POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Data Compression Strategy for the Efficient Uncertainty Quantification of Time-Domain Circuit Responses

Original
A Data Compression Strategy for the Efficient Uncertainty Quantification of Time-Domain Circuit Responses / Manfredi, P., Trinchero, R.. - In: IEEE ACCESS. - ISSN 2169-3536. - ELETTRONICO. - 8:(2020), pp. 92019-92027. [10.1109/ACCESS.2020.2993338]

Availability:
This version is available at: 11583/2836544 since: 2022-03-17T18:43:46Z

Publisher:
Institute of Electrical and Electronics Engineers Inc.

Published
DOI:10.1109/ACCESS.2020.2993338

Terms of use:

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

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

04 November 2026

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 23, 2020, accepted May 5, 2020, date of publication May 8, 2020, date of current version May 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993338

A Data Compression Strategy for the Efficient Uncertainty Quantification of Time-Domain Circuit Responses

PAOLO MANFREDI, (Senior Member, IEEE), AND RICCARDO TRINCHERO, (Member, IEEE)
Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy
Corresponding author: Riccardo Trinchero (riccardo.trinchero@polito.it)

ABSTRACT This paper presents an innovative modeling strategy for the construction of efficient and compact surrogate models for the uncertainty quantification of time-domain responses of digital links. The proposed approach relies on a two-step methodology. First, the initial dataset of available training responses is compressed via principal component analysis (PCA). Then, the compressed dataset is used to train compact surrogate models for the reduced PCA variables using advanced techniques for uncertainty quantification and parametric macromodeling. Specifically, in this work sparse polynomial chaos expansion and least-square support-vector machine regression are used, although the proposed methodology is general and applicable to any surrogate modeling strategy. The preliminary compression allows limiting the number and complexity of the surrogate models, thus leading to a substantial improvement in the efficiency. The feasibility and performance of the proposed approach are investigated by means of two digital link designs with 54 and 115 uncertain parameters, respectively.

INDEX TERMS High-speed link, least-squares support-vector machine, machine learning, polynomial chaos expansion, principal component analysis, signal integrity, uncertainty quantification.

I. INTRODUCTION

The ever-growing demand for higher data rates in high-speed links, along with the increasing complexity and miniaturization, is making the effect of uncontrolled variations of design parameters (e.g., geometry, material parameters, and components tolerances) on system performance far from being negligible.

Among the several available approaches for uncertainty quantification, Monte Carlo (MC) simulation is undoubtedly the most straightforward method for assessing link performance with respect to parameter uncertainty. Indeed, MC allows predicting statistical quantities of the outputs of interests (e.g., voltage/current overshoots, eye digram opening, maximum dissipated power, etc.) using a set of deterministic simulations computed for some suitable random samples of the uncertain input parameters, drawn according to their probability distribution. While being straightforward to implement and virtually available in any design environment, an accurate statistical assessment via MC analysis typically requires an exorbitant number of simulations, thus becoming impractical for real-life scenarios.

To overcome this computational limitation, stochastic macromodeling techniques based on the framework of polynomial chaos expansion (PCE) [1] or machine learning (ML) [2], [3] were extensively investigated in recent years for electrical and electronic engineering applications, see for example [4]–[16], respectively. The common underlying idea is to use a small set of simulation results, exploring the parameter space as much as possible, to ''train'' a closed-form surrogate model of the expensive ''full-computational'' model. This surrogate model is then used to inexpensively predict the system performance for any possible configuration of the uncertain input parameters, thus allowing for the rapid calculation of statistical information.

In the most versatile implementations, the surrogate model parameters are calculated, and possibly tuned, via suitable regression techniques [4], [5], [11]. However, while in standard PCE implementations a common set of basis functions is used for any sweep point and any output of interest, non-parametric regression-based ML tools and advanced PCE techniques require to tune the hyper-parameters and/or the

The associate editor coordinating the review of this manuscript and approving it for publication was Wiren Becker.

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

basis functions, and therefore to solve a different optimization problem for each output of interest and sweep point (e.g., [8], [9], [14], [15]). This makes the abovementioned approaches prohibitive for problems with multiple and possibly time- or frequency-dependent outputs.

Examples include sparse PCE approaches, such as those based on tensor recovery [17] or other adaptive schemes [18], [19], and non-parametric ML regression techniques like support-vector machine (SVM) [20], [21], least-squares SVM (LS-SVM) [22], and Gaussian process regression (GPR) [23]. All these techniques help mitigate the ''curse of dimensionality'', i.e., the efficiency reduction when the number of uncertain input parameters increases. Therefore, advanced stochastic surrogate models are the winning choice when applied to a limited set of outputs of interest. Otherwise, their direct application to the modeling of multiple transient responses turns out to be rather cumbersome, because a large number of time points should be considered to capture highly nonlinear dynamics.

Usually, there are two different approaches to tackle the above issue. One solution is to include the effect of both system parameters and transient dynamics into a single recursive model [24], for example via a neural network [3], [25], [26]. Such models also include the realizations at previous time points as additional input parameters. Despite its compactness, the resulting model is so highly complex that a very large number of training samples (in the order of thousands) is needed, since a huge number of model parameters must be simultaneously tuned. This approach might be a viable solution for low-order linear systems, but the model complexity grows exponentially for higher-order nonlinear systems.

A reasonable alternative is to build a different surrogate model for each output variable and time instant of interest (see, e.g., [9]). While this reduces the complexity of the single model, the overall number of models to be created is potentially huge, especially if a large number of time points is required, like for example when simulating eye diagrams.

This paper proposes an alternative solution to overcome the shortcomings of this second approach. Specifically, principal component analysis (PCA) [27] is used to remove redundant information from input data samples, which are therefore reduced to a minimum subset [28]. Briefly speaking, the underlying idea is to exploit and remove the inherent correlation existing among several realizations of various responses of the system evaluated at different time points. A compression rate of several orders of magnitude is usually achieved, thus making the application of advanced stochastic surrogate models feasible for this reduced dataset. The implementation through singular value decomposition (SVD) allows controlling the accuracy of the compression.

The feasibility and strength of the proposed technique are assessed by means of two high-speed links: a 16-bit Flash memory bus operating at 66 MHz ($\approx$1 Gbps) and affected by 54 uncertain design parameters, and a single electronic link affected by 155 stochastic variables and driven by a DDR buffer transmitting at 133 Mbps. For both test cases, PCA is applied to obtain a compressed representation of the training data. Then, two types of surrogate models are considered, namely a sparse PCE in combination with least-angle regression (LAR) [19] and a LS-SVM regression [22].

The paper is organized as follows. The problem and the goals addressed in this work are stated in Section II. Section III outlines the proposed compression scheme based on PCA. The performance of the proposed methodology is investigated in Section IV by means of two digital link designs. Finally, Section V concludes the paper. Throughout the paper, $\mathcal{X}$, x, $\mathbf{x}$, $\mathbf{X}$, $\boldsymbol{\mathcal{X}}$ denote a set, a scalar, a vector, a matrix, and a tensor, respectively.

II. PROBLEM STATEMENT

This section briefly introduces the problem under consideration. We consider a generic dynamic nonlinear system

$$\mathbf{y}(t) = \mathcal{M}(t; \mathbf{x}), \tag{1}$$

where $\mathbf{x} = [x_1, \ldots, x_d]^T \in \mathbb{R}^d$ is a set of (uncertain) input design parameters, $\mathbf{y}(t) = [y_1(t), \ldots, y_M(t)]^T \in \mathbb{R}^M$ are the system outputs, and t is an independent sweep variable which the outputs depend on.[1] Without loss of generality, we do assume that all d uncertain parameters in $\mathbf{x}$ are independent and equally significant. Techniques are available to possibly reduce the set of input parameters when they are correlated [29].

Let us introduce the following set of training pairs:

$$\mathcal{D} = \{(\mathbf{x}_l, \mathbf{y}_l(t_k))\}_{l,k=1}^{L,K} \tag{2}$$

where $\mathbf{y}_l(t_k)$ are vectors collecting the M system outputs, computed from (1) for a specific configuration $\mathbf{x}_l$ of the input parameters at K distinct and increasing time points $\{t_k\}_{k=1}^{K}$. For the sake of notation compactness, the set of training pairs is rewritten as a union of subsets

$$\mathcal{D} = \bigcup_{k,m=1}^{K,M} \mathcal{D}_{k,m} \tag{3}$$

where each subset is defined as

$$\mathcal{D}_{k,m} = \{(\mathbf{x}_l, y_{l,k,m})\}_{l=1}^{L}, \tag{4}$$

and $y_{l,k,m}$ denotes the mth output evaluated at the kth time instant for the lth configuration of the input parameters, i.e., $y_{l,k,m} = \mathcal{M}_m(t_k; \mathbf{x}_l)$, with subscript m denoting the specific system output considered.

Starting from a set of training pairs $\mathcal{D}_{k,m}$, we seek for the best configurations of parameters $\{\mathbf{w}_{k,m}\}_{k,m=1}^{K,M}$, each defining a surrogate model $\tilde{\mathcal{M}}_m(t_k; \mathbf{x}, \mathbf{w}_{k,m})$ that minimizes the empirical risk functional

$$R(\mathbf{w}_{k,m}) = \frac{1}{L} \sum_{l=1}^{L} \ell(\tilde{\mathcal{M}}_m(t_k; \mathbf{x}_l, \mathbf{w}_{k,m}), y_{l,k,m}) \tag{5}$$

[1] In the context of this paper, t denotes time; however, it could also be any other sweep variable like frequency, input power, temperature, etc.

where ℓ is the so-called loss-function. For example, the squared difference

$$\ell(\tilde{\mathcal{M}}(t;\mathbf{x},\mathbf{w}),y) = (\tilde{\mathcal{M}}(t;\mathbf{x},\mathbf{w}) - y)^2 \tag{6}$$

is the loss function used by the ordinary least-squares regression. Specifically, we are looking for the best set of model parameters $\mathbf{w}^*$ that minimize the model error (5) on the available set of training samples $\mathcal{D}$, i.e.,

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} R(\mathbf{w}). \tag{7}$$

In the framework of uncertainty quantification, the training set $\mathcal{D}$ is obtained by drawing samples $\mathbf{x}_l$ of the input variables randomly or pseudo-randomly, and computing the corresponding system responses $\mathbf{y}_l$. In this paper, we use a Latin hypercube strategy to randomly sample the design space with good exploration properties.

III. PCA-COMPRESSED SURROGATE MODELING

Since the system outputs are time-dependent, it is virtually impossible that a single surrogate model can accurately predict the entire system dynamics at each time point, for each output, and for any configuration of the input parameters $\mathbf{x}$. Therefore, the conventional, practical approach is to build a different surrogate model for each time point t_k and output variable y_m for which data are available, with $k = 1,\ldots,K$ and $m = 1,\ldots,M$. In turn, the set of model parameters changes with time and output variable, and therefore we denote it as $\mathbf{w}_{k,m}$. These parameters are obtained by suitable regression techniques using L training responses $\{y_{l,k,m}\}_{l=1}^{L}$ for each of these variables.

The complete dataset of training responses in $\mathcal{D}$, let us denote it with $\mathcal{Y} = \{y_{l,k,m}\}_{l,k,m=1}^{L,K,M}$, can be interpreted as a $L \times K \times M$ three-way tensor. The data in tensor $\mathcal{Y}$ exhibit some inherent correlation, since they come from the same system. Specifically, data for different responses, at different time points, are not completely unrelated to each other, but rather exhibit a large amount of interdependency. If tensor $\mathcal{Y}$ is reshaped by stacking data for all outputs and time points rowwise, a matrix $\mathbf{Y} \in \mathbb{R}^{KM\times L}$ is obtained. This new dataset $\mathbf{Y}$ can be interpreted as a columnwise collection of L realizations $\{\boldsymbol{\zeta}_l\}_{l=1}^{L}$ of a KM-variate stochastic variable $\boldsymbol{\zeta}$.

We calculate the experimental covariance matrix of $\boldsymbol{\zeta}$, i.e.,

$$\boldsymbol{\Sigma}_{\boldsymbol{\zeta}} = \frac{\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^{\mathsf{T}}}{L-1}, \tag{8}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\zeta}} \in \mathbb{R}^{KM\times KM}$ and

$$\tilde{\mathbf{Y}} = \mathbf{Y} - \boldsymbol{\mu}_{\boldsymbol{\zeta}} \tag{9}$$

is the dataset with the samples epurated from the mean

$$\boldsymbol{\mu}_{\boldsymbol{\zeta}} = \frac{1}{L}\sum_{l=1}^{L}\boldsymbol{\zeta}_l, \tag{10}$$

which is subtracted columnwise.

Each of the $\boldsymbol{\zeta}$-realizations can be expressed as [28]

$$\boldsymbol{\zeta}_l = \boldsymbol{\mu}_{\boldsymbol{\zeta}} + \sum_{n=1}^{KM} Z_{l,n}\mathbf{u}_n, \tag{11}$$

for $l = 1,\ldots,L$, where

$$Z_{l,n} = \mathbf{u}_n^{\mathsf{T}}(\boldsymbol{\zeta}_l - \boldsymbol{\mu}_{\boldsymbol{\zeta}}) \tag{12}$$

and $\{\mathbf{u}_n\}_{n=1}^{KM}$ are the left singular vectors, obtained by calculating the SVD

$$\tilde{\mathbf{Y}} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathsf{T}} \tag{13}$$

and taking the columns of $\mathbf{U}$.

We can now truncate (11) to retain only the $\bar{n}$ ''principal components'':

$$\boldsymbol{\zeta}_l \approx \hat{\boldsymbol{\zeta}}_l = \boldsymbol{\mu}_{\boldsymbol{\zeta}} + \sum_{n=1}^{\bar{n}} Z_{l,n}\mathbf{u}_n, \tag{14}$$

which can be identified by setting a threshold ϵ on the relative magnitude of the singular values of $\tilde{\mathbf{Y}}$, collected in descending order into the diagonal matrix $\mathbf{S}$. The singular values of $\tilde{\mathbf{Y}}$ are proportional to the square root of the eigenvalues of its covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\zeta}}$. If we define $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\zeta}}}$ as the covariance matrix of the approximated data (14), using matrix properties we can conclude that

$$\frac{\|\boldsymbol{\Sigma}_{\hat{\boldsymbol{\zeta}}} - \boldsymbol{\Sigma}_{\boldsymbol{\zeta}}\|_2}{\|\boldsymbol{\Sigma}_{\boldsymbol{\zeta}}\|_2} = \left(\frac{\sigma_{\bar{n}+1}}{\sigma_1}\right)^2 \leq \epsilon^2 \tag{15}$$

where σ_1 is the first singular value of $\tilde{\mathbf{Y}}$ and $\sigma_{\bar{n}+1}$ is the first singular value that is *discarded* by the PCA truncation. Hence, setting a threshold on the singular values allows a rigorous control on the approximation in terms of variance. In this paper, we truncate when the magnitude falls below 1% of the first singular value (i.e., $\epsilon = 10^{-2}$).

The key achievement is that the set $\{Z_{l,n}\}_{l,n=1}^{L,\bar{n}}$ of PCA coefficients (12) can be interpreted as a collection of L samples of $\bar{n}$ new output system variables $Z_n(\mathbf{x})$, with $Z_{l,n} = Z_n(\mathbf{x}_l)$, describing the information pertaining to the entire set of original time-dependent outputs $\mathbf{y}(t)$. Since typically $\bar{n} \lll KM$, the PCA truncation leads to a substantial compression of the number of variables to be modeled. Each of these reduced variables can be approximated using any surrogate model. Once a model is available for the compressed variables Z_n, new samples for the original time-dependent output variables can be recovered via (14). It is important to remark that no specific assumption on the nature of the correlation among the available data is required by the PCA algorithm.

In the appendices, we introduce the two surrogate models that are used in the application examples in conjunction PCA compression, namely the sparse PCE [18] and the LS-SVM regression [22]. However, the proposed compression strategy outlined in this section is general and applies to any generic surrogate model.

IV. APPLICATION EXAMPLES

In this section, the feasibility and performance of the proposed methodology are assessed based on two application examples: a 16-bit channel with 54 uncertain parameters driven by Flash memory chips and a single-channel electronic link with 115 uncertain parameters [8] and driven by a DDR buffer. For the drivers, we use behavioral macromodels constructed using the method in [32], [33], but any other model, either behavioral or transistor-level, may be used.

We use the UQLab toolbox [30] for calculating sparse PCE surrogate models and the LS-SVMlab toolbox [31] to carry out LS-SVM regressions. Specifically, we use third-order polynomials and RBF kernels for the two methods, respectively. This provides a satisfactory trade-off in terms of accuracy and training cost for the considered application examples.

All circuit simulations are performed using HSPICE on a Dell Precision 5820 workstation with an Intel(R) Core(TM) i9-7900X, CPU running at 3.30 GHz, and 32 GB of RAM. The time step of the transient simulation is set to one half of the risetime of the digital signals to ensure satisfactory accuracy and resolution of the waveforms.

A. EXAMPLE 1: 16-BIT FLASH-MEMORY BUS

As a first application test case, the proposed methodology is applied to the link depicted in Fig. 1, which represents a 16-bit transmission channel of a memory chip. For the I/O buffers, we use behavioral macromodels compatible with the drivers of a Flash technology operating at 66 MHz. The structure includes a resistive rail for the power supply of the buffers, a RLC network with package parasitics, and a transmission channel consisting of 16 coupled microstrip traces. The far-end terminations are floating to mimic connection to high-impedance receivers. The even-bit drivers transmit slightly asynchronous pulses with a duration of 15 ns and rise/fall times of 0.1 ns, whereas odd-bit drivers remain quiet in the "low" state.

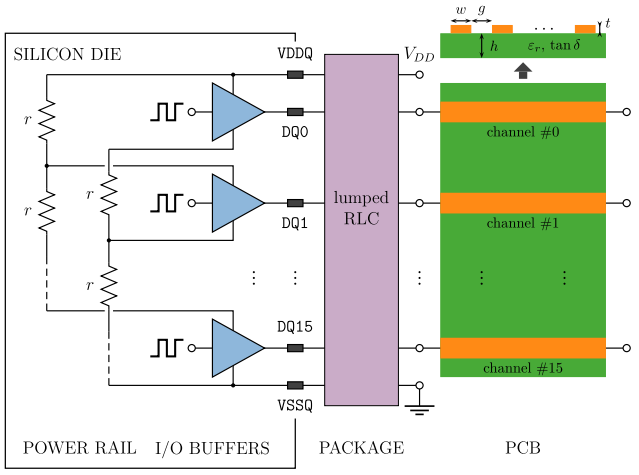


FIGURE 1. Illustration of the 16-bit digital channel for the Flash memory chip.

There are $d = 54$ uncertain design parameters, each with an independent Gaussian distribution and a 10% standard deviation from the nominal value. Namely: the resistance of the power rail (nominal value: $r = 1.14\ \Omega$), the package parasitics (nominal values: resistance $R = 50\ \text{m}\Omega$, inductance $L = 2$ nH, capacitance $C = 5.5$ pF), the width and thickness of each microstrip trace (nominal values: $w = 150\ \mu$m and $t = 30\ \mu$m, respectively), the gap between the traces (nominal value: $g = 150\ \mu$m), and the substrate parameters (nominal values: thickness $h = 460\ \mu$m, relative dielectric permittivity $\varepsilon_r = 4.1$, loss tangent $\tan\delta = 0.02$). The values of package parasitics are here assumed to be independent, even though some correlation usually exists between them. The power supply voltage is $V_{DD} = 1.8$ V and the microstrip trace resistivity is $\rho = 1.72\ \Omega \cdot$ m. The bus is 18-cm long.

The outputs of interest are the terminal voltages at the receiver side of each line and the supply voltages of each driver. These $M = 32$ outputs are evaluated at $K = 901$ equally-spaced time points between 0 and 45 ns. A naive application of advanced surrogate modeling techniques would require the construction of $KM = 28832$ models. On the other hand, the use of a standard, non-sparse PCE implementation with order $p = 3$ would require the determination of the coefficients for $|\mathcal{K}| = 29260$ basis functions, and hence the use of an exorbitant number of training samples to solve the corresponding regression problem.

We consider instead $L = 300$ training configurations of the uncertain parameters, generated according to a Latin hypercube scheme. The corresponding responses are evaluated by means of HSPICE simulations. Figure 2 shows the normalized singular values of the training dataset. The singular values drop below the 1% threshold at $\bar{n} = 51$. Therefore, we retain the first 51 terms in the PCA (11), with a compression rate amounting to less than 0.2% of the original variables. It should be noted that lowering the simulation time step only adds redundant information to the data, which is then cut off by the PCA compression. For example, if the time step is reduced to one tenth of the risetime, the data contains $K = 4501$ time points, but the PCA compression still retains the same number of $\bar{n} = 51$ terms. Moreover, if the PCA is

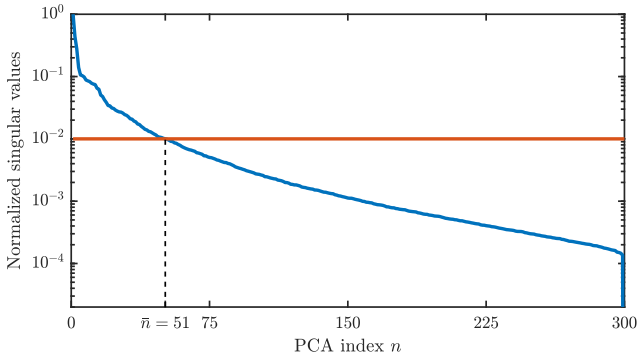


FIGURE 2. Normalized singular value plot of the 16-bit bus dataset (blue line). The horizontal line indicates the 1% threshold for the PCA truncation.

applied separately to each of the 32 outputs (thus only exploiting the correlation of data over time), the overall number of models to be computed becomes 624. This demonstrates that modeling all outputs concurrently allows to effectively take advantage of their interdependency and leads to further compression.

We construct surrogate models for the PCA coefficients using both a sparse third-order PCE and a LS-SVM regression with RBF kernel. In the former case, the cardinality of the subsets $\check{\mathcal{K}}$ for the sparse models of the various PCA coefficients varies between 29 and 132, meaning that the number of non-negligible PCE coefficients is between 0.1% and 0.5% of the total. On the other hand, the hyper-parameters of the LS-SVM regression, such as $\gamma_{k,m}$ in (24) and $\theta_{k,m}$ in (26), are tuned based on the available set of training samples using the leave-one-out cross-validation score [31].

Figure 3 shows the variability and the standard deviation of a selection of four outputs, namely the terminal voltages on channels #0, #1, and #11, and the supply voltage of the buffer of channel #15. The green lines are a set of 500 responses from a reference MC analysis, which help visualize the voltage variation due to the parameter uncertainty. It should be noted that the second and the third plot refer to crosstalk voltages, as the drivers of channels #1 and #11 are not transmitting. Moreover, from the fourth plot, a large fluctuation of the supply voltage is observed, resulting from the commutation of the buffers. The solid blue, dashed red, and dotted yellow curves are the standard deviation estimated from 10000 MC samples, with the sparse PCE, and with the LS-SVM regression, respectively. The results provided by the two surrogate models compare well, and they are also in fairly good agreement with the reference MC results. An exception is the case of the crosstalk voltage on channel #1, for which a rather large discrepancy is observed. The reason could be ascribed to the large amount of ''outlying'' responses that can be observed in Fig. 3, which particularly affects the estimation of the standard deviation, especially around 20 ns.

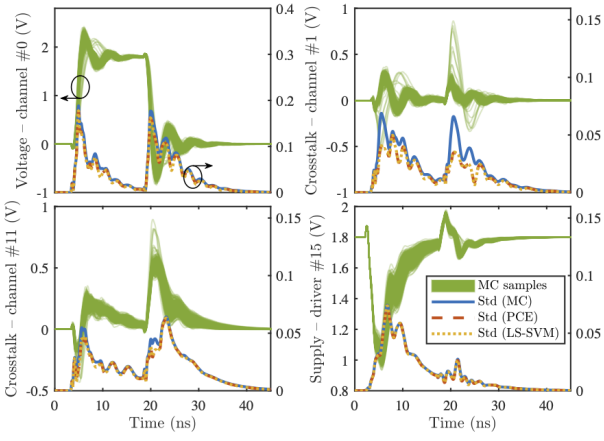


FIGURE 3. Collection of four out of the 32 output voltages for the 16-bit bus of Fig. 1. Green lines: MC samples; solid blue, dashed red, and dotted yellow lines: standard deviation computed with MC, sparse PCE, and LS-SVM regression methods, respectively.

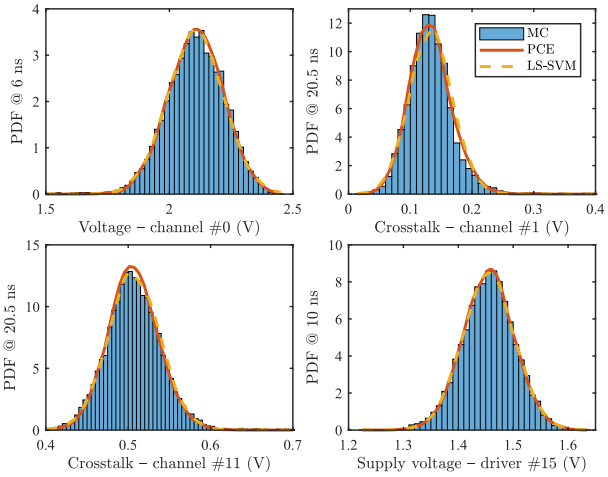


FIGURE 4. PDFs of the output variables in Fig. 3 computed at selected time points of large variability using the MC samples (blue histograms), the sparse PCE (solid red line), and the LS-SVM model (dashed yellow line).

To further assess the accuracy, we calculate the probability density function (PDF) of the quantities in Fig. 3 at time points exhibiting large variability. The results are shown in Fig. 4. A remarkable accuracy is established for both the PCE and the LS-SVM models, even for the crosstalk on channel #1. Similar or better results are found for the remaining outputs.

Table 1 provides the main figures concerning the accuracy and efficiency of the sparse PCE and LS-SVM regression methods in conjunction with PCA compression. The accuracy is assessed in terms of the root-mean-squared (RMS) error over the 10000 samples of the reference MC analysis. The minimum, average, and maximum error over the time points and the various outputs are provided. As far as the computational times are concerned, the table lists the time required by the HSPICE simulation of the training samples, by the PCA compression, for building and evaluating the surrogate models of the PCA coefficients, and by the recovery of the corresponding samples of the orginal variables, from which the statistical information is eventually extracted. For comparison, the HSPICE simulation of the reference MC responses takes about 160000 s (1 day 20 h 27 min). The processing time of both the sparse PCE and LS-SVM regression

TABLE 1. Performance of the sparse PCE and LS-SVM surrogate models in conjunction with PCA compression for the 16-bit bus of Fig. 1.

		sparse PCE	LS-SVM
RMS error (V)	Minimum	5.8095×10^{-7}	2.4235×10^{-7}
	Average	4.8885×10^{-3}	5.2712×10^{-3}
	Maximum	7.0072×10^{-2}	7.1321×10^{-2}
Time (s)	Training	4798.4	
	Compression	1.6	
	Building	84.8	87.4
	Evaluation	4.4	3.1
	Recovery	8.8	
	Total	4898.0	4899.3
	Speed-up	32.7×	32.7×

is minor compared to the simulation of the training samples, and the two techinques exhibit similar performance in terms of accuracy and computational efficiency for this test case.

B. EXAMPLE 2: EYE DIAGRAM OF A DDR LINK

As a second application example, we consider the electronic link investigated in [8]. However, differently from [8], we consider a time-domain simulation in which a behavioral macromodel of a DDR driver operating at 133 Mbps transmits a pseudo-random bit sequence and the uncertainty is provided by $d = 115$ parameters, again following a Gaussian distribution with a 10% relative standard deviation. These parameters include the rail resistance of the driver power supply (nominal value: $r \approx 0.866\ \Omega$), the package parasitics (same nominal values as in the previous test case), as well as the values of all lumped elements and of all geometrical and material parameters of the microstrip lines along the link (we refer to [8] for the numerical values). Both training and validation samples from the full-computational model are again generated by simulating the link using HSPICE.

In this case, we analyze a single output variable, i.e., the voltage at the receiver side (hence, $M = 1$). However, because we consider a sequence of about 1000 bits, each data consist of $K = 150001$ time points, leading to an even larger dataset compared to the previous test case. Different datasets are used to train the surrogate models, with increasing samples sizes of $L = \{50, 100, 200, 300\}$. It should be noted that the number of basis functions for a full PCE would be in this case $|\mathcal{K}| = 266916$.

Figure 5 shows the normalized singular values of the various training datasets. For the smallest dataset with $L = 50$ responses, only the last singular value drops below the 1%-threshold. It is important to mention that the last singular value is always zero because one degree of freedom is lost in removing the mean from the dataset in (9). Therefore, we can conclude that this dataset is not large enough to sufficiently exploit the correlation between the various responses. For the datasets with $L = 100$, $L = 200$, and $L = 300$ training responses instead, the singular values drop below the threshold after index 90, 83, and 76, respectively, leading to a PCA compression between 0.05% and 0.06% of the original variables. It is interesting to note that larger datasets lead to a more effective PCA compression, as a result of the higher

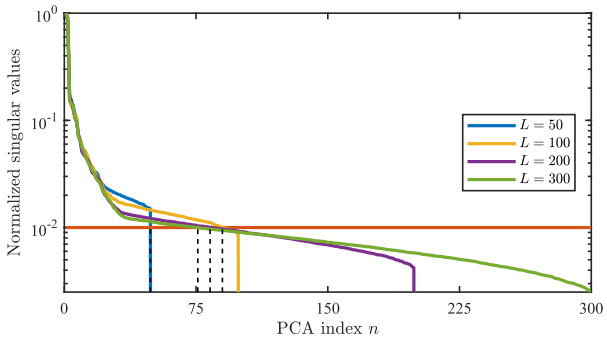


FIGURE 5. Normalized singular value plot of the single-link datasets.

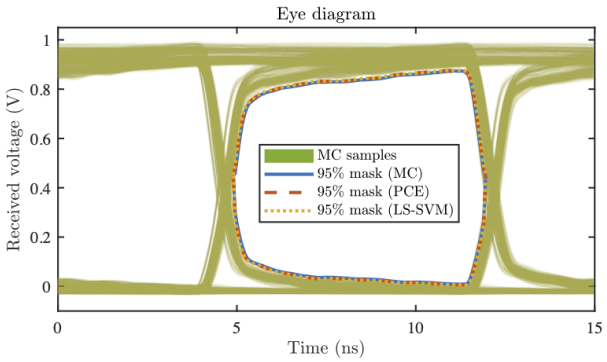


FIGURE 6. Eye diagram of the received voltage in the link of [8] with a DDR transmitter. Green lines: superposition of a subset of bit sequences from the MC analysis; solid blue, dashed red, and dotted yellow lines: mask of the eye opening based on the 95%-quantiles of the high- and low-level voltage distributions.

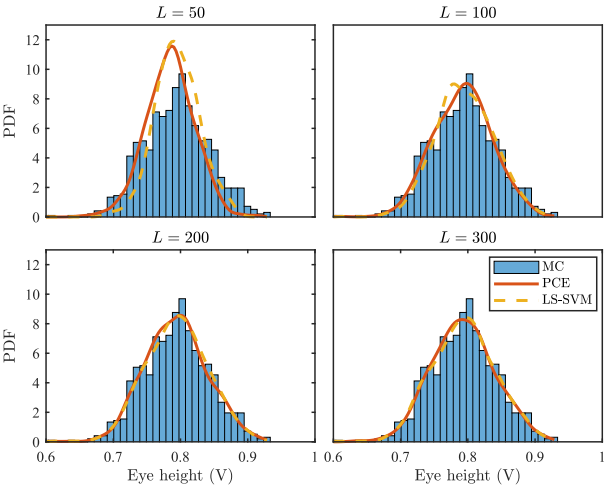


FIGURE 7. PDF of the eye height. The distribution of the MC samples (blue histogram) is compared in the four panels to the PDFs estimated with the sparse PCE (solid red lines) and LS-SVM regression (dashed yellow lines) using different training set sizes.

amount of information contained. For the sparse PCE trained with the largest dataset ($L = 300$ samples), the size of the reduced sets of basis functions ranges from 47 to 147, with a sparsity well below 0.06%.

Figure 6 shows the eye diagram resulting from the superposition of the received bit sequences for a small number of stochastic link realizations. The colored lines represents the eye masks obtained by considering the 95%-quantiles of the distributions of the high and low voltage levels based on 1000 link realizations. The solid blue line is the result obtained with the reference full-computational HSPICE simulations. The dashed red and dotted yellow lines are the eye masks based on the the sparse PCE and LS-SVM surrogate models, respectively, both trained with $L = 300$ responses. Excellent agreement between these techniques is again established.

Furthermore, the four panels of Fig. 7 compare the PDFs of the eye height (maximum opening) obtained with different training set sizes and the reference distribution estimated from the MC analysis (blue histogram). The standard

TABLE 2. Performance of the sparse PCE and LS-SVM surrogate models in conjunction with PCA compression with different training set sizes for the electronic link in [8] driven by a DDR buffer.

		$L = 50$		$L = 100$		$L = 200$		$L = 300$	
		sparse PCE	LS-SVM	sparse PCE	LS-SVM	sparse PCE	LS-SVM	sparse PCE	LS-SVM
Height standard deviation (mV)		36.4	33.2	43.6	42.6	45.9	45.7	46.7	47.1
RMS error (V)	Minimum	1.3438×10^{-3}	1.1758×10^{-3}	1.2737×10^{-3}	9.7132×10^{-4}	1.1394×10^{-3}	9.2412×10^{-4}	1.0261×10^{-3}	7.7654×10^{-4}
	Average	1.1013×10^{-2}	1.4516×10^{-2}	6.6618×10^{-3}	8.3816×10^{-3}	5.1892×10^{-3}	4.9290×10^{-3}	4.4366×10^{-3}	4.5496×10^{-3}
	Maximum	6.7498×10^{-2}	1.3732×10^{-1}	5.6511×10^{-2}	7.0687×10^{-2}	4.0517×10^{-2}	4.9146×10^{-2}	3.0716×10^{-2}	4.8434×10^{-2}
Time (s)	Training	3446.7		6893.4		13786.8		20680.2	
	Compression	1.2		4.1		7.5		10.5	
	Building	177.1	13.6	615.5	32.7	1067.7	66.8	1374.2	119.4
	Evaluation	2.3	0.3	4.5	0.4	4.7	0.7	4.0	0.8
	Recovery	4.7		8.2		7.6		7.1	
	Total	3632.0	3466.5	7525.7	6938.8	14874.3	13869.4	22076.0	20818.0
	Speed-up	19.0×	19.9×	9.2×	9.9×	4.6×	5.0×	3.1×	3.3×

deviation of the eye height estimated from the MC samples is 49.0 mV. The values obtained with the sparse PCE and LS-SVM models are reported in Table 2. It is important to remark at this point that the proposed technique allows for the uncertainty quantification of the entire received bit sequence, and not of just a single synthetic output quantity like the eye height, and hence the determination of more complex information such as the probabilistic eye mask of Fig. 6.

Finally, Table 2 also provides other relevant figures about the accuracy and efficiency of the two surrogate models for this second test case. It is observed that the calculation of the PCA compression, i.e., of the SVD (13) and the subsequent projection (12), has a negligible impact on the overall computational cost, even for this large-size example. Moreover, the sparse PCE achieves slightly lower average and maximum RMS error compared to LS-SVM regression. However, for this application, the latter is much more efficient in the model building phase. This is readily explained by the fact that, as opposed to PCE, the LS-SVM model complexity depends on the training set size rather than on the number of uncertain parameters, as discussed in Appendix B. The time required by the HSPICE simulation of the 1000 reference MC samples is 68934 s (19 h 9 min). This analysis was limited to a smaller number of samples due to the difficulty in handling larger datasets.

V. CONCLUSIONS

This paper presented an uncertainty quantification framework for large time-domain datasets. The proposed approach consists of combining a PCA compression with advanced surrogate modeling strategies such as those based on PCE or ML. The PCA allows reducing the amount of output data to be modeled to a minimum set of variables by exploiting the inherent correlation between the various responses at different time points.

The advocated technique allows the straightforward application of advanced surrogate modeling techniques to the uncertainty quantification of systems with multiple and time-dependent outputs. Two application examples concerning the signal integrity assessment of digital links affected by 54 and 115 uncertain parameters illustrate the strength and feasibility of the proposed approach.

APPENDIX A
SPARSE PCE

Given a set of training pairs $\mathcal{D}_{k,m}$, computed for the time point t_k and output y_m, the corresponding PCE surrogate model has the form

$$\tilde{\mathcal{M}}_m^{\mathrm{PCE}}(t_k;\mathbf{x},\mathbf{w}_{k,m}) = \sum_{\boldsymbol{\kappa}\in\mathcal{K}} \beta_{\boldsymbol{\kappa}}^{k,m}\varphi_{\boldsymbol{\kappa}}(\mathbf{x}), \tag{16}$$

where the functions $\varphi_{\boldsymbol{\kappa}}$ form a basis of orthonormal multivariate polynomials in the input design parameters $\mathbf{x}$. They are typically built as the product combination of univariate polynomials, i.e.,

$$\varphi_{\boldsymbol{\kappa}}(\mathbf{x}) = \prod_{i=1}^{d}\varphi_{\kappa_i}(x_i), \tag{17}$$

and $\mathcal{K}$ is a set of multi-indices $\boldsymbol{\kappa} = [\kappa_1,\ldots,\kappa_d]$ indicating the degrees of the polynomials in each variable. In this case, the model parameters are the PCE coefficients, i.e., $\mathbf{w}_{k,m} = \{\beta_{\boldsymbol{\kappa}}^{k,m}\}_{\boldsymbol{\kappa}\in\mathcal{K}}$.

It is important to point out that in standard and naive PCE implementations [4], [5], the set $\mathcal{K}$ is formed by the multi-indices up to a given *total degree* p, i.e.,

$$\mathcal{K} = \left\{\boldsymbol{\kappa} : \textstyle\sum_{i=1}^{d}\kappa_i \le p\right\}, \tag{18}$$

leading to a cardinality of $|\mathcal{K}| = (p+d)!/(p!d!)$. The same set $\mathcal{K}$ is used for each time point and output variable, which implies that also the basis functions $\varphi_{\boldsymbol{\kappa}}$ do not change. This makes the calculation of the model parameters relatively simple.

Indeed, given a set of training samples $\mathcal{D}_{k,m} = \{(\mathbf{x}_l, y_{l,k,m})\}_{l=1}^{L}$, with $y_{l,k,m} = \mathcal{M}_m(t_k;\mathbf{x}_l)$, the risk functional (5) reads

$$\begin{aligned} R(\mathbf{w}_{k,m}) &= \sum_{l=1}^{L}\left(\sum_{\boldsymbol{\kappa}\in\mathcal{A}}\beta_{\boldsymbol{\kappa}}^{k,m}\varphi_{\boldsymbol{\kappa}}(\mathbf{x}_l) - y_{l,k,m}\right)^2 \\ &= \|\boldsymbol{\Phi}\mathbf{w}_{k,m} - \mathbf{q}_{k,m}\|^2 \end{aligned} \tag{19}$$

where $\mathbf{q}_{k,m} = [y_{1,k,m},\ldots,y_{L,k,m}]^{\mathsf{T}} \in \mathbb{R}^L$ and $\boldsymbol{\Phi} \in \mathbb{R}^{L\times|\mathcal{K}|}$ is a Vandermonde-like matrix with entries $\varphi_{\boldsymbol{\kappa}}(\mathbf{x}_l)$, $\forall\boldsymbol{\kappa}\in\mathcal{K}$, $\forall\mathbf{x}_l\in\mathcal{D}_{k,m}$. The optimal parameter set, minimazing the risk-functional, is readily found as

$$\mathbf{w}_{k,m}^{*} = \boldsymbol{\Phi}^{+}\mathbf{q}_{k,m}, \tag{20}$$

where $\mathbf{\Phi}^+ = (\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi})^{-1}\mathbf{\Phi}^\mathsf{T}$ is the Moore-Penrose pseudo-inverse of $\mathbf{\Phi}$.

Since the regression matrix $\mathbf{\Phi}$ is the same $\forall k, m$, its pseudo-inverse $\mathbf{\Phi}^+$ is computed only once and the above calculation is readily vectorized by stacking data for different datasets $\mathcal{D}_{k,m}$ columnwise, thus obtaining the parameter set for all time points and port variables simultaneously. On the other hand, such a simple model structure may hinder the applicability of this method to high-dimensional problems. Since the regression problem needs to be overdetermined w.r.t. to the number of unknowns $|\mathcal{K}|$, typically at least $L = 2|\mathcal{K}|$ training samples are used, thus rapidly causing the well-known "curse of dimensionality" as the expansion order and/or the number of uncertain design parameters is increased.

In order to mitigate the above limitation, sparse PCEs have been proposed [17], [19]. These techniques exploit the "sparsity-of-effects principle", meaning that most of the coefficients $\beta_{\boldsymbol{\kappa}}^{k,m}$ in (16) are actually negligible [18]. This suggests the adaptive identification of a small subset $\check{\mathcal{K}} \subset \mathcal{K}$ of basis functions and corresponding coefficients, with $|\check{\mathcal{K}}| \ll |\mathcal{K}|$. However, the subset $\check{\mathcal{K}}$ typically differs for each output variable and time point. Hence, a separate regression problem must be set up and solved for each of these variables, with a detrimental impact on the efficiency when dealing with multiple and/or time-dependent outputs.

APPENDIX B
LS-SVM REGRESSION

Given again a set of training samples $\mathcal{D}_{k,m}$, we now look for the best set of parameters $\mathbf{w}_{k,m}$ of the LS-SVM regression in the primal space [22], which reads:

$$\tilde{\mathcal{M}}_m^{\text{LS-SVM}}(t_k; \mathbf{x}, \mathbf{w}_{k,m}) = \langle \mathbf{w}_{k,m}, \boldsymbol{\phi}(\mathbf{x}) \rangle + b_{k,m} \tag{21}$$

where $\mathbf{w}_{k,m} = [w_{k,m}^1, \ldots, w_{k,m}^D]^T$ is a vector collecting the regression coefficients and $\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \ldots, \phi_D(\mathbf{x})]^T$ is a vector collecting the set of basis functions, such that $\boldsymbol{\phi}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^D$.

The risk function minimization problem in (5) for the LS-SVM regression reads

$$\begin{aligned} \min_{\mathbf{w}_{k,m}, \gamma_{k,m}, \mathbf{e}} \; & \frac{1}{2}\|\mathbf{w}_{k,m}\|^2 + \frac{\gamma_{k,m}}{2}\sum_{l=1}^{L} e_l^2 \\ \text{subject to: } & y_{l,k,m} = \langle \mathbf{w}_{k,m}, \boldsymbol{\phi}(\mathbf{x}) \rangle + b_{k,m} + e_l, \end{aligned} \tag{22}$$

where the loss-function ℓ combines the least-squares error e_l^2 with a Tikhonov regularization. The parameter $\gamma_{k,m}$ provides a trade-off between model flatness and accuracy on the available training samples, thus reducing the overfitting. The above formulation of the LS-SVM in the primal space is equivalent to ridge regression.

By using the Lagrangian, the solution of the optimization problem (22) can be recast in terms of the following linear systems of equations:

$$\begin{cases} \sum_{l=1}^{L} \alpha_{l,k,m} = 0 \\ y_{1,k,m} - \sum_{l=1}^{L} \alpha_{l,k,m} K(\mathbf{x}_1, \mathbf{x}_l) - \dfrac{\alpha_{1,k,m}}{\gamma_{k,m}} - b_{k,m} = 0 \\ \qquad\qquad \vdots \\ y_{L,k,m} - \sum_{i=1}^{L} \alpha_{l,k,m} K(\mathbf{x}_L, \mathbf{x}_l) - \dfrac{\alpha_{L,k,m}}{\gamma_{k,m}} - b_{k,m} = 0, \end{cases} \tag{23}$$

where the coefficients $\alpha_{l,k,m}$, for $l = 1, \ldots, L$, are the Lagrangian multipliers associated to the LS-SVM model for the output y_m at the time instant t_k, whereas $K(\mathbf{x}, \mathbf{x}') = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle$ is the so-called "kernel function", such that $K(\cdot, \cdot) : \mathbb{R}^{d \times d} \to \mathbb{R}$.

By introducing the kernel function $K(\mathbf{x}, \mathbf{x}')$, the linear system (23) can be recast in matrix form:

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{\Omega}^{k,m} + \mathbf{I}/\gamma_{k,m} \end{bmatrix} \begin{bmatrix} b_{k,m} \\ \boldsymbol{\alpha}_{k,m} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y}_{k,m} \end{bmatrix} \tag{24}$$

where $\boldsymbol{\alpha}_{k,m} = [\alpha_{1,k,m}, \ldots, \alpha_{L,k,m}]^T \in \mathbb{R}^L$, $\mathbf{y}_{k,m} = [y_{1,k,m}, \ldots, y_{L,k,m}]^T \in \mathbb{R}^L$, $\mathbf{1}^T = [1, \ldots, 1] \in \mathbb{R}^L$, $\mathbf{I} \in \mathbb{R}^{L \times L}$ is the identity matrix, and $\mathbf{\Omega}^{k,m} \in \mathbb{R}^{L \times L}$ is the kernel matrix with elements $\Omega_{ij}^{m,k} = K(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta}_{k,m})$, $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_{k,m}$ with $i, j = 1, \ldots, L$, and in which $\boldsymbol{\theta}_{k,m}$ is a set of hyper-parameters characterizing the kernel function.

By solving (24), we can write the LS-SVM formulation in the dual space:

$$\tilde{\mathcal{M}}_{k,m}^{\text{LS-SVM}}(\mathbf{x}) = \sum_{l=1}^{L} \alpha_{l,k,m} K(\mathbf{x}, \mathbf{x}_l; \boldsymbol{\theta}_{k,m}) + b_{k,m}, \tag{25}$$

where the regression coefficients $\alpha_{l,k,m}$, the bias term $b_{k,m}$, and the kernel hyper-parameters $\boldsymbol{\theta}_{k,m}$, need to be computed for each set of training pairs $\mathcal{D}_{k,m}$, i.e., for each output y_m and time point t_k.

As opposed to the PCE, the LS-SVM regression in the dual form is a non-parametric technique in which the number of coefficients $\alpha_{l,k,m}$ to be estimated equals the number L of training samples, and it is completely independent from the number d of uncertain design parameters. Thanks to the kernel function $K(\cdot, \cdot)$, the dual space formulation does not require an explicit definition of the basis functions $\boldsymbol{\phi}(\mathbf{x})$. This is the so-called "kernel trick".

In this paper, we use the Gaussian radial basis function (RBF) kernel:

$$K(\mathbf{x}, \mathbf{x}'; \theta_{k,m}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma_{k,m}^2}\right), \tag{26}$$

where $\theta_{k,m} = \sigma_{k,m}$ is a single hyper-parameter tuned according to the training samples.

REFERENCES

[1] D. Xiu and G. E. Karniadakis, "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM J. Sci. Comput.*, vol. 24, no. 2, pp. 619–622, 2002.
[2] T. J. Santner, B. J. Williams, and W. Notz, *The Design and Analysis of Computer Experiments*. New York, NY, USA: Springer, 2003.

[3] S. Haykin, *Neural Networks and Learning Machines*. New Jersey, NJ, USA: Pearson Education, 2009.
[4] P. Manfredi, D. Vande Ginste, I. S. Stievano, D. De Zutter, and F. G. Canavero, "Stochastic transmission line analysis via polynomial chaos methods: An overview," *IEEE Electromagn. Compat. Mag.*, vol. 6, no. 3, pp. 77–84, 3rd Quart., 2017.
[5] A. Kaintura, T. Dhaene, and D. Spina, "Review of polynomial chaos-based methods for uncertainty quantification in modern integrated circuits," *Electronics*, vol. 7, no. 3, p. 30, 2018.
[6] M. R. Rufuie, E. Gad, M. Nakhla, and R. Achar, "Generalized Hermite polynomial chaos for variability analysis of macromodels embedded in nonlinear circuits," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 4, no. 4, pp. 673–684, Apr. 2014.
[7] M. Ahadi and S. Roy, "Sparse linear regression (SPLINER) approach for efficient multidimensional uncertainty quantification of high-speed circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 10, pp. 1640–1652, Oct. 2016.
[8] M. Larbi, I. S. Stievano, F. G. Canavero, and P. Besnier, "Variability impact of many design parameters: The case of a realistic electronic link," *IEEE Trans. Electromagn. Compat.*, vol. 60, no. 1, pp. 34–41, Feb. 2018.
[9] P. Manfredi and S. Grivet-Talocia, "Rational polynomial chaos expansions for the stochastic macromodeling of network responses," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 1, pp. 225–234, Jan. 2020.
[10] R. Trinchero, P. Manfredi, I. S. Stievano, and F. G. Canavero, "Machine learning for the performance assessment of high-speed links," *IEEE Trans. Electromagn. Compat.*, vol. 60, no. 6, pp. 1627–1634, Dec. 2018.
[11] R. Trinchero, M. Larbi, H. M. 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, 2019.
[12] R. Trinchero, M. Larbi, M. Swaminathan, and F. G. Canavero, "Statistical analysis of the efficiency of an integrated voltage regulator by means of a machine learning model coupled with kriging regression," in *Proc. IEEE 23rd Workshop Signal Power Integrity (SPI)*, Jun. 2019, pp. 1–4.
[13] T. Lu, J. Sun, K. Wu, and Z. Yang, "High-speed channel modeling with machine learning methods for signal integrity analysis," *IEEE Trans. Electromagn. Compat.*, vol. 60, no. 6, pp. 1957–1964, Dec. 2018.
[14] R. Trinchero, I. S. Stievano, and F. G. Canavero, "Black-box modeling of the maximum currents induced in harnesses during automotive radiated immunity tests," *IEEE Trans. Electromagn. Compat.*, vol. 62, no. 2, pp. 627–630, Apr. 2020.
[15] R. Trinchero and F. G. Canavero, "Combining LS-SVM and GP regression for the uncertainty quantification of the EMI of power converters affected by several uncertain parameters," *IEEE Trans. Electromagn. Compat.*, early access, Jan. 14, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8959370, doi: 10.1109/TEMC.2019.2962899.
[16] H. Ma, E.-P. Li, A. C. Cangellaris, and X. Chen, "Comparison of machine learning techniques for predictive modeling of high-speed links," in *Proc. IEEE 28th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Montreal, QC, Canada, Oct. 2019, pp. 1–3.
[17] Z. Zhang, T.-W. Weng, and L. Daniel, "Big-data tensor recovery for high-dimensional uncertainty quantification of process variations," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 7, no. 5, pp. 687–697, May 2017.
[18] G. Blatman and B. Sudret, "An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis," *Probabilistic Eng. Mech.*, vol. 25, no. 2, pp. 183–197, Apr. 2010.
[19] G. Blatman and B. Sudret, "Adaptive sparse polynomial chaos expansion based on least angle regression," *J. Comput. Phys.*, vol. 230, no. 6, pp. 2345–2367, Mar. 2011.
[20] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York, NY, USA: Springer, 1999.
[21] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.
[22] J. A. K. Suykens, *Least Squares Support Vector Machines*. Singapore: World Scientific, 2002.
[23] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[24] L. Ljung, *System Identification: Theory for the User*, Upper Saddle River, NJ, USA: Prentice-Hall, 1999.
[25] H. M. Torun, H. Yu, N. Dasari, V. C. K. Chekuri, A. Singh, J. Kim, S. K. Lim, S. Mukhopadhyay, and M. Swaminathan, "A spectral convolutional net for co-optimization of integrated voltage regulators and embedded inductors," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Westminster, CO, USA, Nov. 2019, pp. 1–8.
[26] H. Yu, T. Michalka, M. Larbi, and M. Swaminathan, "Behavioral modeling of tunable I/O drivers with preemphasis including power supply noise," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 1, pp. 233–242, Jan. 2020.
[27] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY, USA: Springer-Verlag, 2002.
[28] V. Yaghoubi, S. Marelli, B. Sudret, and T. Abrahamsson, "Sparse polynomial chaos expansions of frequency response functions using stochastic frequency transformation," *Probabilistic Eng. Mech.*, vol. 48, pp. 39–58, Apr. 2017.
[29] J. S. Ochoa and A. C. Cangellaris, "Random-space dimensionality reduction for expedient yield estimation of passive microwave structures," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 12, pp. 4313–4321, Dec. 2013.
[30] S. Marelli and B. Sudret, "UQLab: A framework for uncertainty quantification in MATLAB," in *Proc. Vulnerability, Uncertainty, Risk*, Liverpool, U.K., Jun. 2014, pp. 2554–2563.
[31] (2011). *LS-SVMlab, Version 1.8; Department of Electrical Engineering (ESAT)*, Katholieke Universiteit Leuven: Leuven, Belgium. [Online]. Available: http://www.esat.kuleuven.be/sista/lssvmlab/
[32] I. S. Stievano, L. Rigazio, F. G. Canavero, T. R. Cunha, J. C. Pedro, H. M. Teixeira, A. Girardi, R. Izzi, and F. Vitale, "Behavioral modeling of IC memories from measured data," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 10, pp. 3471–3479, Oct. 2011.
[33] C. Siviero, R. Trinchero, S. Grivet-Talocia, I. S. Stievano, G. Signorini, and M. Telescu, "Constructive signal approximations for fast transient simulation of coupled channels," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 9, no. 10, pp. 2087–2096, Oct. 2019.



PAOLO MANFREDI (Senior Member, IEEE) received the M.Sc. degree in electronic engineering from the Politecnico di Torino, Turin, Italy, in 2009, and the Ph.D. degree in information and communication technology from the Scuola Interpolitecnica di Dottorato, Politecnico di Torino, in 2013.

From 2013 to 2014, he was a Postdoctoral Researcher with the EMC Group, Politecnico di Torino. From 2014 to 2017, he was a Postdoctoral Research Fellow of the Research Foundation–Flanders (FWO) with the Electromagnetics Group, Department of Information Technology, Ghent University, Ghent, Belgium. He is currently an Assistant Professor with the EMC Group, Department of Electronics and Telecommunications, Politecnico di Torino. His research interests comprise the several aspects of circuit and interconnect modeling and simulation, including statistical and worst-case analysis, signal integrity, and electromagnetic compatibility.

Dr. Manfredi was a recipient of the Outstanding Young Scientist Award at the 2018 Joint IEEE International Symposium on Electromagnetic Compatibility and Asia-Pacific Symposium on Electromagnetic Compatibility, the Best Paper Award at the 2016 IEEE Electrical Design of Advanced Packaging and Systems Symposium, the Best Oral Paper Award and the Best Student Paper Award at the 22nd and 19th IEEE Conference on Electrical Performance of Electronic Packaging and Systems, respectively, the Young Scientist Award at the XXX International Union of Radio Science General Assembly and Scientific Symposium, and an Honorable Mention at the 2011 IEEE Microwave Theory and Techniques Society International Microwave Symposium.



RICCARDO TRINCHERO (Member, IEEE) received the M.Sc. and Ph.D. degrees in electronics and communication engineering from the Politecnico di Torino, Torino, Italy, in 2011 and 2015, respectively.

He is currently a Researcher with the EMC Group, Department of Electronics and Telecommunications, Politecnico di Torino. His research interests include the analysis of linear time-varying systems, modeling and simulation of switching converters, and statistical simulation of circuits and systems.