance matrix starting from a given function. However, it is unclear what the function should be for the covariance matrix to have some given properties. In [13] the proposed covariance function exhibits a sinusoidal behavior in its real and imaginary parts that is in general not suitable for the application at hand.

B. Convolution approach

We propose to follow the convolutional approach [27], [28] as a link between the two points of view in the design of the covariance functions. The idea is to generate a complex random process as the sum of the outputs of linear filters driven by real white noises. The starting point is the selection of functions that provide the desired measures of similarity and modeling for the covariances. Those functions are used as filters to generate a random process. The calculation of the covariance and pseudo-covariance of the process provides the covariance and pseudo-covariance functions. This way we ensure that the resulting covariance and pseudo-covariance functions are valid, i.e., generate a valid hermitian $\mathbf{K}$ and a valid symmetric $\tilde{\mathbf{K}}$. Then, the design of the filters conditions the properties of the kernels and the associated similarity between pairs of inputs.

Let $U(\mathbf{x})$ be the complex process written as the output of linear filters:

$$\begin{aligned} U(\mathbf{x}) = &(h_1(\mathbf{x}) + \text{j}h_2(\mathbf{x})) \star S_{\text{r}}(\mathbf{x}) \\ &+ (h_3(\mathbf{x}) + \text{j}h_4(\mathbf{x})) \star S_{\text{j}}(\mathbf{x}) \end{aligned} \tag{27}$$

where $h_m(\mathbf{x})$, $m \in \{1, 2, 3, 4\}$ represent the filters and $\star$ denotes the convolution operation. $S_{\text{r}}(\mathbf{x})$ and $S_{\text{j}}(\mathbf{x})$ are independent real white noises with zero mean and unit variance. The inputs are complex-valued vectors $\mathbf{x} \in \mathbb{C}^d$. The covariance of $U(\mathbf{x})$ for two different inputs is used as covariance function, $k(\mathbf{x}, \mathbf{x}') = \mathbf{C}_U(\mathbf{x}, \mathbf{x}') = \mathbb{E}[U(\mathbf{x})U^*(\mathbf{x}')]$, and the pseudo-covariance is used as pseudo-covariance function, $\tilde{k}(\mathbf{x}, \mathbf{x}') = \tilde{\mathbf{C}}_U(\mathbf{x}, \mathbf{x}') = \mathbb{E}[U(\mathbf{x})U(\mathbf{x}')]$. Details of the calculations of $\mathbf{C}_U(\mathbf{x}, \mathbf{x}')$ and $\tilde{\mathbf{C}}_U(\mathbf{x}, \mathbf{x}')$ can be found in Appendix A. The complexity here lies in the selection of the filters, $h_m(\mathbf{x})$. They are designed to model the system under study. In the following we propose as measure of similarity the inner product $\mathbf{d}_{\mathbf{x}}^{\text{H}}\mathbf{d}_{\mathbf{x}}$

of the difference between complex-valued inputs $\mathbf{d_x} = \mathbf{x} - \mathbf{x}'$, and use parameterized exponential filters to yield an isotropic and time invariant covariance function. Some examples are provided in the following subsection. This procedure can be applied with other types of filters to model different properties: periodicity, other measures of similarity between the inputs, etc.

1) General case: As the first example, we propose to generate the stationary process using filters $h_1(\mathbf{x}) = h_3(\mathbf{x}) = v_r \exp\left(-\mathbf{x}^{\mathrm{H}}\mathbf{x}/\gamma_r\right)$ and $h_2(\mathbf{x}) = h_4(\mathbf{x}) = v_j \exp\left(-\mathbf{x}^{\mathrm{H}}\mathbf{x}/\gamma_j\right)$, where v_r, γ_r, v_j and γ_j are filter parameters, and the inputs are $\mathbf{x} \in \mathbb{C}^d$. The covariance and pseudo-covariance of the process yield the following functions:

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= 2v_r^2 \left(\frac{\pi\gamma_r}{2}\right)^d \exp\left(-\frac{\mathbf{d}_{\mathbf{x}}^{\mathrm{H}}\mathbf{d}_{\mathbf{x}}}{2\gamma_r}\right) \\ &\quad + 2v_j^2 \left(\frac{\pi\gamma_j}{2}\right)^d \exp\left(-\frac{\mathbf{d}_{\mathbf{x}}^{\mathrm{H}}\mathbf{d}_{\mathbf{x}}}{2\gamma_j}\right), \end{aligned} \tag{28}$$

$$\begin{aligned} \tilde{k}(\mathbf{x}, \mathbf{x}') &= 2v_r^2 \left(\frac{\pi\gamma_r}{2}\right)^d \exp\left(-\frac{\mathbf{d}_{\mathbf{x}}^{\mathrm{H}}\mathbf{d}_{\mathbf{x}}}{2\gamma_r}\right) \\ &\quad - 2v_j^2 \left(\frac{\pi\gamma_j}{2}\right)^d \exp\left(-\frac{\mathbf{d}_{\mathbf{x}}^{\mathrm{H}}\mathbf{d}_{\mathbf{x}}}{2\gamma_j}\right) \\ &\quad + 4\mathrm{j}v_r v_j \left(\frac{\pi\gamma_r\gamma_j}{\gamma_r + \gamma_j}\right)^d \exp\left(-\frac{\mathbf{d}_{\mathbf{x}}^{\mathrm{H}}\mathbf{d}_{\mathbf{x}}}{\gamma_r + \gamma_j}\right). \end{aligned} \tag{29}$$

The three real functions $k_{\mathrm{rr}}(\mathbf{x}, \mathbf{x}')$, $k_{\mathrm{jj}}(\mathbf{x}, \mathbf{x}')$ and $k_{\mathrm{rj}}(\mathbf{x}', \mathbf{x})$ of the complex inputs $\mathbf{x} \in \mathbb{C}^d$ are easily identified in this example. Note that the covariance (28) is real-valued while the pseudo-covariance (29) is complex-valued. This is due to the fact that $\mathbf{K}_{\mathrm{jr}} = \mathbf{K}_{\mathrm{rj}}^{\top} = \mathbf{K}_{\mathrm{rj}}$ for the second order stationary process generated with the filters.

2) Independent real and imaginary parts: In this scenario the cross-covariance between real and imaginary parts is null, $\mathbf{K}_{\mathrm{rj}} = \mathbf{0}$, and we should set $k_{\mathrm{rj}}(\mathbf{x}', \mathbf{x}) = 0$. We can use the same filters proposed for the general case but with the following change of sign: $h_4(\mathbf{x}) = -h_2(\mathbf{x})$. Therefore, the covariance function remains as in (28), while the pseudo-covariance function is as in (29) but with the imaginary part equal to zero.

3) Proper case with $k_{rj}(\mathbf{x}', \mathbf{x}) = 0$: In the proper case $\tilde{k}(\mathbf{x}, \mathbf{x}') = 0$ and if in addition $k_{\mathrm{rj}}(\mathbf{x}', \mathbf{x}) = 0$, we use the function in (28) that yields a simple real covariance function:

$$k(\mathbf{x}, \mathbf{x}') = v \exp\left(-\mathbf{d}_{\mathbf{x}}^{\mathrm{H}}\mathbf{d}_{\mathbf{x}}/\gamma\right). \tag{30}$$

4) Proper case with $k_{rj}(\mathbf{x}', \mathbf{x}) \neq 0$: This scenario arises when $\mathbf{K}_{\mathrm{jr}}$ is skew-symmetric. The kernel functions in (24) and (26) could be used in this case. However, the first one involves some quite particular similarity properties with exponential growth for some pair of points. The second assumes constant similarity for distant points as long as they have same real and imaginary parts. We develop a complex-valued function k to model a correlation between the real part of the process and a displaced or translated imaginary part, with displacement given by $\boldsymbol{\mu} \in \mathbb{C}^d, \boldsymbol{\mu} \neq \mathbf{0}$. $\mathbf{K}_{\mathrm{jr}}$ is skew-symmetric if there is also a correlation between the real part and a displaced imaginary part when the displacement is given by $-\boldsymbol{\mu}$, and this correlation has the same value with opposite sign. To model this behavior, we propose now filters $h_1(\mathbf{x}) = h_3(\mathbf{x}) = v_r \exp\left(-\mathbf{d}_{\mathbf{x}}^{\mathrm{H}}\mathbf{d}_{\mathbf{x}}/\gamma\right)$, $h_2(\mathbf{x}) = v_j \exp\left(-(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{H}}(\mathbf{x} - \boldsymbol{\mu})/\gamma\right)$, and $h_4(\mathbf{x}) = -v_j \exp\left(-(\mathbf{x} + \boldsymbol{\mu})^{\mathrm{H}}(\mathbf{x} + \boldsymbol{\mu})/\gamma\right)$. The covariance function yields

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= v_A \exp\left(-\frac{\mathbf{d}_{\mathbf{x}}^{\mathrm{H}}\mathbf{d}_{\mathbf{x}}}{2\gamma}\right) \\ &\quad + \mathrm{j}v_B \left(\exp\left(-\frac{(\mathbf{d}_{\mathbf{x}} - \boldsymbol{\mu})^{\mathrm{H}}(\mathbf{d}_{\mathbf{x}} - \boldsymbol{\mu})}{2\gamma}\right)\right. \\ &\quad \left. - \exp\left(-\frac{(\mathbf{d}_{\mathbf{x}} + \boldsymbol{\mu})^{\mathrm{H}}(\mathbf{d}_{\mathbf{x}} + \boldsymbol{\mu})}{2\gamma}\right)\right), \end{aligned} \tag{31}$$

where $v_A = 2\left(v_{\mathrm{r}}^2 + v_{\mathrm{j}}^2\right)\left(\frac{\pi\gamma}{2}\right)^d$ and $v_B = 2v_{\mathrm{r}}v_{\mathrm{j}}\left(\frac{\pi\gamma}{2}\right)^d$. The pseudo-covariance yields

$$\tilde{k}(x, x') = 2\left(v_{\mathrm{r}}^2 - v_{\mathrm{j}}^2\right)\left(\frac{\pi\gamma}{2}\right)^d \exp\left(-\frac{\mathbf{d}_{\mathbf{x}}^{\mathrm{H}}\mathbf{d}_{\mathbf{x}}}{2\gamma}\right). \tag{32}$$

By setting $v_{\mathrm{r}} = v_{\mathrm{j}}$ we get the proper case.

IV. Hyperparameters Estimation

A major advantage of GPR is that the hyperparameters can also be estimated by maximizing the marginal likelihood [16]. From the marginal likelihood (5) we can compute the log marginal likelihood

$$L(\theta) = \log p\left(\mathbf{y}|\mathbf{X}\right) = -\frac{1}{2}\underline{\mathbf{y}}^{\mathrm{H}}\underline{\mathbf{C}}^{-1}\underline{\mathbf{y}} - \frac{1}{2}\log\det\underline{\mathbf{C}} - n\log\pi. \tag{33}$$

The augmented covariance matrix $\underline{\mathbf{C}} = \underline{\mathbf{C}}(\boldsymbol{\theta})$ can be parameterized in terms of the hyperparameters θ_i, which are the parameters of the covariance and pseudo-covariance functions and the noise variance and pseudo-covariance. $L\left(\boldsymbol{\theta}\right)$ is a real function of a complex-valued Hermitian matrix. Therefore, in the maximization of (33), we must seek generalized complex-valued matrix derivatives [26], [33]. The result for the gradient, as developed in Appendix B, is as follows,

$$\frac{\partial L}{\partial \theta_i} = \mathrm{Tr}\left(\left(\underline{\mathbf{C}}^{-1}\underline{\mathbf{y}}\,\underline{\mathbf{y}}^{\mathrm{H}}\underline{\mathbf{C}}^{-1} - \underline{\mathbf{C}}^{-1}\right)\frac{\partial \underline{\mathbf{C}}}{\partial \theta_i}\right). \tag{34}$$

In the proper case, when $\tilde{\mathbf{C}} = \mathbf{0}$, the gradient simplifies to

$$\frac{\partial L}{\partial \theta_i} = 2\mathrm{Tr}\left(\left(\mathbf{C}^{-1}\mathbf{y}\mathbf{y}^{\mathrm{H}}\mathbf{C}^{-1} - \mathbf{C}^{-1}\right)\frac{\partial \mathbf{C}}{\partial \theta_i}\right). \tag{35}$$

V. Experiments

We include three experiments. First, we evaluate the full CGPR solution against the simpler proper CGPR. Then we illustrate the performance of the proper CGPR in an scenario where the covariance is complex-valued. Finally, we face the equalization of nonlinear channels to compare to previous solutions. In this last experiment, we use the recursive version of the proper CGPR.

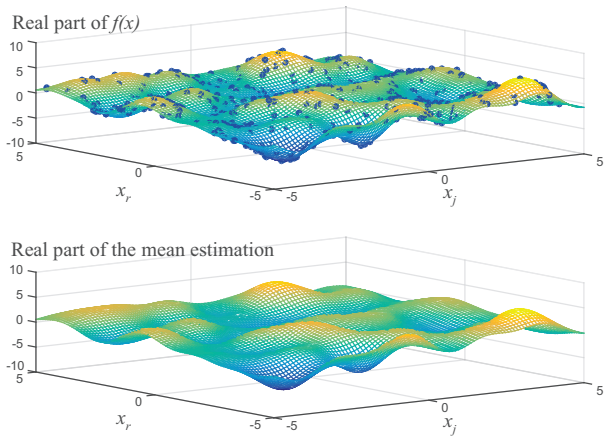

Fig. 1: Real part of the sample function $f(x)$ of the process (top) and real part of the mean estimation (14) (bottom) versus the real and imaginary parts of the input, x_r and x_j. The training samples are depicted as blue circles.

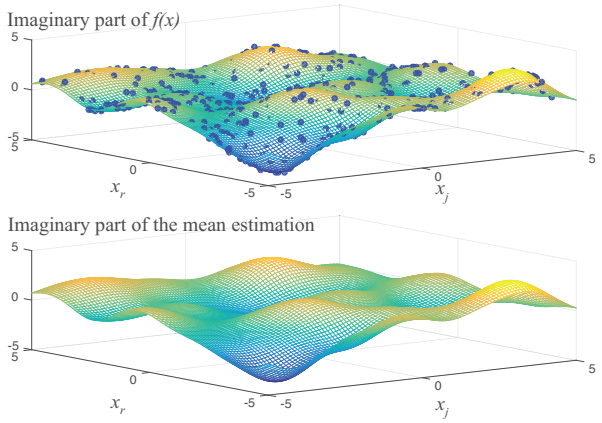

Fig. 2: Imaginary part of the sample function $f(x)$ of the process (top) and real part of the mean estimation (14) (bottom) versus the real and imaginary parts of the input, x_r and x_j. The training samples are depicted as blue circles.

A. *Full CGPR*

We generate a sample function of a non-proper complex-valued Gaussian process as described in Subsection III-B-1). The inputs in this experiment are complex-valued scalars, i.e., $\mathbf{x} = x \in \mathbb{C}$, to easily represent the sample function of the process in a figure. We have set $h_1(x) = h_3(x) = 0.1\exp(-x^*x/0.6)$ and $h_2(x) = h_4(x) = 0.05\exp(-x^*x/1.5)$. The covariance function is given by (28), $k(x,x') = 0.006\pi\exp(-(d_{\mathbf{x}}^* d_{\mathbf{x}})/1.2) + 0.0038\pi\exp(-(d_{\mathbf{x}}^* d_{\mathbf{x}})/3)$, while the pseudo-covariance function is as in (29), $\tilde{k}(x,x') = 0.006\pi\exp(-(d_{\mathbf{x}}^* d_{\mathbf{x}})/1.2) - 0.0038\pi\exp(-(d_{\mathbf{x}}^* d_{\mathbf{x}})/3) + \mathrm{j}0.0086\pi\exp(-(d_{\mathbf{x}}^* d_{\mathbf{x}})/2.1)$. As an example, the real part of a sample function obtained, $f(x)$, is shown in Fig. 1. The imaginary part is shown in Fig. 2. Gaussian noise with variance σ_ϵ^2 and pseudo-variance $\rho\sigma_\epsilon^2$ is added to represent measurement uncertainty, where $\rho = 0.8\exp(\mathrm{j}3\pi/2)$ and σ_ϵ^2 is set to be 25 dB below the variance of the sample function (signal-to-noise ratio, SNR = 25 dB). Then, we randomly choose $n = 500$ noisy training samples and learn the sample function of the process by using the predictive mean (14), variance (15) and pseudo-variance (16). The training samples used are marked as circles in Figs. 1 and 2. The real part of the predictive mean is shown in Fig. 1 (bottom) while the imaginary part is shown in Fig. 2 (bottom). The mean squared error (MSE) of the estimation is $10\log_{10}(\text{MSE}) = -8.2$ dB, computed for 10^4 inputs in this example.

The predictive capability of the complex GPR in (14) is compared with that of the proper case in (20). The mean squared error (MSE) of the estimation for the proper case is $10\log_{10}(\text{MSE}) = -4.67$ dB in this example. We show in Figs. 3 and 4 randomly chosen slices of the sample function in Figs. 1 and 2. The imaginary part of the input was fixed to $x_j = -0.1515$. In Fig. 3 we include the real part of the prediction in (14) and the grey shaded area that represents the point-wise mean plus and minus two times the standard deviation. The mean of the prediction for the proper case

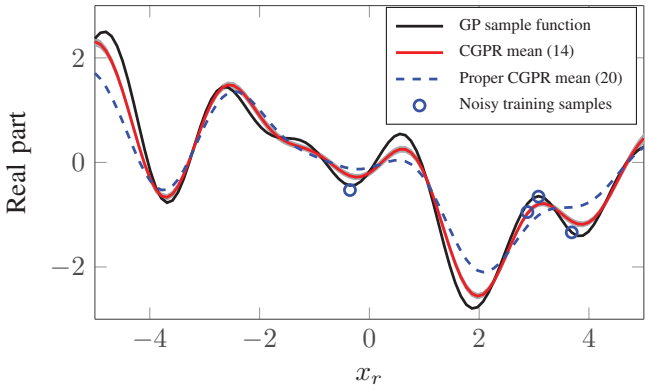

Fig. 3: Real parts of the sample function of the process $f(\mathbf{x})$, the predictive CGPR mean (14), and the predictive mean for the proper CGPR case (20), versus the real part of the input x_r, for $x_j = -0.1515$.

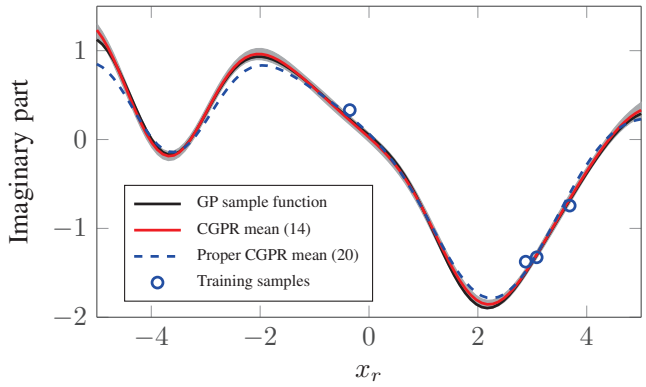

Fig. 4: Imaginary parts of the sample function of the process $f(\mathbf{x})$, the predictive CGPR mean (14), and the predictive mean for the proper CGPR case (20), versus the real part of the input x_r, for $x_j = -0.1515$.

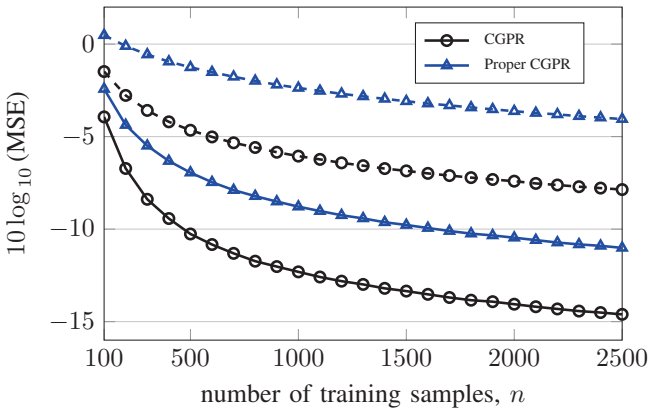

Fig. 5: Averaged $10\log_{10}$(MSE) versus the number of training samples for the predictive CGPR mean (14) and the proper CGPR case (20). Solid line: SNR = 25 dB. Dashed line: SNR = 10 dB.

in (20) is plotted as a dashed line. In Fig. 4 we include the imaginary part. The general CGPR prediction is always closer to the actual value of $f(x)$ than the prediction for the proper case, as expected, since the general CGPR also uses the information of the pseudo-covariance. This prediction improvement is highlighted in Fig. 5, where we compare the mean squared error for both estimations along the number of training samples. A higher noise case (SNR = 10 dB) is also included. Results are the average of 100 simulated trials for each case in this example. The proposed complex GPR performs better than the proper CGPR, with a remarkable reduction in the number of training samples. We achieve the same MSE of -10 dB with a sizable reduction in the number of training examples, i.e. from 1500 to 500 for an SNR of 25 dB.

B. Proper CGPR

To illustrate the performance of the hyperparameter estimation we face the learning of a proper complex Gaussian process with complex-valued covariance function. In this scenario $\mathbf{K}_{\text{jr}}$ is skew-symmetric. Again, the inputs in this experiment are complex-valued scalar, i.e., $\mathbf{x} = x \in \mathbb{C}$, for representation purposes. We use the covariance function in (31), with $v_A = 2$, $v_B = 1$, $\gamma = 1.125$ and $\mu = 2 + 2\text{j}$, to generate a sample function of the process, $f(x)$. The real part of the sample function is shown in Fig. 6, while the imaginary part is depicted in Fig. 7. Circular complex Gaussian noise with $\sigma_\epsilon = 0.1$ is added to represent measurement uncertainty and $n = 200$ training noisy samples are randomly chosen as training data. The maximization of the log marginal likelihood in (33) using (35) yields the following estimated values of the hyperparameters: $\hat{v}_A = 2.1169$, $\hat{v}_B = 1.1425$, $\hat{\gamma} = 1.1373$, $\hat{\mu} = 1.9371 + \text{j}1.9983$, and $\hat{\sigma}_\epsilon = 0.0968$. Then, the mean (20) and variance (21) of the predictive distribution are found using the training samples and the estimated values of the hyperparameters. The real part of the predictive mean (20) is depicted in Fig. 6 (bottom), while the imaginary part is depicted in Fig. 7 (bottom). The MSE of the estimation is

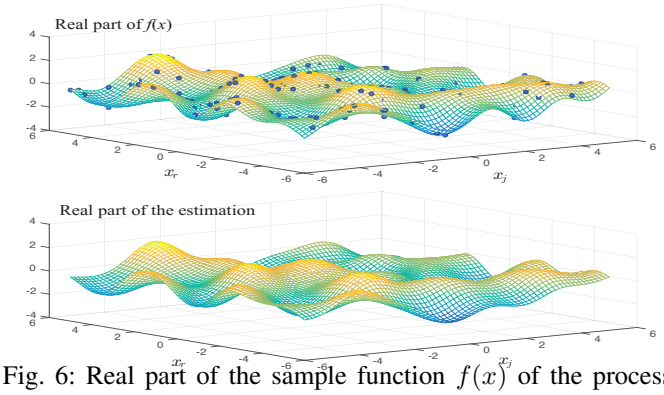

Fig. 6: Real part of the sample function $f(x)$ of the process (top) and real part of the mean estimation (20) (bottom) versus the real and imaginary parts of the input. The training samples are depicted as blue circles.

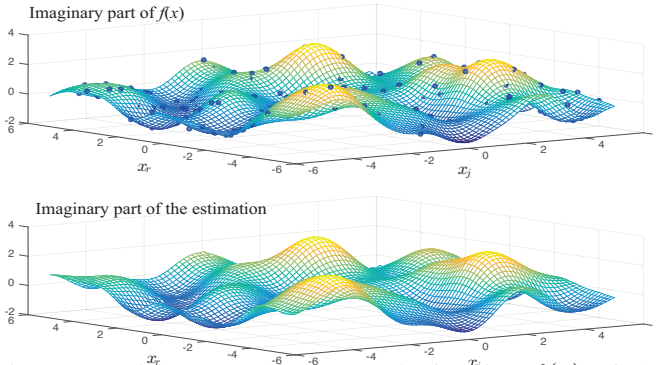

Fig. 7: Imaginary part of the sample function $f(x)$ of the process (top) and imaginary part of the mean estimation (20) (bottom) versus the real and imaginary parts of the input. The training samples are depicted as blue circles.

-13.8807 dB. We include in Figs. 8 and 9 randomly chosen slices of the sample function. Fig. 8 shows the real part of the sample function and the real part of the prediction (20) versus the real part of the input, for $x_\text{j} = 3.4684$. Fig. 9 shows the imaginary part of the sample function, and the imaginary part of the prediction (20) versus the real part of the input, for $x_\text{j} = -5.4430$. The training samples are depicted as blue circles. Also, four instances of the posterior are plotted in both Figs. 8 and 9.

To complete the analysis we show in Figs. 10 and 11 the MSE of the estimation for each hyperparameter by maximizing the log marginal likelihood in (33) using (35) under different settings. Fig. 10 shows the MSE versus the SNR for a fixed number of training samples $n = 200$, while Fig. 11 shows the MSE versus the number of training samples for a fixed SNR of 16 dB. The MSE is the averaged value for 100 independent trials. Finally, we compare in Fig. 12 the MSE of the learning when these estimated hyperparameters are used to calculate the prediction (20) with the prediction calculated with the true hyperparameters. In Fig. 12 (top) the number of training samples was fixed to $n = 200$, while in Fig. 12 (bottom) the SNR was fixed to 16 dB. The MSE curves are very close in both cases.

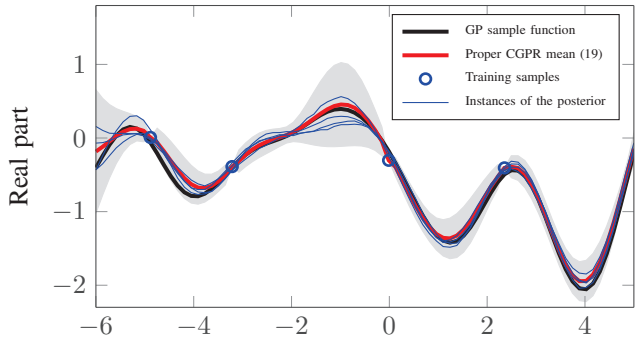

Fig. 8: Real parts of the output and the predictive mean (20) versus the real part of the input x_r for $x_\mathrm{j} = 3.4684$. Training samples are depicted as blue circles. Four instances of the posterior are also plotted.

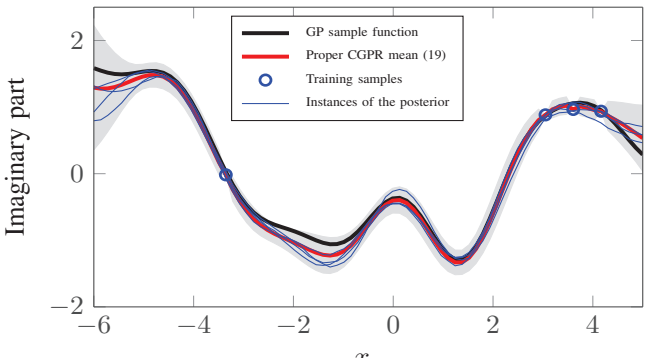

Fig. 9: Imaginary parts of the output and the predictive mean (20) versus the real part of the input x_r for $x_\mathrm{j} = -5.4430$. Training samples are depicted as blue circles. Four instances of the posterior are also plotted.

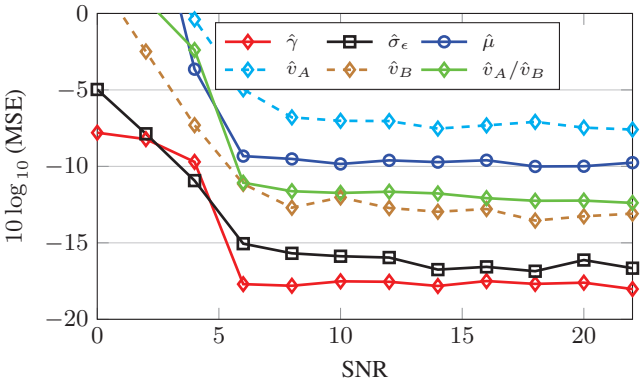

Fig. 10: Hyperparameters learning curve versus the SNR using 200 training samples.

We include in Figs. 10 and 11 the MSE of the estimation of the ratio v_A/v_B. This ratio is more important than the actual value of v_A or v_B for the estimation of the mean in (20). The ratio v_A/v_B and σ_ϵ are responsible for the amplitude accuracy of the estimation. As shown in Figs. 10 and 11 the MSE for $\hat{v}_A/\hat{v}_B$ and $\hat{\sigma}_\epsilon$ are low and, therefore, the MSE of the function estimation is low, as shown in Fig. 12.

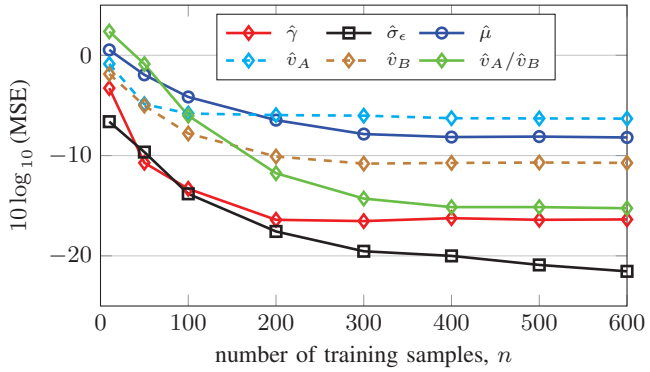

Fig. 11: Hyperparameters learning curve versus the number of training samples for $SNR = 16$.

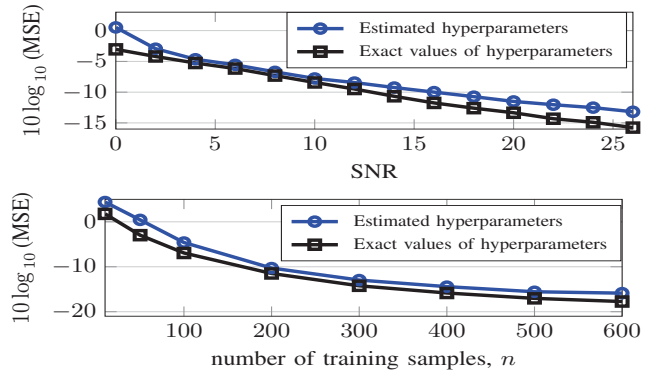

Fig. 12: Predictive MSE versus the SNR for 200 training samples (above), and Predictive MSE versus the number of training samples for $SNR = 16$ (below).

C. Nonlinear channel equalization

The performance of the proposed complex GPR is tested in the context of the nonlinear channel equalization task in [11] and [8]. Two nonlinear channels are considered. Both channel models consist of a linear filter $t(n) = (-0.9 + 0.8\mathrm{j}) \cdot s(n) + (0.6 - 0.7\mathrm{j}) \cdot s(n-1)$ and a nonlinear function. The linear filter represents a communication channel with memory, while the nonlinear function represents the effect of nonlinear circuits, such as amplifiers. The nonlinearity is $q(n) = t(n) + (0.1 + 0.15\mathrm{j}) \cdot t^2(n) + (0.06 + 0.05\mathrm{j}) \cdot t^3(n)$ for the first case (labeled as *soft nonlinear channel*), and $q(n) = t(n) + (0.2 + 0.25\mathrm{j}) \cdot t^2(n) + (0.12 + 0.09\mathrm{j}) \cdot t^3(n)$ for the second case (labeled as *strong nonlinear channel*). The input signals are $s(n) = 0.70(\sqrt{1-\rho^2}X(n) + \mathrm{j}\rho Y(n))$, and $X(n)$ and $Y(n)$ were Gaussian random variables. 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 a SNR of 16 dB, as in [11].

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 $\mathbf{x}(n) = [r(n+D), r(n+D-1), \cdots, r(n+D-L+1)]^\top$, where

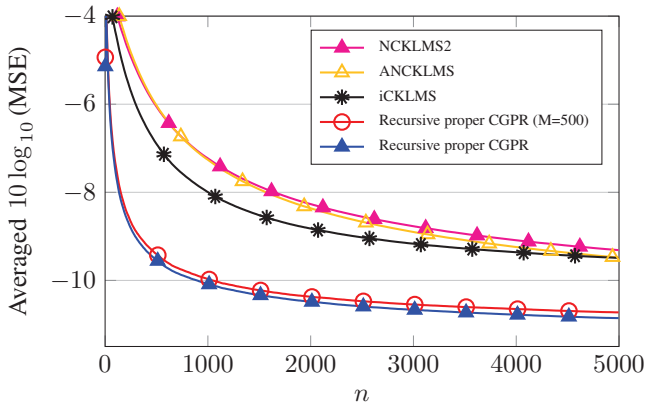

Fig. 13: Averaged MSE along n for NCKLMS2, ANCKLMS, iCKLMS, the recursive proper CGPR and the recursive proper CGPR with M=500 basis for the soft nonlinear channel equalization problem and the circular input case.

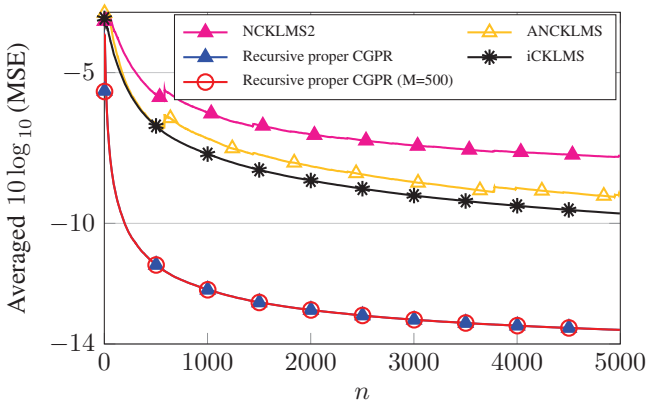

Fig. 14: Averaged MSE along n for NCKLMS2, ANCKLMS, iCKLMS, the recursive proper CGPR and the recursive proper CGPR with M=500 basis for the strong nonlinear channel equalization and the noncircular input case ($\rho = 0.1$).

$L > 0$ is the filter length and D is the equalization time delay. Experiments are conducted as in [11] and [8], where $L = 5$ 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 (*soft* and *strong*) nonlinear channels. In all cases the results are averaged over 500 trials where the input signal samples $s(n)$ and noise are randomly generated.

The performance of our proposal is compared with the NCKLMS2 algorithm in [11], the ACKLMS algorithm in [8] and the iCKLMS in [9]. Both the NCKLMS2 and ACKLMS algorithms use the complex Gaussian kernel in (24). The iCKLMS is as the NCKLMS2 algorithm but using the independent kernel (26) with $\kappa_{\mathbb{R}}$ being the real Gaussian kernel. We use the code available in [34] to run these algorithms. All the parameters required for the NCKLMS2 and the ACKLMS algorithms (γ in kernel and step update parameter) are set to the values described in [11] and [8], except for the *strong nonlinear channel* noncircular case, where in order to ensure convergence we increase γ to $\gamma = 400$ for both algorithms. For the iCKLMS, $\gamma = 25$ and the step update parameter is 1/8 (except for the *strong nonlinear channel* noncircular case, where it is reduced to 1/16), tuned for the best possible result. For the three algorithms the novelty criterion [35], [36] is used for the sparsification with $\delta_1 = 0.15$ and $\delta_2 = 0.2$, as in [34].

We design a CGPR solution as follows. The CGPR outputs here are the signals $s(n)$. Note first that the real and the imaginary parts of $s(n)$ are generated independently and therefore have null cross-covariances, $\mathbf{K}_{\text{rj}} = \mathbf{0}$. In such a case, $\mathbf{K} = \mathbf{K}_{\text{rr}} + \mathbf{K}_{\text{jj}}$ in (22) and $\tilde{\mathbf{K}} = \mathbf{K}_{\text{rr}} - \mathbf{K}_{\text{jj}}$ in (23), and it is not necessary to use complex-valued covariance functions, as it was explained in Section III. Both $\mathbf{K}_{\text{rr}}$ and $\mathbf{K}_{\text{jj}}$ can be obtained from a real kernel as in (30). Also, in this equalization application, when trying to set the values of the hyperparameters, we found that independently of the factor ρ the best solution is achieved with $\mathbf{K}_{\text{rr}} = \mathbf{K}_{\text{jj}}$. Therefore, in this scenario the general case, the full CGPR, reduces to the proper CGPR, as we can set $\tilde{\mathbf{K}} = \mathbf{K}_{\text{rr}} - \mathbf{K}_{\text{jj}} = \mathbf{0}$. Hence, the proper version of the CGPR in (20) suffices and

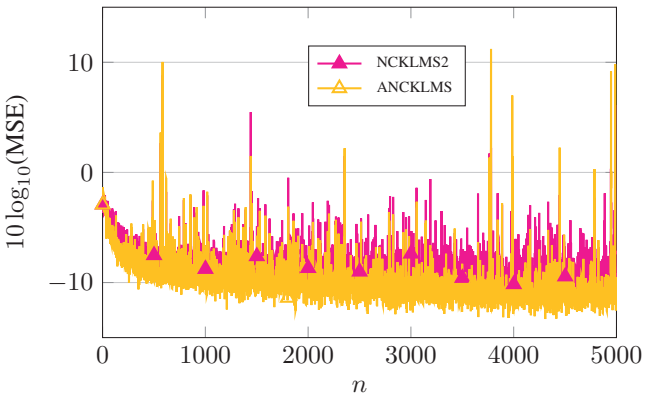

Fig. 15: MSE along n for NCKLMS2 and ANCKLMS for the strong nonlinear channel equalization problem for the noncircular input case ($\rho = 0.1$).

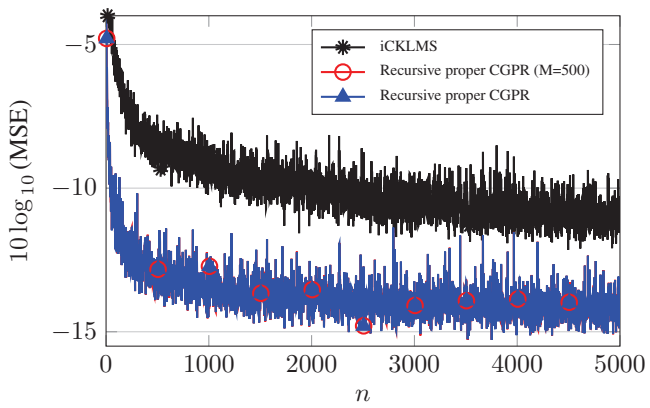

Fig. 16: MSE along n for iCKLMS, the recursive proper CGPR and the recursive proper CGPR with M=500 basis for the strong nonlinear channel equalization problem for the noncircular input case ($\rho = 0.1$).

we propose a real-valued covariance function as the one in (30). However, since both the NCKLMS2 and the ACKLMS

are online sequential algorithms we have to use an online algorithm for the proper CGPR also, in order to provide a fair comparison. In [29], the authors provide a Bayesian derivation for the kernel recursive least-squares algorithm and a criterion to remove the least relevant basis (the set of inputs at which the joint posterior is available, i.e., the training samples in our setting). Since the pseudo-covariance cancels, the method in [29] can be easily adapted to the proper CGPR case in order to yield a recursive proper CGPR. The objective is to infer the conditional distribution $p\left(\mathbf{f}_{n+1}|\mathcal{D}, \mathbf{x}', y'\right)$ of $\mathbf{f}_{n+1} = [\mathbf{f}_n^\top, f']^\top$ given the training set $\mathcal{D} = \{[\mathbf{x}_1, ..., \mathbf{x}_n], [y_1, ..., y_n]^\top\}$ and a new input $\mathbf{x}' = \mathbf{x}(n+1)$ with corresponding output $y' = y(n+1)$, where $f' = f(\mathbf{x}')$. We apply the recursive proper CGPR with basis removal criterion using $M = 500$ bases where the first 250 samples were used for the hyperparameters estimation by maximizing the log marginal likelihood in (33) using (35). As reference we also include the recursive proper CGPR solution without basis removal criterion with 1000 randomly chosen samples among the total of 5000 used to find a better estimation of the hyperparameters. Note that the number of bases used by the NCKLMS2, ACKLMS or iCKLMS algorithms with the novelty sparsification criterion grew above 2000 in these experiments, and therefore the choice of $M = 500$ bases is far below that number.

We show in Figs. 13 and 14 the averaged MSE along the input samples for the NCKLMS2, the ACKLMS, the iCKLMS and the two recursive proper CGPR algorithms (with and without basis removal criterion), for the *soft* nonlinear channel circular and the *strong* nonlinear with $\rho = 0.1$ cases. The MSE value depicted for each sample is the averaged MSE for all previous outputs, as in [11], [34]. It can be observed in the figures the remarkable good results of the recursive proper CGPR in all cases, even with only 250 bases used for the hyperparameters estimation and $M = 500$ bases used for the prediction, with the additional advantage of the estimation of the hyperparameters from the samples, avoiding cross-validation. This solution is very close to the proper CGPR approach used as reference. The raw MSE, i.e., not averaged, for the *soft* nonlinear channel circular and the *strong* nonlinear with $\rho = 0.1$ cases are shown in Figs. 15 to 18.

By using the proposed solution we avoid convergence problems found in the learning process of both the NCKLMS2 and ACKLMS algorithms. These problems can be observed if the MSE is not averaged for the previous outputs. As examples, we provide in Fig. 15 the same results that were included in Fig. 14 for the NCKLMS2 and ACKLMS algorithms, but now the MSE is not averaged. Notice that those algorithms are not able to provide a good prediction for some outputs, with MSE peak values above 5 dB. This is not the case for the iCKLMS and the recursive proper CGPR, as can be observed in Fig. 16. We believe that the NCKLMS2 or ACKLMS algorithms fail to provide a good prediction for some outputs because of the kernel they use. The independent kernel in the iCKLMS algorithm seems a better choice than the complex Gaussian kernel. However, the proper CGPR with the real kernel provides, by far, the best solution.

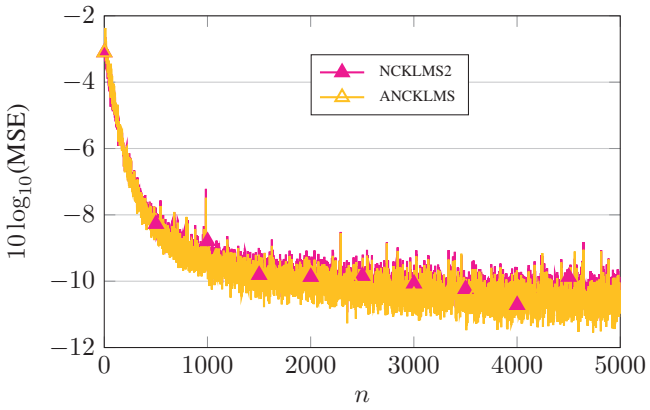

Fig. 17: MSE along n for NCKLMS2 and ANCKLMS for the soft nonlinear channel equalization problem for the circular input case.

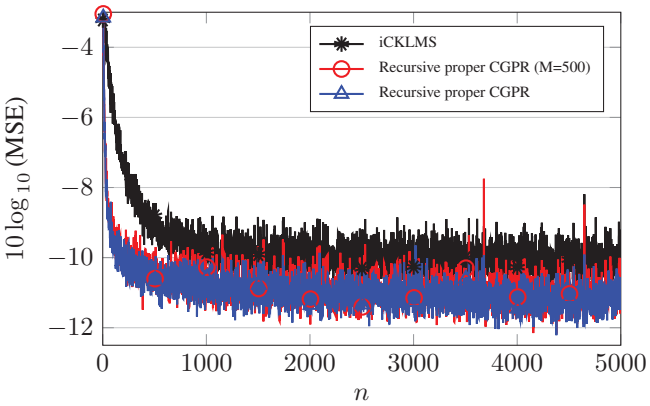

Fig. 18: MSE along n for iCKLMS, the recursive proper CGPR and the recursive proper CGPR with M=500 basis for the soft nonlinear channel equalization problem for the circular input case.

VI. Conclusions

Regression in the complex-valued case has been addressed by dealing with real and imaginary parts independently, using a straightforward extension of the real case or learning a vector with real and imaginary parts stacked. However, in these approaches the design of the kernels remains an open problem and the complex-valued formulation is lost. On the other hand, the straightforward adaptation of the real case to the complex one corresponds to the proper case, and is not able to deal with any scenario. In this paper we present a new approach based on the results for complex-valued Gaussian processes. To the best of our knowledge this is the first tool working in the complex field, suitable for any scenario.

We exploit the GPR framework to provide a full statistical description of the general complex-valued solution. We highlight the importance of the pseudo-covariance term, and the mean and covariance of the posterior are developed. Only when the pseudo-covariance cancels, the method simplifies to the proper case. We develop the optimization of the marginal likelihood to estimate the hyperparameters, by taking into

account generalized complex-valued matrix derivatives. The selection or design of the covariance function or kernel is also an important issue that we deal with in this paper. We analyze the terms in the real and imaginary parts of the covariance and pseudo-covariance functions, and their symmetries. We review some previous proposals, and the way these kernels measure similarity between the complex-valued inputs. We propose a more general method to design the covariance and pseudo-covariance functions from filters. We highlight the importance of focusing on the properties of the covariance and pseudo-covariance for the problem at hand to get the simplest solution needed. In particular, when the function we would like to fit does not have null pseudo-covariance, then the general CGPR formulation provides better results. On the other hand, if the pseudo-covariance is null, the simpler proper CGPR is enough. Two experiments are included to illustrate these facts, showing the learning of non-proper and proper models, along with the learning of the hyperparameters. Also, when the cross-covariance between the real and imaginary parts is symmetric or null, there is no need for a complex-valued covariance function. These developments are in the line of solving the equalization of nonlinear channels in the experiments section, where we propose a real-valued covariance function while previous solutions use a complex-valued one. We apply a recursive version of the proper CGPR with a basis selection criterion and compare the results to previous approaches. The recursive proper CGPR yields a remarkable reduction of the MSE, up to 4 dB, and with a number of bases that is less than 25% of the number required for previous approaches. Also, the proper CGPR approach allows us to learn the hyperparameters from the data, so there is no need to set them by extensive search or cross-validation techniques.

APPENDIX

A. Design of a Complex Covariance Function

We follow here a procedure similar to that in [27]. Consider two independent, real, Gaussian white noise processes, with zero mean and unit variance, $S_{\mathrm{r}}(\mathbf{x})$ and $S_{\mathrm{j}}(\mathbf{x})$, where $\mathbf{x} \in \mathbb{C}^d$, producing an output $U(\mathbf{x})$ defined by the sum of convolutions

$$\begin{aligned} U(\mathbf{x}) &= (h_1(\mathbf{x}) + \mathrm{j}h_2(\mathbf{x})) \star S_{\mathrm{r}}(\mathbf{x}) + (h_3(\mathbf{x}) + \mathrm{j}h_4(\mathbf{x})) \star S_{\mathrm{j}}(\mathbf{x}) \\ &= \sum_{m=1}^{4} \lambda_m h_m(\mathbf{x}) \star S_m(\mathbf{x}), \end{aligned} \tag{36}$$

where $\lambda_1 = \lambda_3 = 1$ and $\lambda_2 = \lambda_4 = \mathrm{j}$, $S_1(\mathbf{x}) = S_2(\mathbf{x}) = S_{\mathrm{r}}(\mathbf{x})$, and $S_3(\mathbf{x}) = S_4(\mathbf{x}) = S_{\mathrm{j}}(\mathbf{x})$.

The covariance of $U(\mathbf{x})$ is derived as follows:

$$\begin{aligned} \mathbf{C}_U(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left[U(\mathbf{x})U^*(\mathbf{x}')\right] \\ &= \mathbb{E}\left[\sum_{m=1}^{4} \int_{\mathbb{C}^d} \lambda_m h_m(\boldsymbol{\alpha}) S_m(\mathbf{x} - \boldsymbol{\alpha}) \mathrm{d}^d\boldsymbol{\alpha} \right. \\ &\quad \left. \cdot \sum_{n=1}^{4} \int_{\mathbb{C}^d} \lambda_n^* h_n^*(\boldsymbol{\beta}) S_n(\mathbf{x}' - \boldsymbol{\beta}) \mathrm{d}^d\boldsymbol{\beta}\right] \\ &= \sum_{m=1}^{4}\sum_{n=1}^{4} \left\{ \int_{\mathbb{C}^d} \int_{\mathbb{C}^d} \lambda_m \lambda_n^* h_m(\boldsymbol{\alpha}) h_n^*(\boldsymbol{\beta}) \right. \\ &\quad \left. \cdot \mathbb{E}\left[S_m(\mathbf{x} - \boldsymbol{\alpha}) S_n(\mathbf{x}' - \boldsymbol{\beta})\right] \mathrm{d}^d\boldsymbol{\alpha} \mathrm{d}^d\boldsymbol{\beta} \right\}. \end{aligned} \tag{37}$$

Processes $S_m(\mathbf{x} - \boldsymbol{\alpha})$ and $S_n(\mathbf{x}' - \boldsymbol{\beta})$ covary only if $m, n \in \{1, 2\}$ or $m, n \in \{3, 4\}$, and $(\mathbf{x} - \boldsymbol{\alpha}) = (\mathbf{x}' - \boldsymbol{\beta})$. In such cases, $\mathbb{E}\left[S_m(\mathbf{x} - \boldsymbol{\alpha}) S_n(\mathbf{x}' - \boldsymbol{\beta})\right] = \delta(\boldsymbol{\alpha} - (\mathbf{x} - \mathbf{x}' + \boldsymbol{\beta})) = \delta(\boldsymbol{\alpha} - (\mathbf{d}_{\mathbf{x}} + \boldsymbol{\beta}))$, where $\delta(\cdot)$ is the Dirac delta function, and the integrals in (37) yield

$$\begin{aligned} &f_{mn}(\mathbf{d}_{\mathbf{x}}) = \\ &\int_{\mathbb{C}^d} \int_{\mathbb{C}^d} \lambda_m \lambda_n^* h_m(\boldsymbol{\alpha}) h_n^*(\boldsymbol{\beta}) \delta(\boldsymbol{\alpha} - (\mathbf{d}_{\mathbf{x}} + \boldsymbol{\beta})) \mathrm{d}^d\boldsymbol{\alpha} \mathrm{d}^d\boldsymbol{\beta} \\ &= \int_{\mathbb{C}^d} \lambda_m \lambda_n^* h_m(\boldsymbol{\beta} + \mathbf{d}_{\mathbf{x}}) h_n^*(\boldsymbol{\beta}) \mathrm{d}^d\boldsymbol{\beta}. \end{aligned} \tag{38}$$

Hence,

$$\mathbf{C}_U(\mathbf{x}, \mathbf{x}') = \sum_{m=1}^{2}\sum_{n=1}^{2} f_{mn}(\mathbf{d}_{\mathbf{x}}) + \sum_{m=3}^{4}\sum_{n=3}^{4} f_{mn}(\mathbf{d}_{\mathbf{x}}). \tag{39}$$

The pseudo-covariance of $U(\mathbf{x})$, $\tilde{\mathbf{C}}_U(\mathbf{x}, \mathbf{x}') = \mathbb{E}\left[U(\mathbf{x})U(\mathbf{x}')\right]$ is derived in a similar way, and its calculation involves terms as

$$g_{mn}(\mathbf{d}_{\mathbf{x}}) = \int_{\mathbb{C}^d} \lambda_m \lambda_n h_m(\boldsymbol{\beta} + \mathbf{d}_{\mathbf{x}}) h_n(\boldsymbol{\beta}) \mathrm{d}^d\boldsymbol{\beta}. \tag{40}$$

One general example is to set the filters as parameterized exponentials, $h_i(\mathbf{x}) = v_i \exp(-(\mathbf{x} - \boldsymbol{\mu}_i)^{\mathrm{H}}(\mathbf{x} - \boldsymbol{\mu}_i)/\gamma_i)$, so (38) yields

$$\begin{aligned} &f_{mn}(\mathbf{d}_{\mathbf{x}}) = \\ &\lambda_m \lambda_n^* v_m v_n \exp\left(-\frac{(\mathbf{d}_{\mathbf{x}} - \boldsymbol{\mu}_m + \boldsymbol{\mu}_n)^{\mathrm{H}} (\mathbf{d}_{\mathbf{x}} - \boldsymbol{\mu}_m + \boldsymbol{\mu}_n)}{\gamma_m + \gamma_n}\right) \\ &\cdot \left(\int_{\mathbb{C}^d} \exp\left(-\frac{(\gamma_m + \gamma_n)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^{\mathrm{H}}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})}{\gamma_m \gamma_n}\right) \mathrm{d}^d\boldsymbol{\beta}\right) \\ &= \lambda_m \lambda_n^* v_m v_n \left(\frac{\pi\gamma_m\gamma_n}{\gamma_m + \gamma_n}\right)^d \\ &\cdot \exp\left(-\frac{(\mathbf{d}_{\mathbf{x}} - \boldsymbol{\mu}_m + \boldsymbol{\mu}_n)^{\mathrm{H}} (\mathbf{d}_{\mathbf{x}} - \boldsymbol{\mu}_m + \boldsymbol{\mu}_n)}{\gamma_m + \gamma_n}\right) \\ &= \lambda_m \lambda_n^* \bar{f}_{mn}(\mathbf{d}_{\mathbf{x}}), \end{aligned} \tag{41}$$

where $\hat{\boldsymbol{\beta}} = (\boldsymbol{\mu}_n\gamma_m - (\mathbf{d}_{\mathbf{x}} - \boldsymbol{\mu}_m)\gamma_n)/(\gamma_m + \gamma_n)$, and

$$\begin{aligned} \bar{f}_{mn}(\mathbf{d}_{\mathbf{x}}) &= v_m v_n \left(\frac{\pi\gamma_m\gamma_n}{\gamma_m + \gamma_n}\right)^d \\ &\quad \cdot \exp\left(-\frac{(\mathbf{d}_{\mathbf{x}} - \boldsymbol{\mu}_m + \boldsymbol{\mu}_n)^{\mathrm{H}} (\mathbf{d}_{\mathbf{x}} - \boldsymbol{\mu}_m + \boldsymbol{\mu}_n)}{\gamma_m + \gamma_n}\right). \end{aligned} \tag{42}$$

Analogous calculations yield

$$g_{mn}(\mathbf{d}_{\mathbf{x}}) = \lambda_m \lambda_n \bar{f}_{mn}(\mathbf{d}_{\mathbf{x}}). \tag{43}$$

If, as an example, $\boldsymbol{\mu}_m = \boldsymbol{\mu}_n = \mathbf{0}$ for all posible values of m or n, after some simple mathematical manipulations,

$$\mathbf{C}_U(\mathbf{x}, \mathbf{x}') = \sum_{m=1}^{4} \bar{f}_{mm}(\mathbf{d}_{\mathbf{x}}), \tag{44}$$

where $\bar{f}_{mm}(\mathbf{d}_{\mathbf{x}})$ simplifies to

$$\bar{f}_{mm}(\mathbf{d}_{\mathbf{x}}) = v_m^2 \left(\frac{\pi\gamma_m}{2}\right)^d \exp\left(-\frac{\mathbf{d}_{\mathbf{x}}^{\mathrm{H}}\mathbf{d}_{\mathbf{x}}}{2\gamma_m}\right). \tag{45}$$

And the pseudo-covariance is

$$\tilde{\mathbf{C}}_U(\mathbf{x},\mathbf{x}') = \left(\bar{f}_{11}(\mathbf{d_x}) + \bar{f}_{33}(\mathbf{d_x})\right) - \left(\bar{f}_{22}(\mathbf{d_x}) + \bar{f}_{44}(\mathbf{d_x})\right) + \mathrm{j}\left(2\bar{f}_{12}(\mathbf{d_x}) + 2\bar{f}_{34}(\mathbf{d_x})\right), \quad (46)$$

where $\bar{f}_{mm}$ is given in (45), and $\bar{f}_{mn}$ now simplifies to

$$\bar{f}_{mn}(\mathbf{d_x}) = v_m v_n \left(\frac{\pi\gamma_m\gamma_n}{\gamma_m+\gamma_n}\right)^d \exp\left(-\frac{\mathbf{d}_\mathbf{x}^{\mathrm{H}}\mathbf{d_x}}{\gamma_m+\gamma_n}\right). \quad (47)$$

Notice that in this example the covariance function $k(\mathbf{x},\mathbf{x}') = \mathbf{C}_U(\mathbf{x},\mathbf{x}')$ is real-valued while the pseudo-covariance $\tilde{k}(\mathbf{x},\mathbf{x}') = \tilde{\mathbf{C}}_U(\mathbf{x},\mathbf{x}')$ is complex-valued. This is due to the fact that $\mathbf{K}_{\mathrm{jr}} = \mathbf{K}_{\mathrm{rj}}^\top = \mathbf{K}_{\mathrm{rj}}$ for the process generated with the filters in this example. The examples in (28)-(29) are derived from (44)-(46) when $v_1 = v_3 = v_r$, $\gamma_1 = \gamma_3 = \gamma_r$, and $v_2 = v_4 = v_j$, $\gamma_2 = \gamma_4 = \gamma_j$.

In order to yield a complex covariance function, we need $\mathbf{K}_{\mathrm{jr}} = \mathbf{K}_{\mathrm{rj}}^\top \neq \mathbf{K}_{\mathrm{rj}}$. An example arises when $\mathbf{K}_{\mathrm{jr}}$ is skew-symmetric; $\mathbf{K}_{\mathrm{jr}} = \mathbf{K}_{\mathrm{rj}}^\top = -\mathbf{K}_{\mathrm{rj}}$. In such a case the pseudo-covariance is real while the covariance is complex. In order to get a skew-symmetric $\mathbf{K}_{\mathrm{jr}}$ there must be is a correlation between the real part and a displaced or translated imaginary part, with displacement given by $\boldsymbol{\mu} \in \mathbb{C}, \boldsymbol{\mu} \neq \mathbf{0}$, while there is also a correlation between the real part and a displaced imaginary part when the displacement is given by $-\boldsymbol{\mu}$, and this correlation has the same value with the opposite sign. This is achieved with the following parameter values. For $h_1(\mathbf{x}) = h_3(\mathbf{x})$ we set $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_3 = \mathbf{0}$, $\gamma_1 = \gamma_3 = \gamma_r$ and $v_1 = v_3 = v_\mathrm{r}$. For $h_2(\mathbf{x})$ we set $\boldsymbol{\mu}_2 = \boldsymbol{\mu}$, $v_2 = v_\mathrm{j}$ and $\gamma_2 = \gamma_j$. And for $h_4(\mathbf{x})$ we set $\boldsymbol{\mu}_4 = -\boldsymbol{\mu}$, $v_4 = -v_\mathrm{j}$ and $\gamma_4 = \gamma_j$. In this case, the covariance and pseudo-covariance yield

$$\mathbf{C}_U(\mathbf{x},\mathbf{x}') = 2\bar{f}_{11}(\mathbf{d_x}) + 2\bar{f}_{22}(\mathbf{d_x}) - \mathrm{j}2\left(\bar{f}_{12}(\mathbf{d_x}) + \bar{f}_{34}(\mathbf{d_x})\right), \quad (48)$$

$$\tilde{\mathbf{C}}_U(\mathbf{x},\mathbf{x}') = 2\bar{f}_{11}(\mathbf{d_x}) - 2\bar{f}_{22}(\mathbf{d_x}), \quad (49)$$

where $\bar{f}_{mn}$ is given in (42).

The example in (31)-(32) is derived from (48)-(49) when the inputs are complex-valued scalars $x \in \mathbb{C}$, the displacement is also a complex-valued scalar $\mu \in \mathbb{C}$, and $\gamma_r = \gamma_j = \gamma$.

B. Gradient Descent of the Marginal Likelihood

The log marginal likelihood $L(\boldsymbol{\theta})$ in (33) is a function of a complex-valued Hermitian matrix $\mathbf{C}(\boldsymbol{\theta})$. Therefore, for its maximization we must seek generalized complex-valued matrix derivatives [26], [33]. We start by defining the following function

$$g(\underline{\hat{\mathbf{C}}}, \underline{\hat{\mathbf{C}}}^*) = -\frac{1}{2}\underline{\mathbf{y}}^{\mathrm{H}}\underline{\hat{\mathbf{C}}}^{-1}\underline{\mathbf{y}} - \frac{1}{2}\log\det\underline{\hat{\mathbf{C}}} \quad (50)$$

where $\underline{\hat{\mathbf{C}}}$ is a matrix with independent components, i.e., not Hermitian. The unpatterned matrix input variables $\underline{\hat{\mathbf{C}}}$ and $\underline{\hat{\mathbf{C}}}^*$ should be treated as independent when finding complex-valued matrix derivatives of the function $g(\underline{\hat{\mathbf{C}}}, \underline{\hat{\mathbf{C}}}^*)$. We can find the derivatives of $L(\boldsymbol{\theta})$ in (33) with respect to $\underline{\mathbf{C}}$ and $\underline{\mathbf{C}}^*$ as follows [33]

$$\frac{\partial L}{\partial \underline{\mathbf{C}}} = \left[\frac{\partial g(\underline{\hat{\mathbf{C}}}, \underline{\hat{\mathbf{C}}}^*)}{\partial \underline{\hat{\mathbf{C}}}} + \left(\frac{\partial g(\underline{\hat{\mathbf{C}}}, \underline{\hat{\mathbf{C}}}^*)}{\partial \underline{\hat{\mathbf{C}}}^*}\right)^\top\right]_{\underline{\hat{\mathbf{C}}}=\underline{\mathbf{C}}(\boldsymbol{\theta})}, \quad (51)$$

and

$$\frac{\partial L}{\partial \underline{\mathbf{C}}^*} = \left[\frac{\partial g(\underline{\hat{\mathbf{C}}}, \underline{\hat{\mathbf{C}}}^*)}{\partial \underline{\hat{\mathbf{C}}}^*} + \left(\frac{\partial g(\underline{\hat{\mathbf{C}}}, \underline{\hat{\mathbf{C}}})}{\partial \underline{\hat{\mathbf{C}}}}\right)^\top\right]_{\underline{\hat{\mathbf{C}}}=\underline{\mathbf{C}}(\boldsymbol{\theta})}. \quad (52)$$

Here the problem simplifies since $\partial g(\underline{\hat{\mathbf{C}}}, \underline{\hat{\mathbf{C}}}^*)/\partial \underline{\hat{\mathbf{C}}}^* = \mathbf{0}$. The derivative of $L(\boldsymbol{\theta})$ in (33) with respect to the hyperparameters is found by using the chain rule

$$\begin{aligned}\frac{\partial L}{\partial \theta_i} &= \mathrm{Tr}\left(\left(\frac{\partial L}{\partial \underline{\mathbf{C}}}\right)^\top \frac{\partial \underline{\mathbf{C}}}{\partial \theta_i} + \left(\frac{\partial L}{\partial \underline{\mathbf{C}}^*}\right)^\top \frac{\partial \underline{\mathbf{C}}^*}{\partial \theta_i}\right) \\ &= 2\mathrm{Tr}\left(\left(\frac{\partial g(\underline{\hat{\mathbf{C}}}, \underline{\hat{\mathbf{C}}}^*)}{\partial \underline{\hat{\mathbf{C}}}}\bigg|_{\underline{\hat{\mathbf{C}}}=\underline{\mathbf{C}}(\boldsymbol{\theta})}\right)^\top \frac{\partial \underline{\mathbf{C}}}{\partial \theta_i}\right). \end{aligned} \quad (53)$$

The derivative of the first term of $g(\underline{\hat{\mathbf{C}}}, \underline{\hat{\mathbf{C}}}^*)$ with respect to $\hat{\mathbf{C}}$ yields

$$\frac{\partial}{\partial \underline{\hat{\mathbf{C}}}}\left(-\frac{1}{2}\underline{\mathbf{y}}^{\mathrm{H}}\underline{\hat{\mathbf{C}}}^{-1}\underline{\mathbf{y}}\right) = \frac{1}{2}(\underline{\hat{\mathbf{C}}}^\top)^{-1}\left(\underline{\mathbf{y}}\,\underline{\mathbf{y}}^{\mathrm{H}}\right)^\top(\underline{\hat{\mathbf{C}}}^\top)^{-1}. \quad (54)$$

The derivative of the second term of $g(\underline{\hat{\mathbf{C}}}, \underline{\hat{\mathbf{C}}}^*)$ with respect to $\hat{\mathbf{C}}$ yields

$$\frac{\partial}{\partial \underline{\hat{\mathbf{C}}}}\left(-\frac{1}{2}\log\det\underline{\hat{\mathbf{C}}}\right) = -\frac{1}{2}(\underline{\hat{\mathbf{C}}}^\top)^{-1}. \quad (55)$$

Substitution of (54) and (55) in (53) yield (34).

REFERENCES

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

[2] D. Mandic and V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*. Wiley Publishing, 2009.

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

[4] M. E. Valle, "Complex-valued recurrent correlation neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 9, pp. 1600–1612, Sept 2014.

[5] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, ser. Adaptive computation and machine learning. MIT Press, 2002.

[6] I. Steinwart, D. Hush, and C. Scovel, "An explicit description of the reproducing kernel hilbert spaces of gaussian rbf kernels," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4635–4643, Oct 2006.

[7] T. Ogunfunmi and T. Paul, "On the complex kernel-based adaptive filter," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, May 2011, pp. 1263–1266.

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

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

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

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

[12] C. Lajaunie and R. Béjaoui, "Sur le krigeage des functions complexes," Centre de Geostatistique, Ecole des Mines de Paris, Fontainebleau, July 1991, note N-23/91/G.

[13] S. De Iaco, M. Palma, and D. Posa, "Covariance functions and models for complex-valued random fields," *Stochastic Environmental Research and Risk Assessment*, vol. 17, no. 3, pp. 145–156, 2003.

[14] S. De Iaco and D. Posa, "Wind velocity prediction through complex kriging: formalism and computational aspects," *Environmental and Ecological Statistics*, vol. 23, no. 1, pp. 115–139, 2016.

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

[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: MIT Press, 2006.

[17] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, Mar 2001.

[18] J. Li, B. Zhang, and D. Zhang, "Shared autoencoder gaussian process latent variable model for visual classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–15, 2017.

[19] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela, "Bayesian nonparametric adaptive control using gaussian processes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 537–550, March 2015.

[20] G. Skolidis and G. Sanguinetti, "Semisupervised multitask learning with gaussian processes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 12, pp. 2101–2112, Dec 2013.

[21] M. Lázaro-Gredilla and S. V. Vaerenbergh, "A gaussian process model for data association and a semidefinite programming solution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 1967–1979, Nov 2014.

[22] R. C. Grande, T. J. Walsh, G. Chowdhary, S. Ferguson, and J. P. How, "Online regression for data with changepoints using gaussian processes and reusable models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 9, pp. 2115–2128, Sept 2017.

[23] F. Pérez-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.

[24] R. Boloix-Tortosa, F. J. Payán-Somet, and J. J. Murillo-Fuentes, "Gaussian processes regressors for complex proper signals in digital communications," in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, June 2014, pp. 137–140.

[25] F. D. Neeser and J. L. Massey, "Proper complex random processes with applications to information theory," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1293–1302, Jul 1993.

[26] A. Hjorungnes and D. P. Palomar, "Patterned complex-valued matrix derivatives," in *2008 5th IEEE Sensor Array and Multichannel Signal Processing Workshop*, July 2008, pp. 293–297.

[27] P. Boyle and M. Frean, "Dependent gaussian processes," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, 2005, pp. 217–224. [Online]. Available: http://papers.nips.cc/paper/2561-dependent-gaussian-processes.pdf

[28] C. A. Calder and N. Cressie, "Some topics in convolution-based spatial modeling," in *Proceedings of the 56th Session of the International Statistics Institute*, Lisbon, Portugal, August 2007, pp. 22–29.

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

[30] E. Snelson, C. Rasmussen, and Z. Ghahramani, "Warped gaussian processes," in *Advances in Neural Information Processing Systems 16*, Max-Planck-Gesellschaft. Cambridge, MA, USA: MIT Press, Jun. 2004, pp. 337–344.

[31] M. Lázaro-Gredilla, "Bayesian warped gaussian processes," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1619–1627. [Online]. Available: http://papers.nips.cc/paper/4494-bayesian-warped-gaussian-processes.pdf

[32] E. Ollila, "On the circularity of a complex random variable," *IEEE Signal Processing Letters*, vol. 15, pp. 841–844, 2008.

[33] A. Hjorungnes and D. Gesbert, "Complex-valued matrix differentiation: Techniques and key results," *IEEE Trans. Signal Processing*, vol. 55, no. 6, pp. 2740–2746, June 2007.

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

[35] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, 1st ed. Wiley Publishing, 2010.

[36] J. Platt, "A resource-allocating network for function interpolation," *Neural Computation*, vol. 3, no. 2, pp. 213–225, June 1991.