POLITECNICO DI TORINO
Repository ISTITUZIONALE

Bridging the Gap Between Artificial Neural Networks and Kernel Regressions for Vector-Valued Problems in Microwave Applications

Original
Bridging the Gap Between Artificial Neural Networks and Kernel Regressions for Vector-Valued Problems in Microwave Applications / Soleimani, N., Trinchero, R., Canavero, F.G.. - In: IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES. - ISSN 0018-9480. - STAMPA. - 71:6(2023), pp. 2319-2332. [10.1109/TMTT.2022.3232895]

Availability:
This version is available at: 11583/2974826 since: 2023-06-06T12:02:30Z

Publisher:
IEEE

Published
DOI:10.1109/TMTT.2022.3232895

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

18 November 2026

Bridging the Gap Between Artificial Neural Networks and Kernel Regressions for Vector-Valued Problems in Microwave Applications

Nastaran Soleimani, *Student Member, IEEE*, Riccardo Trinchero, *Member, IEEE*, and Flavio G. Canavero, *Life Fellow, IEEE*

***Abstract*— Thanks to their convex formulation, kernel regressions have shown an improved accuracy with respect to artificial neural network (ANN) structures in regression problems where a reduced set of training samples are available. However, despite the above interesting features, kernel regressions are inherently less flexible than ANN structures since their implementations are usually limited to scalar-output regression problems. This article presents a vector-valued (multioutput) formulation of the kernel ridge regression (KRR) aimed at bridging the gap between multioutput ANN structures and scalar kernel-based approaches. The proposed vector-valued KRR relies on a generalized definition of the reproducing kernel Hilbert space (RKHS) and on a new multioutput kernel structure. The mathematical background of the proposed vector-valued formulation is extensively discussed together with different matrix kernel functions and training schemes. Moreover, a compression strategy based on the Nystrom approximation is presented to reduce the computational complexity of the model training. The effectiveness and the performance of the proposed vector-valued KRR are discussed on an illustrative example consisting of a high-speed link and on the optimization of a Doherty amplifier.**

***Index Terms*— Kernel, kernel ridge regression (KRR), microwave structures, optimization, parametric modeling, reproducing kernel Hilbert space (RKHS), vector-valued kernel regression.**

I. Introduction

In THE last decades, machine learning (ML) and data-driven techniques have been widely adopted to construct accurate and fast-to-evaluate surrogate models [1], [2], [2], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] and macromodels [17], [18], [19] able to mimic the parametric behavior of complex electromagnetic (EM) structures provided by computationally expensive models (i.e., EM or circuital solvers). The underlying idea is to adopt supervised regressions to construct a closed-form approximation of the input–output behavior of the structures under modeling by using a "small" set of training samples (i.e., simulations carried out with EM or circuital solvers for different configurations of the input parameters) [20]. The resulting surrogate model can then be inexpensively adopted to explore the design space within optimization and uncertainty quantification algorithms, thus providing an efficient alternative to computationally expensive simulations in microwave applications. In the above scenario, two different classes of supervised ML regressions have emerged, such as artificial neural network (ANN) [1], [2], [2], [4], [5], [6], [7] and kernel regression techniques [8], [9], [10], [11], [12], [13].

According to the universal approximation theorem [21], ANN structures can approximate any nonlinear function or a set of functions for the multioutput case, via a collection of artificial neurons connected together and organized in layers. The overall structure turns out to be extremely flexible, without any limitation in terms of number of layers, neurons per layer, number of outputs, and so on. Moreover, the mathematical model describing the input–output map obtained by the ANN is usually not linear with respect to the model unknowns (i.e., the weights and bias) since they appear within the argument of nonlinear functions (i.e., the activation functions). This allows learning very complex nonlinear behaviors, but on the other hand, the nonlinear structure of the ANN model leads to a nonconvex optimization problem with several local minima. Such nonconvex optimization makes the training phase for the ANN rather complicated and data-hungry [14], [16].

Kernel-based regressions [22], [23], [24], [25] provide an interesting alternative to the above ANN structures, especially in regression problems in which a "relatively small" set of training data is available. Kernel regressions can be seen as a linear and less flexible interpretation of the more general ANN formulation. As shown in Fig. 1, a generic kernel model can be interpreted as ANN structure with a single hidden layer, in which the unknown weights (i.e., $\{\alpha_1, \ldots, \alpha_L\}$) appear linearly as the connection between the hidden and the output layer [22], [23], [24]. It is important to remark that in such structure, the number of both weights and neurons in the hidden layer is fixed and turns out to be equal to the number of training samples [24] (or less for the support vector machine regression [22], [23]). This means that the overall model

Manuscript received 23 September 2022; revised 2 December 2022; accepted 21 December 2022. Date of publication 4 January 2023; date of current version 5 June 2023. This article is an expanded version from the 2022 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization, Limoges (France), July 6–8. *(Corresponding author: Riccardo Trinchero.)*

The authors are with the EMC Group, Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Turin, Italy (e-mail: nastaran.soleimani@polito.it; riccardo.trinchero@polito.it; flavio.canavero@polito.it).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMTT.2022.3232895.

Digital Object Identifier 10.1109/TMTT.2022.3232895

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

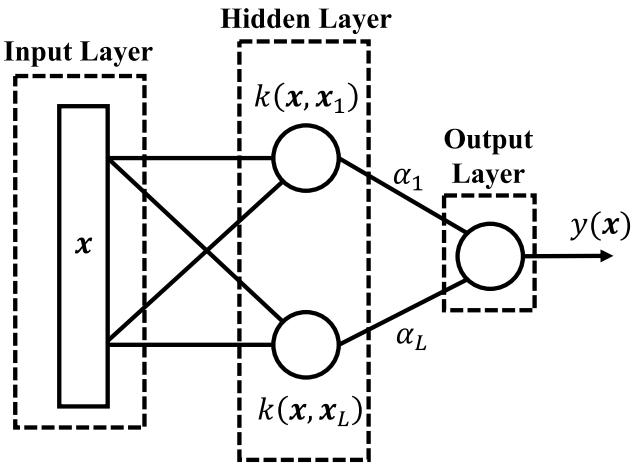


Fig. 1. ANN interpretation of a scalar-output kernel regression (the picture is inspired by [24]).

complexity in terms of regression unknowns turns out to be independent of the number of input parameters considered by the model [22]. Unlike ANNs, the linear model structure adopted by kernel regression (i.e., the model unknowns appear linearly) has the key advantage of heavily simplifying the training phase, which leads to the solution of a standard convex optimization problem [25], thus providing several advantages in terms of training time and accuracy with respect to the number of training samples [12], [14], [16], [25].

Conversely, the lack of flexibility needed to guarantee the linear structure in the advocated scalar kernel regressions leads to some limitations with respect to ANN. Indeed, due to their inherently scalar nature, plain kernel regressions are not able to deal with multioutput problems [26]. Unfortunately, multioutput or vector-valued regression problems are quite common in microwave and electronic applications. As an example, we can consider the problem of building a parametric model able to learn the parametric behavior of the scattering parameters of an amplifier as a function of the values of its geometrical and electrical parameters. In the above scenario, under the assumption that the realizations for each output dimension (e.g., frequency samples) are uncorrelated, the learning problem turns out to be equivalent to learn a set of single-output models, one for each output dimension [12], [27]. Unfortunately, when the number of output dimensions is in the order of several hundreds, such a procedure turns out to be quite cumbersome since it would require to independently train a large number of scalar models, along with the tuning of a huge number of hyperparameters. Also, such an approach unavoidably neglects any possible correlation among the output dimensions, thus leading to an overall model with possible overfitting issue and highly vulnerable to noise [26].

Data compression techniques can be seen as a promising alternative to the above brute-force approach. Compression strategy, such as principal component analysis (PCA) [28], can be used to explore and remove redundant information from the available dataset, leading to a compressed representation of the output dimensions. After the above compression, the number of output components to be modeled can be significantly reduced, thus requiring the training of a reduced set of single-output regressions [12], [27]. Such a technique exploits the statistical correlations among the different components of the output dimensions. Due to its statistical nature, it provides extremely accurate results in the uncertainty quantification scenario [12], [27], [29]. On the contrary, if the number of components in the compressed representation is not carefully tuned, the obtained model can have a limited generalization capability on unseen data, thus leading to possible lack of accuracy for the case of deterministic parametric models (e.g., the ones used for optimization purposes) [29].

This article makes use of a vector-valued formulation of the kernel ridge regression (KRR) able to deal with multioutput regression problems. The proposed methodology is based on a generalized definition of the reproducing kernel Hilbert space (RKHS) and kernel functions in the case of vector-valued learning problem presented in [26], [30], [31], and [32] and extends the preliminary implementation briefly presented in [29]. The proposed multioutput KRR aims at reducing the gap between kernel-based regressions and multioutput ANN structures. The advantages and drawbacks of the proposed vector-values KRR will be widely discussed in this article, as well as the features of different multioutput kernel functions. Moreover, a Nystrom approximation is here proposed in order to mitigate the computational complexity of model training [25], [33], [34], [35]. The effectiveness and the performance of the proposed approach will be investigated on an illustrative example consisting of a high-speed link and by considering the optimization of a Doherty amplifier.

The remainder of this article is organized as follows. Section II briefly introduces the scalar KRR. Section III presents the extension of the KRR to vector-valued problems. Section IV discusses different kernel functions for the proposed vector-valued formulation, along with the corresponding training strategy. Section V presents a compression technique able to reduce the training complexity based on the Nystrom approximation. The performance of the proposed approach is discussed in Sections VI and VII based on an illustrative example and for the optimization of a Doherty amplifier. Finally, conclusions are drawn in Section VIII.

II. Representer Theorem and Scalar KRR

This section discusses the mathematical background of supervised scalar-output kernel regressions, with specific emphasis on KRR.

A. Representer Theorem for Scalar Kernel Regression

First, let us start introducing the representer theorem for a generic scalar kernel regression. We consider a set of training pairs $\mathcal{S} = \{(\boldsymbol{x}_l, y_l)\}_{l=1}^{L}$, where $\boldsymbol{x}_l \in \mathcal{X} \subseteq \mathbb{R}^p$ represents the training input and $y_l \in \mathcal{Y} \subseteq \mathbb{R}$ are the corresponding scalar outputs generated by the actual function under modeling $f(\boldsymbol{x})$ (i.e., $y_l = f(\boldsymbol{x}_l) + \eta$, where η represents a random noise). Knowing the training set, we seek the "best" structure of a generic function $\hat{f}(\boldsymbol{x})$ able to approximate $f(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathcal{X}$ by minimizing the following functional, also known as the

empirical risk minimization (ERM) [36]:

$$\hat{f} = \underset{\tilde{f}\in\mathcal{H}}{\arg\min} \sum_{l=1}^{L} \ell\big(\boldsymbol{x}_l, y_l, \tilde{f}(\boldsymbol{x}_l)\big) + \lambda\Omega\big(\|\tilde{f}\|_{\mathcal{H}}\big) \tag{1}$$

where $\ell(\cdot)$ is the generic loss function providing the "error" between the training outputs and the predictions of a generic model $\tilde{f} \in \mathcal{H}$ evaluated on the corresponding training inputs, $\Omega(\cdot) : \mathbb{R} \to \mathbb{R}$ is a strictly increasing function applied to the L_2-norm of the function $\tilde{f}$ acting as a regularizer, and λ is the hyperparameter associated with it.

According to the representation theorem [36], any optimal solution $\hat{f}(\boldsymbol{x})$ of (1) can be written as

$$\hat{f}(\boldsymbol{x}) = \sum_{l=1}^{L} \alpha_l k(\boldsymbol{x}_l, \boldsymbol{x}) \tag{2}$$

where $k(\cdot,\cdot) : \mathbb{R}^{p\times p} \to \mathbb{R}$ is the so-called kernel function (additional details are provided in Appendix A).

B. Scalar KRR

Scalar KRR can be seen as a special case of the above general framework, in which a squared loss is used as a loss function $\ell(\cdot)$ and $\Omega(\|\tilde{f}\|_{\mathcal{H}}) = \|\tilde{f}\|_{\mathcal{H}}^2$ is a Tikhonov regularizer [37]. Under the above assumptions, the optimization in (1) for the KRR writes

$$\hat{f} = \underset{\tilde{f}\in\mathcal{H}}{\arg\min} \sum_{l=1}^{L} \big(y_l - \tilde{f}(\boldsymbol{x}_l)\big)^2 + \lambda\|\tilde{f}\|_{\mathcal{H}}^2. \tag{3}$$

From the representation theorem, we know that any solution $\hat{f}$ takes the form in (2). Plugging (2) into (3), we can write

$$\min_{\boldsymbol{\alpha}} \sum_{l=1}^{L} \left(y_l - \sum_{m=1}^{L} \alpha_m k(\boldsymbol{x}_m, \boldsymbol{x}_l) \right)^2 + \lambda \left\| \sum_{l=1}^{L} \alpha_l k(\boldsymbol{x}_l, \boldsymbol{x}) \right\|_{\mathcal{H}}^2 \tag{4}$$

where according to the kernel properties [36]

$$\begin{aligned} \|\hat{f}\|_{\mathcal{H}}^2 &= \left\| \sum_{l=1}^{L} \alpha_l k(\boldsymbol{x}_l, \boldsymbol{x}) \right\|_{\mathcal{H}}^2 \\ &= \sum_{l,m=1}^{L} \alpha_l \alpha_m k(\boldsymbol{x}_l, \boldsymbol{x}_m) = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}. \end{aligned} \tag{5}$$

In (5), $\mathbf{K} \in \mathbb{R}^{L\times L}$ is the empirical kernel matrix, also known as kernel Gram matrix, defined by evaluating the kernel function on each configuration pair belonging to the training input set such that

$$[\mathbf{K}]_{ij} = k\big(\boldsymbol{x}_i, \boldsymbol{x}_j\big) \tag{6}$$

for any $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the training input set.

The optimization problem in (3) can be written in its matrix form as

$$\min_{\boldsymbol{\alpha}} (\boldsymbol{y} - \mathbf{K}\boldsymbol{\alpha})^T (\boldsymbol{y} - \mathbf{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \tag{7}$$

where $\boldsymbol{y} = [y_1, \ldots, y_L]^T$ is a vector collecting the training outputs, whereas $\mathbf{K}\boldsymbol{\alpha}$ represents the corresponding predictions computed via (2), such that for the nth training output, we get

$$y_n \approx \sum_{l=1}^{L} \alpha_l k(\boldsymbol{x}_l, \boldsymbol{x}_n) = \mathbf{K}_{[n,:]}\boldsymbol{\alpha} \tag{8}$$

where $\mathbf{K}_{[n,:]}$ represents the nth row of the Gram matrix $\mathbf{K}$ [38], [39].

The cost function in (7) can be expanded as

$$\begin{aligned} E(\boldsymbol{\alpha}) = \; & \boldsymbol{y}^T \boldsymbol{y} - \boldsymbol{\alpha}^T \mathbf{K}^T \boldsymbol{y} + \\ & - \boldsymbol{y}^T \mathbf{K}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{K}^T \mathbf{K}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha}. \end{aligned} \tag{9}$$

The above cost function can be minimized by computing the zero of its partial derivative with respect to the vector of unknowns $\boldsymbol{\alpha}$, which is written as

$$\frac{\partial E(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = -2\mathbf{K}^T \boldsymbol{y} + 2\mathbf{K}^T \mathbf{K}\boldsymbol{\alpha} + 2\lambda \mathbf{K}\boldsymbol{\alpha} = \mathbf{0}. \tag{10}$$

Since the kernel function is symmetric by construction (i.e., $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_j, \boldsymbol{x}_i)$), the Gram matrix $\mathbf{K}$ associated with the kernel is a square symmetric matrix, such as $\mathbf{K}^T = \mathbf{K}$. This means that (10) can be simplified as follows:

$$-\mathbf{K}\boldsymbol{y} + \mathbf{K}^2\boldsymbol{\alpha} + \lambda\mathbf{K}\boldsymbol{\alpha} = 0 \tag{11}$$

which can be recast in terms of the following linear system:

$$(\mathbf{K} + \lambda \mathbf{I}_L)\mathbf{K}\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{y} \tag{12}$$

where $\mathbf{I}_L$ refers to the $L \times L$ identity matrix.

Since all the matrices in the left side are symmetric matrices (i.e., $\mathbf{AB} = \mathbf{BA}$), the above linear system is equivalent to

$$\mathbf{K}(\mathbf{K} + \lambda \mathbf{I}_L)\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{y} \tag{13}$$

which leads to the well-known formulation of the KRR

$$(\mathbf{K} + \lambda \mathbf{I}_L)\boldsymbol{\alpha} = \boldsymbol{y}. \tag{14}$$

Therefore, the model coefficients in the vector $\boldsymbol{\alpha}$ can be suitably computed by solving the above linear system of equations, i.e.,

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I}_L)^{-1} \boldsymbol{y}. \tag{15}$$

III. From Scalar- to Vector-Valued KRR

This section aims at providing a generalized formulation of the scalar-output kernel-based regression presented in Section II for vector-valued output or multitask regression. For the sake of simplicity, this article will focus on the specific case of vector-valued regression for which the training set is defined as $\mathcal{S} = \{(\boldsymbol{x}_l, \boldsymbol{y}_l)\}_{l=1}^{L}$, in which $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^p$ is a vector collecting the configurations of the input parameters (e.g., geometrical and electrical parameters of an EM structure) and $\boldsymbol{y}_i = [y_i^{(1)}, \ldots, y_i^{(D)}]^T \in \mathbb{R}^D$ is a vector collecting the corresponding vector-valued training outputs (e.g., the frequency samples of frequency response). The above training set can be rewritten in its compact form as $\mathcal{S} = \{(\mathbf{X}, \mathbf{Y})\}$, where $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L]^T$ is an $L \times p$ matrix collecting the configurations of the training input and $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L]^T$ is an $L \times D$ matrix associated with the training outputs.

A. Reproducing Hilbert Space for Vector-Valued KRR

Given the information available in the training set $\mathcal{S}$, our goal is to learn D scalar functions $\hat{f}^{(d)} : \mathcal{X} \to \mathbb{R}$ with

$d = 1, \ldots, D$, able to provide an accurate prediction of the actual output vector $\boldsymbol{y}(\boldsymbol{x})$ for any configuration of the parameters $\boldsymbol{x} \in \mathcal{X}$.[1] In order to deal with the above vector-valued regression problem, the learning problem in (3) must be generalized as follows:

$$\hat{\boldsymbol{f}} = \arg\min_{\tilde{\boldsymbol{f}} \in \mathcal{H}} \sum_{d=1}^{D} \sum_{l=1}^{L} \left(y_l^{(d)} - \tilde{f}^{(d)}(\boldsymbol{x}_l) \right)^2 + \lambda \|\tilde{\boldsymbol{f}}\|_{\mathcal{H}}^2 \quad (16)$$

where $y_l^{(d)}$ and $\tilde{f}^{(d)}(\boldsymbol{x}_l)$ represent the dth component of the lth training output and the corresponding model prediction, respectively.

According to the represented theorem for vector-valued regression problem presented in [31], any solution $\hat{\boldsymbol{f}}$ of (16) takes the form

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{l=1}^{L} \mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l) \boldsymbol{c}_l \quad (17)$$

where $\mathbf{K}(\cdot, \cdot) : \mathbb{R}^{p \times p} \to \mathbb{R}^{D \times D}$ is a matrix multioutput kernel acting on the column vectors $\boldsymbol{c}_l \in \mathbb{R}^D$ collecting the regression unknowns for $l = 1, \ldots, L$. For a generic output n, (17) can be written as

$$\begin{aligned} \hat{f}^{(n)}(\boldsymbol{x}) &= \sum_{l=1}^{L} [\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)]_{[n,:]} \boldsymbol{c}_l \\ &= \sum_{d=1}^{D} \sum_{l=1}^{L} [\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)]_{[n,d]} c_{d,l} \end{aligned} \quad (18)$$

where $[\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)]_{[n,:]}$ and $[\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)]_{[n,d]}$ denote the nth row and the (n, d)-element of the matrix kernel $\mathbf{K}(\cdot, \cdot)$, respectively, and $c_{d,l}$ is the dth element of the vector $\boldsymbol{c}_l$.

Equation (18) can be rewritten in its scalar form, i.e.,

$$f^{(n)}(\boldsymbol{x}) = \sum_{d=1}^{D} \sum_{l=1}^{L} k((\boldsymbol{x}, n), (\boldsymbol{x}_l, d)) c_{d,l} \quad (19)$$

where $k((\boldsymbol{x}, n), (\boldsymbol{x}_l, d)) : \mathbb{R}^{p \times p} \times \mathbb{R}^{\{1,\ldots,D\} \times \{1,\ldots,D\}} \to \mathbb{R}$ represents the (n, d) entry of the multioutput kernel matrix $\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)$ such that $k((\boldsymbol{x}, n), (\boldsymbol{x}_l, d)) = [\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)]_{[n,d]}$.

B. Separable Multioutput Kernels for Vector-Valued Regression

The kernel structure in (18) and (19) was introduced in [30]. The multioutput kernel should be able to account for the correlation in both the parameter space and output components. Unfortunately, there do not exist off-the-shelf kernel functions, which can be directly applied in such context. The simplest solution is to work on a specific class of multioutput kernels such as the separable kernel or sum of separable kernels [26], [30], [31]. Specifically, we will consider matrix kernel functions $\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}')$, obtained as the product between two scalar kernels acting either on the input space or on the output dimensions, such that

$$\begin{aligned} \left[\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}')\right]_{[d,d']} &= k\left((\boldsymbol{x}, d), (\boldsymbol{x}_l, d')\right) \\ &= k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}') k_o(d, d') \end{aligned} \quad (20)$$

[1]The proposed formulation can be extended to the more general case of multitask formulation in which the number of training samples L_d can vary for each output d, as well as the number of parameters p_d.

where $k_{\boldsymbol{x}}$ and k_o are scalar kernels acting independently on the input space (i.e., $k_{\boldsymbol{x}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$) and output dimensions (i.e., $k_o : \{1, \ldots, D\} \times \{1, \ldots, D\} \to \mathbb{R}$).

Therefore, for each pair $\boldsymbol{x}$ and $\boldsymbol{x}'$ belonging to the input space $\mathcal{X}$, the resulting multioutput kernel matrix $\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}')$ can be written as

$$\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}') = k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}') \mathbf{B} \quad (21)$$

where $\mathbf{B} \in \mathbb{R}^{D \times D}$ is a symmetric semidefinite matrix completely independent of the input parameters $\boldsymbol{x}$ and $\boldsymbol{x}'$, in which its elements are obtained by evaluating the scalar kernel k_o on the output dimensions (i.e., $\{1, \ldots, D\} \times \{1, \ldots, D\}$). The overall kernel matrix $\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}')$ is a $D \times D$ symmetric matrix by construction since it is the product of a symmetric function $k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}')$ with a symmetric matrix $\mathbf{B}$.

By combining the optimal solution in (17) for the vector-output scenario, with the separable kernel structure in (20), we can write

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{l=1}^{L} \mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l) \boldsymbol{c}_l = \sum_{l=1}^{L} k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}_l) \mathbf{B} \boldsymbol{c}_l. \quad (22)$$

C. Matrix Formulation for the Vector-Valued KRR With Separable Kernel

Let us now consider the following matrix formulation of the ERM in (16) developed for the vector-valued scenario:

$$\min_{\boldsymbol{f} \in \mathcal{H}} \|\mathbf{Y} - \mathbf{F}\|_F^2 + \lambda \|\boldsymbol{f}\|_{\mathcal{H}}^2 \quad (23)$$

where $\mathbf{F} = [\boldsymbol{f}_1^T, \ldots, \boldsymbol{f}_L^T]$ is an $L \times D$ matrix collecting the model predictions for the samples in the training set, such that $[\mathbf{F}]_{ij} = f^{(j)}(\boldsymbol{x}_i)$, and $\|\cdot\|_F$ is the Frobenius norm defined as

$$\|\mathbf{A}\|_F^2 = \sum_{i=1} \sum_{j=1} a_{ij}^2 = \mathrm{Tr}\left(\mathbf{A}\mathbf{A}^T\right). \quad (24)$$

According to (18) and (21), the n-row of the matrix $\mathbf{F}$ in (23) can be written as

$$\begin{aligned} [\mathbf{F}]_{[n,:]} = \hat{\boldsymbol{f}}(\boldsymbol{x}_n)^T &= \sum_{l=1}^{L} k_x(\boldsymbol{x}_n, \boldsymbol{x}_l) \boldsymbol{c}_l^T \mathbf{B}^T \\ &= \sum_{l=1}^{L} k_x(\boldsymbol{x}_n, \boldsymbol{x}_l) \boldsymbol{c}_l^T \mathbf{B}. \end{aligned} \quad (25)$$

Since $\mathbf{B}$ is a symmetric matrix (i.e., $\mathbf{B} = \mathbf{B}^T$), the matrix $\mathbf{F}$ can be rewritten as [40]

$$\mathbf{F} = \mathbf{K}_{\boldsymbol{x}} \mathbf{C} \mathbf{B} \quad (26)$$

where $\mathbf{K}_x \in \mathbb{R}^{L \times L}$ with $[\mathbf{K}_{\boldsymbol{x}}]_{[ij]} = k_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Gram matrix associated with the kernel $k_{\boldsymbol{x}}$ evaluated on the input training samples and $\mathbf{C} \in \mathbb{R}^{L \times D}$ is a matrix collecting the regression unknowns $\boldsymbol{c}_l$ such that $\mathbf{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L]^T$

By substituting the above model structure in (23), we get the following optimization problem:

$$\min_{\mathbf{C}} \|\mathbf{Y} - \mathbf{K}_{\boldsymbol{x}} \mathbf{C} \mathbf{B}\|_F^2 + \lambda \left\langle \mathbf{C}^T \mathbf{K}_{\boldsymbol{x}} \mathbf{C}, \mathbf{B} \right\rangle_F \quad (27)$$

in which $\langle \cdot, \cdot \rangle_F$ is the inner Frobenius product, which for the case of matrices $\mathbf{A}$ and $\mathbf{B}$ can be written as

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = \sum_{i,j} A_{ij} B_{ij} = \mathrm{Tr}\big(\mathbf{A}^T \mathbf{B}\big). \tag{28}$$

The optimal values for the entries of the coefficient matrix $\mathbf{C}$ can be estimated as the ones for which the partial derivatives of the cost function in (27), computed with respect to them, are equal to zero. By doing this, after some calculations provided in Appendix B, we get the following discrete-time Sylvester equation:

$$\mathbf{K}_{\boldsymbol{x}} \mathbf{C} \mathbf{B} + \lambda \mathbf{C} = \mathbf{Y}. \tag{29}$$

Equation (29) can be solved in a closed form by using the Kronecker formulation [41] such that [40]

$$\underbrace{(\mathbf{B} \otimes \mathbf{K}_{\boldsymbol{x}} + \lambda \mathbf{I}_{LD})}_{\mathbf{A}} \mathrm{vec}(\mathbf{C}) = \mathrm{vec}(\mathbf{Y}) \tag{30}$$

where $\otimes$ is the Kronecker product and $\mathbf{I}_{LD}$ refers to the $LD \times LD$ identity matrix and the vec$(\cdot)$ operator stacks column of its argument matrix into a column vector; therefore, vec$(\mathbf{C}) \in \mathbb{R}^{LD}$ is a vector collecting the regression coefficients $\boldsymbol{c}_l$ in (17), with $\mathbf{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L]^T \in \mathbb{R}^{L \times D}$. Like the scalar case, (30) can be rewritten in terms of the Gram vector-valued matrix $\mathbf{K}$ such that

$$(\mathbf{K} + \lambda \mathbf{I}_{LD}) \, \mathrm{vec}(\mathbf{C}) = \mathrm{vec}(\mathbf{Y}) \tag{31}$$

where the Gram vector-valued matrix $\mathbf{K} \in \mathbb{R}^{(LD) \times (LD)}$ associated with the whole input training dataset $\mathbf{X}$ and output components can be written as

$$\mathbf{K} = \mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k_{\boldsymbol{x}}(X, X) = \mathbf{B} \otimes \mathbf{K}_{\boldsymbol{x}}. \tag{32}$$

It is straightforward to see that the unknown coefficients collected in the vector vec$(\mathbf{C})$ can be computed by solving a linear system of equations, which can be written as

$$\mathrm{vec}(\mathbf{C}) = (\mathbf{B} \otimes \mathbf{K}_{\boldsymbol{x}} + \lambda \mathbf{I}_{LD})^{-1} \mathrm{vec}(\mathbf{Y}). \tag{33}$$

After computing the regression coefficients, (22) can be used to make predictions for a generic input configuration $\boldsymbol{x} \in \mathcal{X}$

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{l=1}^{L} k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}_l) \mathbf{B} \boldsymbol{c}_l = \sum_{l=1}^{L} \mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l) \boldsymbol{c}_l. \tag{34}$$

IV. Separable Kernels for Vector-Valued KRR and Inversion Strategies

This section aims at discussing possible solutions for the design of separable kernel functions tailored for vector-valued KRR, as well as their key features and training strategies.

A. Block-Diagonal Multioutput Kernel Matrix

The discussion starts considering a special case of the separable kernel function in (20), in which the kernel acting on the output dimensions $k_o(d, d') = \delta_{d,d'}$ such that

$$k_{\boldsymbol{x}}\big(\boldsymbol{x}, \boldsymbol{x}'\big) k_o\big(d, d'\big) = k_{\boldsymbol{x}}\big(\boldsymbol{x}, \boldsymbol{x}'\big) \delta_{d,d'} \tag{35}$$

where $\delta_{d,d'}$ is the Kronecker delta. This means that in (22), we are considering $\mathbf{B} = \mathbf{I}_D$ (i.e., the identity matrix).

In the above case, the overall regression problem turns out to be equivalent to train D scalar regression problems using the same kernel function $k_{\boldsymbol{x}}$. Therefore, the associated Gram kernel matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$ becomes an $LD \times LD$ block-diagonal matrix

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathrm{diag}(\mathbf{K}_{\boldsymbol{x}}, \ldots, \mathbf{K}_{\boldsymbol{x}}) = \begin{bmatrix} \mathbf{K}_{\boldsymbol{x}} & \mathbf{0} & \vdots \\ \mathbf{0} & \ddots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{K}_{\boldsymbol{x}} \end{bmatrix}. \tag{36}$$

Such decoupled interpretation of the vector-valued KRR has several advantages with respect to the standard modeling scheme in which a plain scalar kernel regression is applied to construct a set of independent surrogate models, one for each output dimension. Indeed, even if the multioutput kernel in (35) still considers the output dimensions to be independent, it allows to learn them in one shoot via the solution of single optimization problem. This means that the number of hyperparameters to be tuned during the model training is independent of the number of output dimensions since it is determined by the structure of scalar kernel $k_{\boldsymbol{x}}$, only. Also, possible correlations among the output dimensions are inherently accounted for during the training phase by means of the hyperparameters tuning since the latter operation is carried out on the whole training set and output dimensions.

It is important to notice that due to the block-diagonal structure of the Gram kernel matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$ in (36), the regression training turns out to be extremely efficient. Indeed, the overall inversion of the $LD \times LD$ Gram matrix $\mathbf{K}$ reduces to invert an $L \times L$ matrix, i.e.,

$$\begin{aligned} [\mathbf{K} + \lambda \mathbf{I}_{LD}]^{-1} &= \big[\mathrm{diag}((\mathbf{K}_x + \lambda \mathbf{I}_L), \ldots, (\mathbf{K}_x + \lambda \mathbf{I}_L))\big]^{-1} \\ &= \mathrm{diag}\big([\mathbf{K}_x + \lambda \mathbf{I}_L]^{-1}, \ldots, [\mathbf{K}_x + \lambda \mathbf{I}_L]^{-1}\big) \\ &= \mathbf{I}_D \otimes [\mathbf{K}_x + \lambda \mathbf{I}_L]^{-1}. \end{aligned} \tag{37}$$

Due to the block-diagonal structure of the vector-valued kernel matrix $\mathbf{K}$ in (36), the overall computational complexity required by the matrix inversion reduces from $\mathcal{O}(L^3 D^3)$ to $\mathcal{O}(L^3)$ (i.e., the computational cost required to invert the submatrix $[\mathbf{K}_x + \lambda \mathbf{I}_L]$) since the hyperparameters of the kernel $k_{\boldsymbol{x}}$ and λ are shared by all the output dimensions.

B. Coupled Multioutput Kernel Matrix

A possible alternative for the kernel k_o acting on the output dimensions is provided by the so-called mixed kernel [30], which can be written as

$$k_o\big(d, d'\big) = \omega + (1 - \omega) \delta_{d,d'} \tag{38}$$

or equivalently to a matrix $\mathbf{B}$ in (22)

$$\mathbf{B} = \omega \mathbf{1} + (1 - \omega) \mathbf{I}_D \tag{39}$$

where $\mathbf{1}$ is a $D \times D$ matrix whose entries are equal to 1 and $\omega \in [0, 1]$ is the kernel hyperparameter.

The resulting Gram kernel matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is a coupled matrix accounting for a possible uniform correlation among

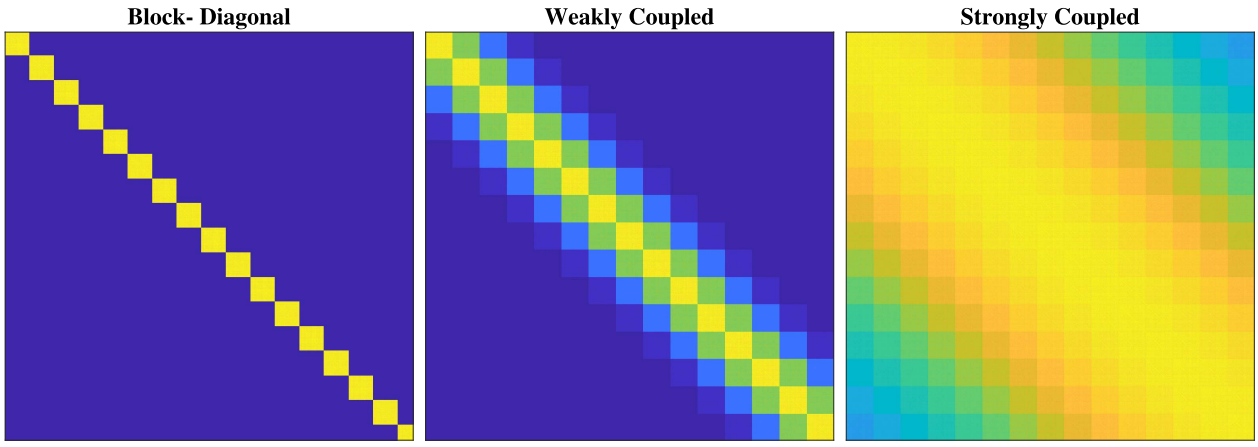


Fig. 2. Graphical interpretation of the resulting block-diagonal (left), weakly coupled (central), and strongly coupled (right) kernel Gram matrices. Dark color is used for matrix entries with smaller values and bright color is used for matrix entries with higher values.

all the output components. It is important to point out that by setting $\omega = 0$, the learning problem turns out to be equivalent to the block-diagonal formulation presented before.

As a tradeoff between the uncoupled and mixed kernel function, this article presents a separable kernel structure based on the product of standard radial basis function (RBF) kernels [29], such as

$$k_{x/o}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \exp\left(-\frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|}{\sigma_{x/o}}\right)^2 \tag{40}$$

where the pair $(\boldsymbol{\theta}, \boldsymbol{\theta}')$ can be any combination of input or output pairs and σ_x and σ_o are the hyperparameters of the scalar kernels k_x and k_o, respectively. Such hyperparameters are shared by all the output dimensions and can be tuned once via either cross validation or validation set [37], for instance, via Bayesian optimization [42].

The idea of using an RBF function for the kernel k_o acting on the output components can be seen as a tradeoff between a block-diagonal kernel and the mixed kernel. Indeed, a large value of σ_o will lead to a strong coupling among the output components, while a small value leads to a block-diagonal problem. For the sake of illustration, Fig. 2 shows a graphical interpretation of block-diagonal (left), weakly coupled (central), and strongly coupled (right) kernel Gram matrices.

Unfortunately, dealing with coupled kernel structures, such as the ones provided by the kernels in (38) and (40) is rather challenging. Indeed, in such cases, the model training requires the inversion of a fully coupled matrix $\mathbf{A}$ in (30) of dimension $LD \times LD$, for which the computational complexity scales as $\mathcal{O}(L^3D^3)$. This makes the direct inversion of the matrix $\mathbf{A}$ extremely inefficient or intractable in a standard laptop when the product between the number of training samples L and the output dimensionality D becomes in the order of thousands.

To overcome the above limitation, the linear system in (30) can be efficiently solved via an iterative procedure based on the gradient descent (GD) algorithm [25], [43]

$$\text{vec}(\mathbf{C})_k = \text{vec}(\mathbf{C})_{k-1} - \alpha\left[\mathbf{A}\,\text{vec}(\mathbf{C})_{k-1} - \text{vec}(\mathbf{Y})\right] \tag{41}$$

where $\text{vec}(\mathbf{C})_k$ represents the unknown regression coefficients estimated at the kth step and α is a scalar number, known as the learning rate, defining the step size at each iteration.

Specifically, the proposed modeling framework implements the conjugate gradient method [43], which provides an efficient version of the above algorithm tailored for semidefinite matrices, as the matrix $\mathbf{A}$ [29]. Such implementation allows reducing the computational complexity of the training phase from $\mathcal{O}(L^3D^3)$ to $\mathcal{O}(KL^2D^2)$, where K is the number of iterations required by the GD algorithm to converge. It is important to remark that due to the benefits in terms of computational cost of the GD algorithms with respect to the plain inversion algorithms, the above inversion scheme implemented in standard laptop allows to deal with regression problems in which $LD \leq 10$ k.

V. Nystrom Subsampling and Compression

This section presents a compression strategy based on the Nystrom approximation aimed at reducing the computational complexity arising from the training of the proposed vector-valued KRR. Unlike data compression strategies which act on the training dataset (e.g., PCA [28]), the proposed Nystrom compression performs directly on the Gram kernel matrix to be inverted during the model training [25], [33], [34], [35]. Such an approach will be here adopted to compress the empirical kernel matrix $\mathbf{K}_x$ in (30) from $L \times L$ to $n \times n$ with $n \leq L$. Possible further generalization and extension of the proposed compression scheme to the whole separable kernel structure will be investigated in future publications.

The Nystrom method or Nystrom approximation allows to approximate any positive semidefinite matrix, such as the kernel Gram matrix $\mathbf{K}_x$, by a smaller matrix collecting a subset of the columns of the original one. To this aim, let us define a subset $\tilde{\mathbf{X}}_n$ collecting $n \leq L$ input samples randomly chosen without replacement from the training set $\mathbf{X}$ such as

$$\tilde{\mathbf{X}}_n = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n]^T \tag{42}$$

where $n \leq L$.

According to the Nystrom approximation, the original Gram matrix $\mathbf{K}_{\boldsymbol{x}}$ can be reconstructed as follows [34], [35]:

$$\mathbf{K}_{\boldsymbol{x}} \approx \tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{K}}_{nn}^{-1}\tilde{\mathbf{K}}_{nL} \tag{43}$$

where $\tilde{\mathbf{K}}_{nn} = k_{\boldsymbol{x}}(\tilde{\mathbf{X}}_n, \tilde{\mathbf{X}}_n)$ is an $n \times n$ symmetric Gram matrix such as $[\tilde{\mathbf{K}}_{nn}]_{ij} = k_{\boldsymbol{x}}(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j)$ for $i, j = 1, \ldots, n$ and $\tilde{\mathbf{K}}_{Ln} = \tilde{\mathbf{K}}_{nL}^T = k_{\boldsymbol{x}}(\mathbf{X}, \tilde{\mathbf{X}}_n)$ is an $L \times n$ rectangular matrix such as $[\tilde{\mathbf{K}}_{Ln}]_{ij} = k_{\boldsymbol{x}}(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_j)$ for $i = 1, \ldots, L$ and $j = 1, \ldots, n$.

Several advanced algorithms have been presented for the selection of the number of configurations n in the reduced subset $\tilde{\mathbf{X}}_n$ in (42) (see, e.g., [25], [34], [35]). However, in our implementation of the Nystrom method, the parameter n is iteratively increased until the relative error of the approximation in (43) is less than a given tolerance

$$\frac{\|\mathbf{K}_{\boldsymbol{x}} - \tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{K}}_{nn}^{-1}\tilde{\mathbf{K}}_{nL}\|_F}{\|\mathbf{K}_{\boldsymbol{x}}\|_F} \times 100 \leq \varepsilon_{\%}. \tag{44}$$

Hereafter, in this article, $\varepsilon_{\%}$ has been set to 0.1%.

By using the Nystrom approximation of the Gram kernel matrix in (43), the ERM in (27) can be written as

$$\min_{\tilde{\mathbf{C}}} \|\mathbf{Y} - \tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{C}}\mathbf{B}\|_F^2 + \lambda\langle\tilde{\mathbf{C}}^T\tilde{\mathbf{K}}_{nn}\tilde{\mathbf{C}}, \mathbf{B}\rangle_F \tag{45}$$

where in this case $\tilde{\mathbf{C}} \in \mathbb{R}^{n\times D}$ is a compressing matrix collecting row-by-row the unknown vectors $[\tilde{\boldsymbol{c}}_1, \ldots, \tilde{\boldsymbol{c}}_n]^T$, with $\tilde{\boldsymbol{c}}_n \in \mathbb{R}^D$.

Similar to what has been done in Section III-C, the above convex optimization problem can be solved by setting to zero the partial derivatives of the cost function computed with respect to the unknown matrix $\tilde{\mathbf{C}}$. Again, after some calculations reported in Appendix C, the solution we are looking for can be written in terms of the following generalized discrete-time Sylvester equation:

$$-\tilde{\mathbf{K}}_{nL}\mathbf{Y} + \tilde{\mathbf{K}}_{nL}\tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{C}}\mathbf{B} + \lambda\tilde{\mathbf{K}}_{nn}\tilde{\mathbf{C}} = \mathbf{0} \tag{46}$$

where the matrix $\tilde{\mathbf{C}}$ can be computed by solving the following linear system [41]:

$$\underbrace{\left(\mathbf{B} \otimes \tilde{\mathbf{K}}_{nL}\tilde{\mathbf{K}}_{Ln} + \lambda\mathbf{I}_D \otimes \tilde{\mathbf{K}}_{nn}\right)}_{\tilde{\mathbf{A}}} \operatorname{vec}\left(\tilde{\mathbf{C}}\right) = \operatorname{vec}\left(\tilde{\mathbf{K}}_{nL}\mathbf{Y}\right) \tag{47}$$

where now the matrix $\tilde{\mathbf{A}}$ is an $nD \times nD$ matrix.

The model prediction for a generic test point $\boldsymbol{x}$ can be written as

$$\hat{f}(\boldsymbol{x}) = \sum_{l=1}^{n} k_{\boldsymbol{x}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}_l)\mathbf{B}\tilde{\boldsymbol{c}}_l = \mathbf{B} \otimes k_{\boldsymbol{x}}\left(\boldsymbol{x}, \tilde{\mathbf{X}}_n\right) \operatorname{vec}\left(\tilde{\mathbf{C}}\right) \tag{48}$$

where $\operatorname{vec}(\tilde{\mathbf{C}}) \in \mathbb{R}^{(nD)\times 1}$ and $k_{\boldsymbol{x}}(\boldsymbol{x}, \tilde{\mathbf{X}}_n) \in \mathbb{R}^{1\times n}$.

It is important to notice that due to the Nystrom compression, the number of unknowns to be estimated during the model training turns out to be $n \times D$ with $n \leq L$, while the dimensionality of the original uncompressed matrix $\mathbf{A}$ is $L \times D$. Therefore, the training cost for inverting the matrix $\tilde{\mathbf{A}}$ with the GD reduces from $\mathcal{O}(KL^2D^2)$ to $\mathcal{O}(n \cdot L + Kn^2D^2)$ for the full coupled kernel and to $\mathcal{O}(n \cdot L + Kn^2)$ for the block-diagonal kernel function. The Nystrom compression strategy presented in this section will be used hereafter in this article to constrain the size of the Gram kernel matrix $\mathbf{K}$ to be less than 10k $\times$ 10k.

TABLE I

MEAN VALUE AND CORRESPONDING RELATIVE RANGE OF VARIATION OF THE 11 PARAMETERS CONSIDERED FOR THE ILLUSTRATIVE EXAMPLE IN SECTION VI

Parameter	Mean Value	Uniform Variation
$C_1(x_1)$	1 pF	20%
$C_2(x_2)$	0.5 pF	20%
$L_1(x_3)$	10 nH	20%
$L_2(x_4)$	10 nH	20%
$\varepsilon_r(x_5)$	4.1	1%
$w(x_6)$	252 μm	1%
$t(x_7)$	35 μm	1%
$h(x_8)$	60 μm	1%
$Len_1(x_9)$	5 cm	5%
$Len_2(x_{10})$	3 cm	5%
$Len_3(x_{11})$	3 cm	5%

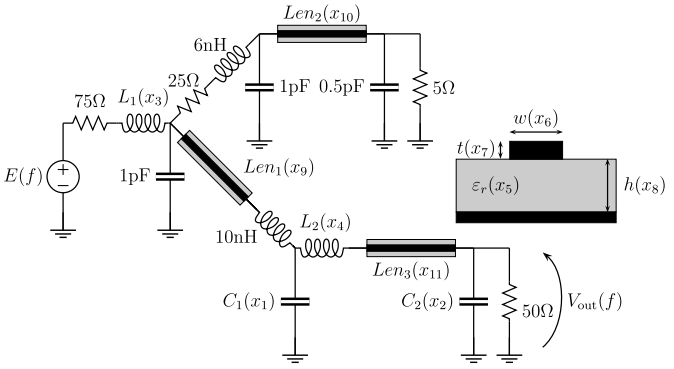


Fig. 3. Schematic of the high-speed link considered as illustrative example in Section VI.

VI. ILLUSTRATIVE EXAMPLE

This section provides a more practical interpretation of the mathematical formulation presented in Sections III–V by means of an illustrative example, with the aim of discussing the advantages and drawbacks of the proposed vector-valued KRR. Without loss of generality, the proposed results will focus on the high-speed link in Fig. 3.

Specifically, the proposed vector-valued KRR is applied to predict the parametric behavior of the magnitude of the frequency response $y(\boldsymbol{x}; f) = |H(f; \boldsymbol{x})| = |V_{\text{out}}(f; \boldsymbol{x})/E(f)|$, as a function of 11 normalized parameters collected in the vector $\boldsymbol{x} = [x_1, \ldots, x_{11}]^T$, in which each parameter $x_i \sim \mathcal{U}([-1, +1])$ is modeled as a normalized uniformly distributed random variable. Additional details about the variability and mean value of the 11 parameters are provided in Table I. The high-speed link has been implemented by means of a parametric simulation in MATLAB. Such implementation is then used to generate the training, validation, and test sets based on a Latin hypercube sampling (LHS).

The performance of the proposed KRR is investigated on three different configurations of the proposed test case.

1) *CASE A:* Noise-free training set in a frequency band from 1 MHz to 2 GHz.
2) *CASE B:* Noisy training set[2] in a frequency band from 1 MHz to 2 GHz.

[2]Uniformly distributed and uncorrelated noise terms affecting the real and imaginary parts of the frequency response $H(f; \boldsymbol{x})$ with an absolute level of 0.05.

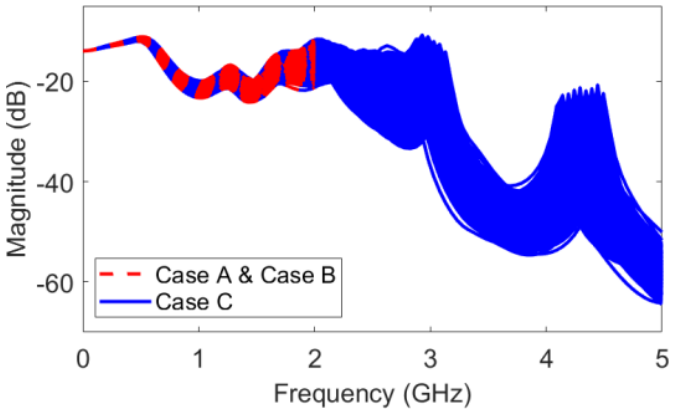


Fig. 4. Parametric behavior of the magnitude of transfer function $y(\boldsymbol{x}; f)$ of the high-speed link in Fig. 3 computed on 1000 test samples for CASEs A, B, and C.

3) *CASE C:* Noise-free training set in a wider frequency band from 1 MHz to 5 GHz.

For each of the above configurations, the parametric behavior of the magnitude of the frequency response $H(f; \boldsymbol{x})$ is investigated for 100 equally spaced frequency points (i.e., the number of outputs is $D = 100$). For the sake of illustration, Fig. 4 shows the spread of 1000 realizations (used as test samples) of the frequency responses for the three considered configurations: CASEs A, B, and C. The plots highlight the complexity of the three datasets, as well as the strong sensitivity of the model output to the considered parameters.

Two vector-valued KRRs with a block-diagonal and a coupled kernel are trained with L training samples. A validation set [37] with 150 samples is used within a Bayesian optimization [42] to tune the regression hyperparmaters by considering the following intervals: $\sigma_{\boldsymbol{x}} = [10^{-2}, 10^{2}]$, $\sigma_o = [10^{-4}, 10^{-2}]$, and $\lambda = [10^{-3}, 10^{-1}]$ for the coupled kernel matrix and $\sigma_{\boldsymbol{x}} = [10^{-2}, 10^{2}]$, $\sigma_o = [10^{-11}, 2 \times 10^{-11}]$, and $\lambda = [10^{-5}, 10^{-2}]$ for the block-diagonal one.

Fig. 5 shows a comparison between the predictions obtained by the proposed vector-valued KRR trained with a block-diagonal and coupled kernel matrix (see Fig. 2) for three different configurations of the input parameters $\boldsymbol{x}$ and the corresponding scatter plots computed on the 1000 test samples. The comparison highlights the excellent capability of trained models to capture the actual variation of the transfer function under modeling for the three considered test-case configurations. Moreover, Table II presents a quantitative comparison among the proposed implementations in terms of training time t_{train} and relative L_2- and L_∞-error computed in a linear scale from the predictions in decibels provided by the proposed models on 1000 test samples. The figures of merit provided in the table lead to the following observations.

1) *Training Time:* As shown in the rows labeled with t_{train} in Table II, the computational cost for the training of the vector-valued KRR with coupled kernel is higher than the one required by the block-diagonal implementation. Indeed, as discussed in Section IV, the computational complexity of the model training depends on the structure of the matrix $\mathbf{A}$ to be inverted in (30), and it is proportional to $\mathcal{O}(KL^2D^2)$ for the implementation of the proposed vector-valued KRR with a coupled kernel and reduces to $\mathcal{O}(KL^2)$ for the uncoupled one.
2) *Model Accuracy:* The KRR implementation based on the block-diagonal kernel provides the most accurate model for CASEs A and C with an L_2-error below 5%. The high value of the L_∞-error (i.e., the worst case error) for the CASE C is motivated by the inherently resonance behavior of the frequency response under modeling in the considered frequency bandwidth. On the other hand, the results for CASE B highlight the benefits of the regularization effect on the output dimension introduced by the coupled kernel [32]. Such regularization allows to suppress the sharp fluctuations induced by the noise, thus leading to a more accurate prediction on the noiseless test set.

TABLE II

COMPARISON OF TRAINING TIME t_{TRAIN} AND RELATIVE L_2- AND L_∞-ERROR COMPUTED FOR THE COUPLED AND UNCOUPLED KERNEL IMPLEMENTATION OF THE PROPOSED VECTOR-VALUED KRR. THE STUDY WAS CONDUCTED ON THE ILLUSTRATIVE EXAMPLE OF FIG. 3, FOR 1000 TEST SAMPLES

Kernel Matrix	Error	Case A ($L = 150$)	Case B ($L = 150$)	Case C ($L = 300$)
Block-Diagonal Kernel (see Sec. IV-A)	L_2	0.43%	5.77%	4.1%
	L_∞	15.69%	31.66%	35%
	t_{train}	13s	16s	23s
Coupled Kernel (see Sec. IV-B)	L_2	1.32%	2.54%	6.7%
	L_∞	11.51%	16.74%	48%
	t_{train}	812s	1547s	1527s

TABLE III

MEAN VALUE AND CORRESPONDING RELATIVE RANGE OF VARIATION OF THE PARAMETERS CONSIDERED FOR THE OPTIMIZATION OF THE DOHERTY AMPLIFIER IN SECTION VII

Parameter	Mean Value	Uniform Variation
W_{TL1}	29.44 mil	50%
W_{TL2}	50.78 mil	50%
W_{TL3}	29.44 mil	50%
W_{TL4}	29.44 mil	50%
W_{TL5}	50.78 mil	50%
W_{TL6}	29.44 mil	50%
W_{TL7}	29.44 mil	50%
W_{TL8}	29.44 mil	50%
L_{TL1} & L_{TL2}	646.85 mil	2.5%
L_{TL4} & L_{TL6}	620.9 mil	2.5%
L_{TL3} & L_{TL5}	646.85 mil	2.5%
L_{TL7} & L_{TL8}	646.85 mil	2.5%
V_{dc1}	2.45 v	5%
V_{dc2}	7 v	5%
V_{dc3} & V_{dc4}	28 v	5%

Summarizing, the block-diagonal kernel provides the best tradeoff between efficiency and accuracy for noiseless multi-output regression problems, but it is also extremely sensitive to noise. Indeed, the block-diagonal formulation does not directly account for a possible correlation among the output dimensions (i.e., the frequency points of the frequency response), thus increasing the model variance and leading to overfitting issue in the output space. On the contrary, the coupled formulation introduces a regularization effect on

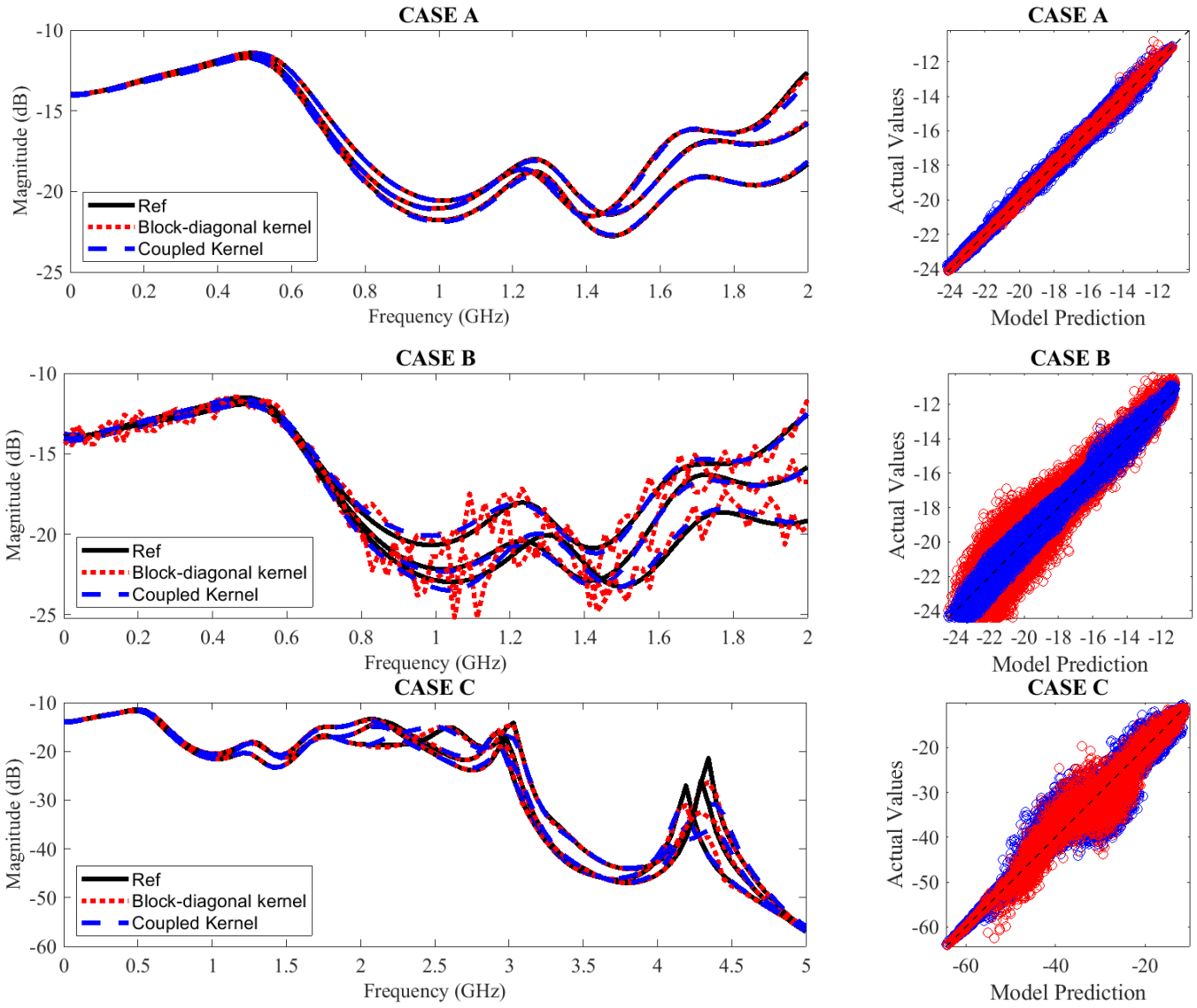


Fig. 5. Parametric and scatter plots comparing the prediction of the proposed vector-valued KRR with block-diagonal and coupled kernels with the corresponding ones obtained from the computational model for CASEs A, B, and C on 1000 test samples.

the output dimensions, leading to a smoother model in the output space able to heavily suppress noise fluctuations. It is important to remark that, despite the Nystrom compression and the GD algorithm proposed in this work, alternative inversion approach will be further investigated in future works to speed up the Gram matrix inversion, as an example, by exploring the mathematical structure of the proposed separable kernel [32].

VII. Application Example: Doherty Amplifier

This section discusses the performance of the proposed method by considering the optimization of the power splitter of the Doherty amplifier shown in Fig. 6 [44]. Specifically, the proposed vector-valued KRR is used to train a parametric model able to predict the S_{11} and S_{21} of the amplifier, as a function of several coupled and uncoupled parameters listed in Table III, characterizing the working point of the amplifier and the geometry of the power splitter (see the red square in Fig. 6).

First, the schematic in Fig. 6 has been implemented as a parametric simulation in ADS. For any configuration of the input parameters, the ADS simulation provides the frequency responses of the scattering parameters S_{11} and S_{21} computed for $D = 1101$ frequency points in a bandwidth from 1.9 to 3 GHz. A set of $L = 700$ training samples and 100 validation samples have been generated via an LHS.

Such samples have been used to train a parametric model for the scattering parameters of interest via the KRR with the block-diagonal kernel. The model training takes 220 s. The obtained models are then used together with a "brute-force" optimization algorithm based on a random grid search [45] implemented in MATLAB, with the aim of optimizing the amplifier parameters in order to meet the following constraints:

$$S_{11} \leq -10 \text{ dB for } 2.4 \text{ GHz} \leq f \leq 2.6 \text{ GHz} \quad (49a)$$

$$10 \text{ dB} \leq S_{21} \leq 12 \text{ dB for } 2.1 \text{ GHz} \leq f \leq 2.9 \text{ GHz}. \quad (49b)$$

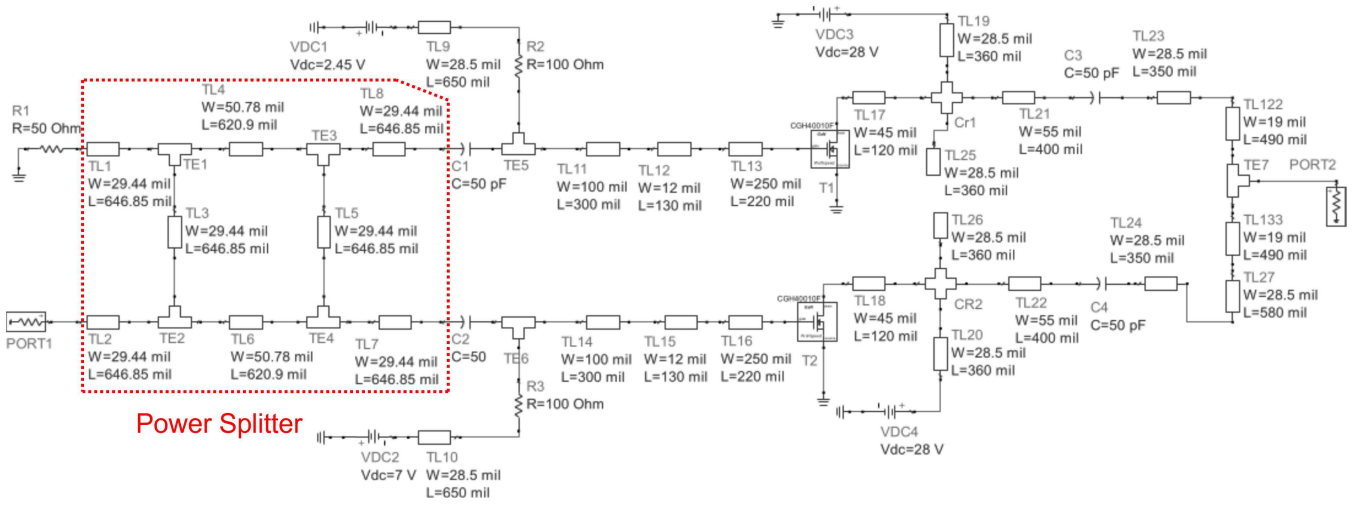


Fig. 6. Schematic of the Doherty amplifier considered in Section VII (inspired by [44]).

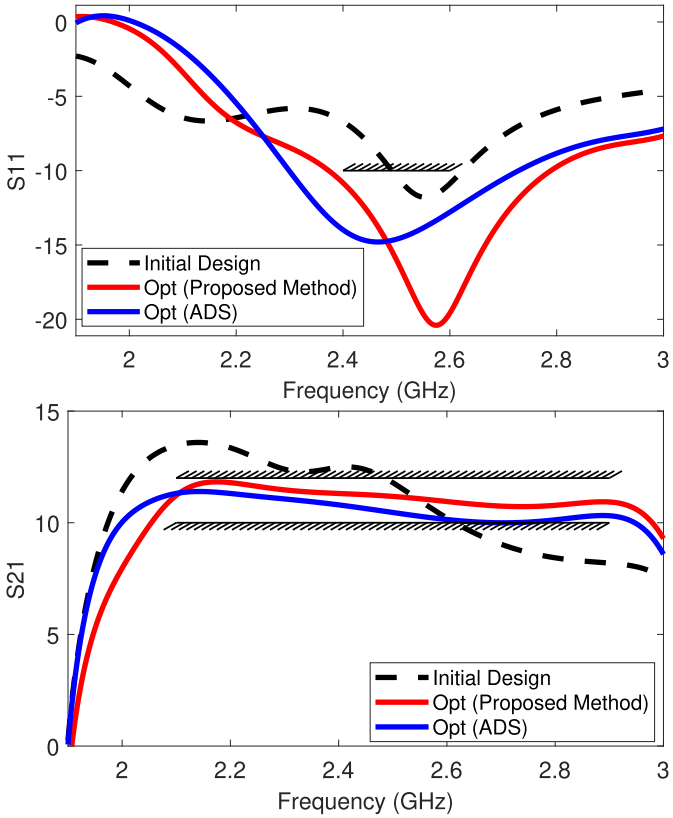


Fig. 7. Scattering parameters of the Doherty amplifier presented in Section VII obtained from the initial design (black dashed line) and after optimization carried out via the ADS random optimizer (red solid line) and the proposed model (blue solid line).

Fig. 7 compares the S_{11} and S_{21} scattering parameters estimated after the optimization via the proposed vector-valued model with those obtained from the initial design. Moreover, the plots show the corresponding results obtained from the random optimizer (default option) in ADS after 2800 iterations. The results clearly highlight the strong agreement and consistency between the optimization results obtained via the proposed modeling scheme and the ones obtained from the ADS optimization. Concerning the computational cost, the overall optimization with our advocated model takes 25 s. On the other hand, the corresponding optimization in ADS requires 2800 iterations and takes 386 s. The proposed simulation approach leads to a speedup 15× with respect to ADS. It is important to stress that the obtained speedup is mitigated by the relatively fast simulation time required by the ADS circuital solver when it is used in a small-signal analysis. Moreover, unlike the ADS optimizer, the obtained model turns out to be independent of the optimization constraints and therefore can be suitably adopted as it is to meet different optimization constraints, as well as for the stochastic analysis within the uncertainty quantification scenario [13], [29].

VIII. Conclusion

This article presented a generalized vector-valued formulation of the KRR, able to deal with the inherently multioutput nature shared by most of the microwave applications. The proposed vector-valued KRR can be seen as a generalization of the mathematical framework used by state-of-the-art scalar kernel regressions. The mathematical formulation has been discussed in detail, also providing several alternatives for the kernel functions and training schemes. Moreover, a compression strategy based on the Nystrom approximation has been proposed with the aim of mitigating the computational complexity of the training phase. The feasibility and the performance of the proposed approach have been investigated on an illustrative example consisting of a high-speed link and for the optimization of a Doherty amplifier.

Appendix A
From Basis Expansion to Scalar-Output KRR

We consider a set of training pairs $\mathcal{S} = \{(\boldsymbol{x}_l, y_l)\}_{l=1}^{L}$, where $\boldsymbol{x}_l \in \mathcal{X} \subseteq \mathbb{R}^p$ represents the training input samples and $y_l \in \mathcal{Y} \subseteq \mathbb{R}$ are the corresponding scalar outputs. We seek a linear model $\tilde{f}$ defined as a standard basis expansion, such

as [36], [38], [39]

$$\tilde{f}(\boldsymbol{x}) = \sum_{k=1}^{P} w_k \phi_k(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}) \rangle \tag{50}$$

where $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \ldots, \phi_P(\boldsymbol{x})]^T$ is a vector collecting the basis functions $\phi_i(\boldsymbol{x})$ such that $\boldsymbol{\phi}(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^P$ provides a nonlinear map between the p-dimensional parameter space and the P-dimensional feature space, $\boldsymbol{w} = [w_1, \ldots, w_P]^T$ is a vector collecting the regression coefficients, and $\langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}) \rangle$ is the inner product in the Hilbert space (i.e., $\langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}) \rangle = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$ for a feature space with finite dimension).

The regression coefficients in the vector $\boldsymbol{w}$ are estimated during the training phase by solving the following ERM:

$$\min_{\boldsymbol{w}} \sum_{l=1}^{L} (y_l - \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}_l) \rangle)^2 + \lambda \|\boldsymbol{w}\|^2 \tag{51}$$

for $\lambda \geq 0$.

The above optimization problem can be rewritten in its matrix form as

$$\min_{\boldsymbol{w}} \left(\boldsymbol{y} - \boldsymbol{\Phi}^T \boldsymbol{w}\right)^T \left(\boldsymbol{y} - \boldsymbol{\Phi}^T \boldsymbol{w}\right) + \lambda \boldsymbol{w}^T \boldsymbol{w} \tag{52}$$

where $\boldsymbol{\Phi}^T \in \mathbb{R}^{L \times P}$ is a matrix collecting the basis functions evaluated on the training inputs

$$\boldsymbol{\Phi}^T = [\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_L)]^T \tag{53}$$

and $\boldsymbol{y} = [y_1, \ldots, y_L]^T \in \mathbb{R}^L$ is a vector collecting the training outputs. According to the above definition, the lth training output is approximated as

$$y_l \approx \left[\boldsymbol{\Phi}^T \boldsymbol{w}\right]_l = \boldsymbol{\phi}(\boldsymbol{x}_l)^T \boldsymbol{w}. \tag{54}$$

After some straightforward calculation, the cost function in (52) can be written as

$$\begin{aligned} E(\boldsymbol{w}) &= \boldsymbol{y}^T \boldsymbol{y} - \boldsymbol{w}^T \boldsymbol{\Phi} \boldsymbol{y} - \boldsymbol{y}^T \boldsymbol{\Phi}^T \boldsymbol{w} + \boldsymbol{w}^T \boldsymbol{\Phi} \boldsymbol{\Phi}^T \boldsymbol{y} + \lambda \boldsymbol{w}^T \boldsymbol{w} \\ &= \boldsymbol{y}^T \boldsymbol{y} - 2\boldsymbol{w}^T \boldsymbol{\Phi} \boldsymbol{y} + \boldsymbol{w}^T \boldsymbol{\Phi} \boldsymbol{\Phi}^T \boldsymbol{y} + \lambda \boldsymbol{w}^T \boldsymbol{w} \end{aligned} \tag{55}$$

since $\boldsymbol{y}^T \boldsymbol{\Phi}^T \boldsymbol{w} = (\boldsymbol{y}^T \boldsymbol{\Phi}^T \boldsymbol{w})^T$. The above cost function can be minimized by calculating its partial derivatives with respect to the regression unknowns $\boldsymbol{w}$

$$\frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} = -2\boldsymbol{\Phi} \boldsymbol{y} + 2\boldsymbol{\Phi} \boldsymbol{\Phi}^T \boldsymbol{w} + 2\lambda \boldsymbol{w} \tag{56}$$

where we are using the following property of the derivatives with vectors:

$$\frac{\partial \boldsymbol{x}^T \mathbf{A} \boldsymbol{x}}{\partial \boldsymbol{x}} = 2\mathbf{A}\boldsymbol{x}. \tag{57}$$

The optimal values of $\boldsymbol{w}$ can be found as

$$\frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} = \mathbf{0} \rightarrow \left(\boldsymbol{\Phi} \boldsymbol{\Phi}^T + \lambda \mathbf{I}_P\right) \boldsymbol{w} - \boldsymbol{\Phi} \boldsymbol{y} = \mathbf{0} \tag{58}$$

which gives the well-known solution for the KRR based on the pseudoinverse matrix

$$\boldsymbol{w} = \left(\boldsymbol{\Phi} \boldsymbol{\Phi}^T + \lambda \mathbf{I}_P\right)^{-1} \boldsymbol{\Phi} \boldsymbol{y} \tag{59}$$

where $\mathbf{I}_P \in \mathbb{R}^{P \times P}$ is the identity matrix and $\boldsymbol{\Phi} \boldsymbol{\Phi}^T \in \mathbb{R}^{P \times P}$. It is important to notice that the overall number of unknowns in (59) is P (i.e., the number of basis functions).

However, it is possible to prove that if $\boldsymbol{\Phi} \boldsymbol{\Phi}^T$ is symmetric, (59) admits its direct dual solution [39] such that

$$\boldsymbol{w} = \left(\boldsymbol{\Phi} \boldsymbol{\Phi}^T + \lambda \mathbf{I}_P\right)^{-1} \boldsymbol{\Phi} \boldsymbol{y} = \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \mathbf{I}_L\right)^{-1} \boldsymbol{y} \tag{60}$$

where, in this case, $\mathbf{I}_L$ is an $L \times L$ identity matrix since $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ is an $L \times L$ matrix.

Let us now define the vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_L]^T$ as

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I}_L)^{-1} \boldsymbol{y} \tag{61}$$

where the matrix $\mathbf{K} = \boldsymbol{\Phi}^T \boldsymbol{\Phi} \in \mathbb{R}^{L \times L}$ is the so-called Gram matrix [38]. Therefore, the original unknown vector $\boldsymbol{w}$ in (59) can be written as

$$\boldsymbol{w} = \boldsymbol{\Phi} \boldsymbol{\alpha} = \sum_{l=1}^{L} \boldsymbol{\phi}(\boldsymbol{x}_l) \alpha_l. \tag{62}$$

Now, we can focus on the Gram matrix $\mathbf{K}$, which is defined as

$$\mathbf{K} = \boldsymbol{\Phi}^T \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1)^T \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}_L)^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1) & \ldots & \boldsymbol{\phi}(\boldsymbol{x}_L) \end{bmatrix} \tag{63}$$

such that the ij-element of the matrix $\mathbf{K}$ can be written as

$$[\mathbf{K}]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}(\boldsymbol{x}_i)^T \boldsymbol{\phi}(\boldsymbol{x}_j) = \langle \boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{\phi}(\boldsymbol{x}_j) \rangle_{\mathcal{H}} \tag{64}$$

where $k(\cdot, \cdot) : \mathbb{R}^{p \times p} \rightarrow \mathbb{R}$ is a kernel function, defined as the inner product in the Hilbert space between the basis functions. According to (63), the Gram matrix $\mathbf{K}$ collects the kernel function evaluated on the training inputs.

By substituting (62) into (50), for any $\boldsymbol{x} \in \mathcal{X}$, we have

$$\begin{aligned} \tilde{f}(\boldsymbol{x}) &= \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}) \rangle = \sum_{l=1}^{L} \alpha_l \langle \boldsymbol{\phi}(\boldsymbol{x}_l), \boldsymbol{\phi}(\boldsymbol{x}) \rangle \\ &= \sum_{l=1}^{L} \alpha_l k(\boldsymbol{x}_l, \boldsymbol{x}). \end{aligned} \tag{65}$$

Equation (65) is the dual formulation of the ridge regression [38], [39]. It is important to notice that the resulting model is completely defined by L unknowns, where L is equal to the number of training samples. Indeed, due to the kernel function, the number of regression unknowns is completely independent of the number of basis functions collected in the vector $\boldsymbol{\phi}$ in (50). In principle, we do not need to explicitly define the basis functions, and we just need to know the corresponding kernel, leading to the so-called kernel trick [22]. In principle, since the kernel is defined as the inner product in the Hilbert space, we can even work in an infinite-dimensional space (i.e., $P \rightarrow \infty$) [22], [23].

APPENDIX B
DERIVATION OF THE VECTOR-OUTPUT KRR

Let us recall the matrix formulation of the ERM for the vector-valued KRR with a separable matrix kernel, which can be written as

$$\min_{\mathbf{C}} \|\mathbf{Y} - \mathbf{K}_{\boldsymbol{x}} \mathbf{C} \mathbf{B}\|_F^2 + \lambda \|\hat{\boldsymbol{f}}\|_{\mathcal{H}}^2 \tag{66}$$

where

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{l=1}^{L} k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}_l)\mathbf{B}\boldsymbol{c}_l. \tag{67}$$

In the above ERM, the regularizer term $\|\hat{\boldsymbol{f}}\|_{\mathcal{H}}^2$ can be rewritten in terms of a Frobenius inner product [40], [46]

$$\begin{aligned}\|\hat{\boldsymbol{f}}\|_F^2 &= \sum_{i,j=1}^{L} \langle \boldsymbol{c}_i, \boldsymbol{B}k_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j)\boldsymbol{c}_j\rangle \\ &= \sum_{i,j=1}^{L} \boldsymbol{c}_i^T \boldsymbol{B}k_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j)\boldsymbol{c}_j \\ &= \mathrm{Tr}\big(\mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\big) = \langle \mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}, \mathbf{B}\rangle_F \end{aligned} \tag{68}$$

in which we have used the following properties of the trace operator:

$$\sum_l \sum_k A_{lk}\boldsymbol{w}_l^T\boldsymbol{w}_k = \mathrm{Tr}\big(\mathbf{W}\mathbf{A}\mathbf{W}^T\big) \tag{69a}$$

$$\mathrm{Tr}(\mathbf{A}\mathbf{B}) = \mathrm{Tr}\big((\mathbf{A}\mathbf{B})^T\big) = \mathrm{Tr}\big(\mathbf{B}^T\mathbf{A}^T\big) \tag{69b}$$

$$\mathrm{Tr}\left(\sum_i \mathbf{A}_i\right) = \sum_i \mathrm{Tr}(\mathbf{A}_i) \tag{69c}$$

together with the definition of the Frobenius inner product in (28).

Therefore, the optimization problem in (66) can be rewritten as

$$\min_{\mathbf{C}} \|\mathbf{Y} - \mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\|_F^2 + \lambda\langle \mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}, \mathbf{B}\rangle_F. \tag{70}$$

Now, we can write $\|\mathbf{Y} - \mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\|_F^2$ as

$$\begin{aligned}&\|\mathbf{Y} - \mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\|_F^2 \\ &= \mathrm{Tr}\big((\mathbf{Y} - \mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B})^T(\mathbf{Y} - \mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B})\big) = \\ &= \mathrm{Tr}\big(\mathbf{Y}^T\mathbf{Y} - (\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B})^T\mathbf{Y} - \mathbf{Y}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B} + (\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B})^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\big) \\ &= \mathrm{Tr}\big(\mathbf{Y}^T\mathbf{Y} - \mathbf{B}\mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{Y} - \mathbf{Y}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B} + \mathbf{B}\mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\big) \\ &= \mathrm{Tr}\big(\mathbf{Y}^T\mathbf{Y} - 2\mathbf{B}\mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{Y} + \mathbf{B}\mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\big)\end{aligned} \tag{71}$$

since

$$\mathrm{Tr}\big(\mathbf{Y}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\big) = \mathrm{Tr}\Big(\big(\mathbf{Y}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\big)^T\Big) = \mathrm{Tr}\big(\mathbf{B}\mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{Y}\big). \tag{72}$$

Therefore, the optimization problem in (66) can be written as

$$\min_{\mathbf{C}} \mathrm{Tr}\big(\mathbf{Y}^T\mathbf{Y} - 2\mathbf{B}\mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{Y} \tag{73}$$

$$+\mathbf{B}\mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B} + \lambda\mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\big) = E(\mathbf{C}). \tag{74}$$

In the above minimization problem, the optimal values of the entries of coefficient matrix $\mathbf{C}$ are estimated as the ones for which

$$\frac{\partial E(\mathbf{C})}{\partial \mathbf{C}} = \mathbf{0}. \tag{75}$$

In order to compute the above partial derivative, let us recall some properties of the trace and derivative operator [41]

$$\frac{\mathrm{Tr}\big(\mathbf{X}^T\mathbf{A}\big)}{\partial \mathbf{X}} = \mathbf{A} \tag{76a}$$

$$\frac{\mathrm{Tr}\big(\mathbf{A}\mathbf{X}^T\mathbf{B}\big)}{\partial \mathbf{X}} = \mathbf{B}\mathbf{A} \tag{76b}$$

$$\frac{\mathrm{Tr}\big(\mathbf{B}^T\mathbf{X}^T\mathbf{C}\mathbf{X}\mathbf{B}\big)}{\partial \mathbf{X}} = \mathbf{C}^T\mathbf{X}\mathbf{B}\mathbf{B}^T + \mathbf{C}\mathbf{X}\mathbf{B}\mathbf{B}^T. \tag{76c}$$

According to the above relationships, (75) can be written as

$$\frac{\partial E(\mathbf{C})}{\partial \mathbf{C}} = -2\mathbf{K}_{\boldsymbol{x}}\mathbf{Y}\mathbf{B} + 2\mathbf{K}_{\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\mathbf{B} + 2\lambda\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B} = \mathbf{0} \tag{77}$$

from which we get

$$\mathbf{K}_{\boldsymbol{x}}(-\mathbf{Y} + \mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B} + \lambda\mathbf{C})\mathbf{B} = \mathbf{0} \tag{78}$$

which leads to the following discrete-time Sylvester equation [40]:

$$\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B} + \lambda\mathbf{C} = \mathbf{Y}. \tag{79}$$

Equation (79) can be solved in a closed form by using the Kronecker formulation [41] such that [40]

$$(\mathbf{B} \otimes \mathbf{K}_{\boldsymbol{x}} + \lambda\mathbf{I}_{DL})\,\mathrm{vec}(\mathbf{C}) = \mathrm{vec}(\mathbf{Y}). \tag{80}$$

After permutations, (80) is equivalent to the one reported in [46], which can be written as

$$(\mathbf{K}_{\boldsymbol{x}} \otimes \mathbf{B} + \lambda\mathbf{I}_{LD})\,\mathrm{vec}\big(\mathbf{C}^T\big) = \mathrm{vec}\big(\mathbf{Y}^T\big) \tag{81}$$

where, in this case, the vectors $\boldsymbol{c}_l$ in (67) are the columns of the matrix $\mathbf{C}^T$ that can be reconstructed from $\mathrm{vec}(\mathbf{C}^T)$.

Appendix C
Derivation of the Nystrom Compressed Vector-Output KRR

Let us recall the matrix form of the ERM obtained from the Nystrom approximation of the Gram kernel matrix in (43)

$$\min_{\tilde{\mathbf{C}}} \|\mathbf{Y} - \tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{C}}\mathbf{B}\|_F^2 + \lambda\langle \tilde{\mathbf{C}}^T\tilde{\mathbf{K}}_{nn}\tilde{\mathbf{C}}, \mathbf{B}\rangle_F \tag{82}$$

where, in this case, $\tilde{\mathbf{C}} \in \mathbb{R}^{nD\times nD}$.

By expanding the term $\|\mathbf{Y} - \mathbf{K}_{Ln}\tilde{\mathbf{C}}\mathbf{B}\|_F^2$, we get

$$\begin{aligned}&\|\mathbf{Y} - \tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{C}}\mathbf{B}\|_F^2 \\ &\quad= \mathrm{Tr}\Big(\big(\mathbf{Y} - \tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{C}}\mathbf{B}\big)^T\big(\mathbf{Y} - \tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{C}}\mathbf{B}\big)\Big) \\ &\quad= \mathrm{Tr}\big(\mathbf{Y}^T\mathbf{Y} - 2\mathbf{B}\tilde{\mathbf{C}}^T\tilde{\mathbf{K}}_{nL}\mathbf{Y} + \mathbf{B}\tilde{\mathbf{C}}^T\tilde{\mathbf{K}}_{nL}\tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{C}}\mathbf{B}\big).\end{aligned} \tag{83}$$

Therefore, (82) can be written as

$$\begin{aligned}\min_{\tilde{\mathbf{C}}} \mathrm{Tr}\big(&\mathbf{Y}^T\mathbf{Y} - 2\mathbf{B}\tilde{\mathbf{C}}^T\tilde{\mathbf{K}}_{nL}\mathbf{Y} \\ &+ \mathbf{B}\tilde{\mathbf{C}}^T\tilde{\mathbf{K}}_{nL}\tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{C}}\mathbf{B} + \lambda\tilde{\mathbf{C}}^T\tilde{\mathbf{K}}_{nn}\tilde{\mathbf{C}}\mathbf{B}\big) = E\big(\tilde{\mathbf{C}}\big).\end{aligned} \tag{84}$$

Again, the cost function $E(\tilde{\mathbf{C}})$ is minimized by setting to zeros its partial derivatives with respect to $\tilde{\mathbf{C}}$, i.e., $(\partial E(\tilde{\mathbf{C}})/\partial\tilde{\mathbf{C}}) = \mathbf{0}$. By using the properties of the trace and the derivative operator in (76a), (76b) and (76c), (84) turns out to be equivalent to the following linear system of equation:

$$-2\tilde{\mathbf{K}}_{nL}\mathbf{Y}\mathbf{B} + 2\tilde{\mathbf{K}}_{nL}\tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{C}}\mathbf{B}\mathbf{B} + 2\lambda\tilde{\mathbf{K}}_{nn}\tilde{\mathbf{C}}\mathbf{B} = \mathbf{0} \tag{85}$$

which also in this case leads to the following discrete-time generalized Sylvester equation:

$$-\tilde{\mathbf{K}}_{nL}\mathbf{Y} + \tilde{\mathbf{K}}_{nL}\tilde{\mathbf{K}}_{Ln}\tilde{\mathbf{C}}\mathbf{B} + \lambda\tilde{\mathbf{K}}_{nn}\tilde{\mathbf{C}} = \mathbf{0} \tag{86}$$

for which the coefficients $\tilde{\mathbf{C}}$ can be computed by solving the following linear system:

$$\big(\mathbf{B} \otimes \tilde{\mathbf{K}}_{nL}\tilde{\mathbf{K}}_{Ln} + \lambda\mathbf{I}_D \otimes \tilde{\mathbf{K}}_{nn}\big)\,\mathrm{vec}\big(\tilde{\mathbf{C}}\big) = \mathrm{vec}\big(\tilde{\mathbf{K}}_{nL}\mathbf{Y}\big). \tag{87}$$

In this case, a prediction for a generic test point $\boldsymbol{x}_* \in \mathcal{X}$ can be written as

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{l=1}^{n} k_{\boldsymbol{x}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}_l)\mathbf{B}\tilde{\boldsymbol{c}}_l = \mathbf{B} \otimes k_{\boldsymbol{x}}\big(\boldsymbol{x}, \tilde{\mathbf{X}}_n\big)\,\mathrm{vec}\big(\tilde{\mathbf{C}}\big). \quad (88)$$

REFERENCES

[1] J. Jin, C. Zhang, F. Feng, W. Na, J. Ma, and Q. Zhang, "Deep neural network technique for high-dimensional microwave modeling and applications to parameter extraction of microwave filters," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 10, pp. 4140–4155, Oct. 2019.

[2] J. Zhang, J. Chen, Q. Guo, W. Liu, F. Feng, and Q.-J. Zhang, "Parameterized modeling incorporating MOR-based rational transfer functions with neural networks for microwave components," *IEEE Microw. Wireless Compon. Lett.*, vol. 32, no. 5, pp. 379–382, May 2022.

[3] W. Zhang et al., "Surrogate-assisted multistate tuning-driven EM optimization for microwave tunable filter," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 4, pp. 2015–2030, Apr. 2022.

[4] H. Kabir, L. Zhang, M. Yu, P. H. Aaen, J. Wood, and Q.-J. Zhang, "Smart modeling of microwave devices," *IEEE Microw. Mag.*, vol. 11, no. 3, pp. 105–118, May 2010.

[5] H. M. Torun, A. C. Durgun, K. Aygun, and M. Swaminathan, "Causal and passive parameterization of S-Parameters using neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 10, pp. 4290–4304, Oct. 2020.

[6] M. Swaminathan, H. M. Torun, H. Yu, J. A. Hejase, and W. D. Becker, "Demystifying machine learning for signal and power integrity problems in packaging," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, no. 8, pp. 1276–1295, Aug. 2020.

[7] R. Kumar et al., "Knowledge-based neural networks for fast design space exploration of hybrid copper-graphene on-chip interconnect networks," *IEEE Trans. Electromagn. Compat.*, vol. 64, no. 1, pp. 182–195, Feb. 2022.

[8] L.-Y. Xiao, W. Shao, X. Ding, and B.-Z. Wang, "Dynamic adjustment kernel extreme learning machine for microwave component design," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 10, pp. 4452–4461, Oct. 2018.

[9] F. Treviso, R. Trinchero, and F. G. Canavero, "Multiple delay identification in long interconnects via LS-SVM regression," *IEEE Access*, vol. 9, pp. 39028–39042, 2021.

[10] H. Ma, E.-P. Li, A. C. Cangellaris, and X. Chen, "Support vector regression-based active subspace (SVR-AS) modeling of high-speed links for fast and accurate sensitivity analysis," *IEEE Access*, vol. 8, pp. 74339–74348, 2020.

[11] R. Trinchero, M. Larbi, H. Torun, F. G. Canavero, and M. Swaminathan, "Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters," *IEEE Access*, vol. 7, pp. 4056–4066, 2018.

[12] N. Soleimani and R. Trinchero, "Compressed complex-valued least squares support vector machine regression for modeling of the frequency-domain responses of electromagnetic structures," *Electronics*, vol. 11, no. 4, p. 551, Feb. 2022.

[13] R. Trinchero and F. Canavero, "Machine learning regression techniques for the modeling of complex systems: An overview," *IEEE Electromagn. Compat. Mag.*, vol. 10, no. 4, pp. 71–79, 4th Quart., 2021.

[14] S. Kushwaha, A. Attar, R. Trinchero, F. Canavero, R. Sharma, and S. Roy, "Fast extraction of per-unit-length parameters of hybrid copper-graphene interconnects via generalized knowledge based machine learning," in *Proc. IEEE 30th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2021, pp. 1–3.

[15] S. Koziel, A. Pietrenko-Dabrowska, and M. Al-Hasan, "Design-oriented two-stage surrogate modeling of miniaturized microstrip circuits with dimensionality reduction," *IEEE Access*, vol. 8, pp. 121744–121754, 2020.

[16] S. Kushwaha et al., "Comparative analysis of prior knowledge-based machine learning metamodels for modeling hybrid copper–graphene on-chip interconnects," *IEEE Trans. Electromagn. Compat.*, vol. 64, no. 6, pp. 2249–2260, Dec. 2022.

[17] T. Bradde, S. Grivet-Talocia, A. Zanco, and G. C. Calafiore, "Data-driven extraction of uniformly stable and passive parameterized macromodels," *IEEE Access*, vol. 10, pp. 15786–15804, 2022.

[18] T. Bradde, S. Grivet-Talocia, M. De Stefano, and A. Zanco, "A scalable reduced-order modeling algorithm for the construction of parameterized interconnect macromodels from scattering responses," in *Proc. IEEE Symp. Electromagn. Compat., Signal Integrity Power Integrity (EMC, SI PI)*, Jul. 2018, pp. 650–655.

[19] A. Zanco and S. Grivet-Talocia, "Toward fully automated high-dimensional parameterized macromodeling," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 11, no. 9, pp. 1402–1416, Sep. 2021.

[20] K. T. Fang, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments*. London, U.K.: Taylor & Francis Group, 2016.

[21] M. C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[22] V. Vapnik, *The Nature of Statistical Learning Theory*. 2nd ed. New York NY, USA: Springer, 1999.

[23] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[24] J. A. K. Suykens et al., *Least Squares Support Vector Machines*. Singapore: World Scientific, 2002.

[25] A. Rudi, L. Carratino, and L. Rosasco, "FALKON: An optimal large scale kernel method," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

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

[27] P. Manfredi and R. Trinchero, "A data compression strategy for the efficient uncertainty quantification of time-domain circuit responses," *IEEE Access*, vol. 8, pp. 92019–92027, 2020.

[28] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY, USA: Springer-Verlag, 2002.

[29] N. Soleimani, R. Trinchero, and F. Canavero, "Vector-valued kernel ridge regression for the modeling of high-speed links," in *IEEE MTT-S Int. Microw. Symp. Dig.* Limoges, France: IEEE, 2022.

[30] A. C. Micchelli and M. Pontil, "Kernels for multi-task learning," in *Proc. 17th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2004, pp. 921–928.

[31] C. A. Micchelli and M. A. Pontil, "On learning vector-valued functions," *Neural Comput.*, vol. 17, no. 1, pp. 177–204, 2005.

[32] L. Baldassarre et al., "Multi-output learning via spectral filtering," *Mach. Learn.*, vol. 87, pp. 259–301, Jun. 2012.

[33] C. K. I. Williams and M. Seeger, "Using the Nystrom method to speed up kernel machines," in *Proc. 13th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2000, pp. 661–667.

[34] A. Rudi, R. Camoriano, and L. Rosasco, "Less is more: Nystrom computational regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, Red Hook, NY, USA, 2015, pp. 1657–1665.

[35] A. J. Smola and B. Scholkop, "Sparse greedy matrix approximation for machine learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2000, pp. 911–918.

[36] B. Scholkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. 14th Annu. Conf. Comput. Learn. Theory*, 2001, pp. 416–426.

[37] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial," 2019, *arXiv:1905.12787*.

[38] J. Shawe-Taylor and N. Cristianini, "Kernel methods: An overview," in *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004, pp. 25–46.

[39] M. Welling. *Kernel Ridge Regression Max Welling's Classnotes in Machine Learning: 1–3*. Accessed: Sep. 2022. [Online]. Available: https://www.ics.uci.edu/welling/classnotes/papers_class/Kernel-Ridge.pdf

[40] F. Dinuzzo, C. S. Ong, P. Gehler, and G. Pillonetto, "Learning output kernels with block coordinate descent," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 49–56.

[41] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

[42] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.

[43] R. M. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Nat. Bur. Standards*, vol. 49, no. 6, pp. 409–496, Dec. 1951.

[44] A. Grebennikov and J. Wong, "A dual-band parallel Doherty power amplifier for wireless applications," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 10, pp. 3214–3222, Oct. 2012.
[45] A. Guarino, R. Trinchero, F. Canavero, and G. Spagnuolo, "A fast fuel cell parametric identification approach based on machine learning inverse models," *Energy*, vol. 239, Jan. 2022, Art. no. 122140.
[46] V. Sindhwani, H. Q. Minh, and A. C. Lozano, "Scalable matrix-valued kernel learning for high-dimensional nonlinear multivariate regression and Granger causality," in *Proc. Uncertainty Artif. Intell. (UAI)*, 2013, pp. 1–22.



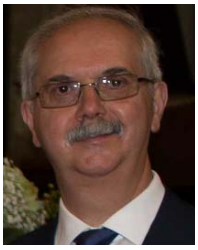
Nastaran Soleimani (Student Member, IEEE) received the M.Sc. degree in electronics engineering from the Politecnico di Torino, Turin, Italy, in 2011, where she is currently pursuing the Ph.D. degree with the Department of Electronics and Telecommunications.

Her research interests include machine learning and statistical simulation of circuits and systems.



Riccardo Trinchero (Member, IEEE) received the M.Sc. degree in electronics engineering from the Politecnico di Torino, Turin, Italy, in 2011, where she is currently pursuing the Ph.D. degree with the Department of Electronics and Telecommunications.

Her research interests include machine learning and statistical simulation of circuits and systems.



Flavio G. Canavero (Life Fellow, IEEE) received the master's degree in electronic engineering from the Politecnico (Technical University) di Torino, Turin, Italy, in 1977, and the Ph.D. degree in geophysical sciences from the Georgia Institute of Technology, Atlanta, GA, USA, in 1986.

In November 2022, he retired from the Politecnico di Torino, where he was a Professor of circuit theory with the Department of Electronics and Telecommunications. His research interests include signal integrity and electromagnetic compatibility (EMC) design issues, interconnect modeling, black-box characterization of digital integrated circuits, and statistics quantification of uncertainty.

Dr. Canavero received several Industry and IEEE Awards, including the prestigious Richard R. Stoddard Award for Outstanding Performance, which is the EMC Society's highest technical award, and the Honored Member Award of EMC Society. He has been the Editor-in-Chief of IEEE TRANSACTIONS ON ELECTROMAGNETIC COMPATIBILITY, the Vice President of Communication Services of the EMC Society, and the Chair of URSI Commission E.

Open Access funding provided by 'Politecnico di Torino' within the CRUI CARE Agreement