A Probabilistic Machine Learning Approach for the Uncertainty Quantification of Electronic Circuits Based on Gaussian Process Regression

A Probabilistic Machine Learning Approach for the Uncertainty Quantification of Electronic Circuits Based on Gaussian Process Regression

Paolo Manfredi, *Senior Member, IEEE,* Riccardo Trinchero, *Member, IEEE.*

***Abstract*—This paper introduces a probabilistic machine learning framework for the uncertainty quantification (UQ) of electronic circuits based on Gaussian process regression (GPR). As opposed to classical surrogate modeling techniques, GPR inherently provides information on the model uncertainty. The main contribution of this work is twofold. First, it describes how, in an UQ scenario, the model uncertainty can be combined with the uncertainty of the input design parameters to provide confidence bounds for the statistical estimates of the system outputs, such as moments and probability distributions. These confidence bounds allows assessing the accuracy of the predicted statistics. Second, in order to deal with dynamic multi-output systems, principal component analysis (PCA) is effectively employed to compress the time-dependent output variables into a smaller set of components, for which the training of individual GPR models becomes feasible. The uncertainty on the principal components is then propagated back to the original output variables. Several application examples, ranging from a trivial RLC circuit to real-life designs, are used to illustrate and validate the advocated approach.**

***Index Terms*—Gaussian process regression, machine learning, principal component analysis, probability, statistical analysis, stochastic processes, surrogate modeling, uncertainty quantification.**

I. Introduction

UNCERTAINTY quantification is becoming more and more essential to take into account the effect of process variations in the early-stage design of electronic devices [1]–[9]. In this scenario, Monte Carlo (MC) analysis is a robust method that is still considered to be the golden reference and is implemented in virtually any commercial simulator for yield estimation [10]. However, it becomes extremely time-consuming for realistic designs, since it is characterized by a slow convergence rate. Even though Latin hypercube sampling (LHS) and quasi-MC methods allow reducing the number of simulation samples [11], thereby speeding up the calculation, the family of MC methods remains "blind" as it concerns the input-output relationship, thus hindering more in-depth analyses such as the calculation of sensitivity information or the solution of optimization tasks.

To overcome these limitations, several modeling approaches were investigated to surrogate computationally-expensive systems and expedite their UQ. The methods based on the generalized polynomial chaos [12] are specifically tailored for UQ, since the model aims at minimizing the error w.r.t. the target function in statistical terms, by taking into account the probability distribution of the design parameters. They can be divided into two categories [13]: intrusive methods [14]–[17] are mostly based on stochastic Galerkin projection and favor accuracy at the expense of efficiency and ease of implementation. Non-intrusive methods are instead sampling-based techniques that share the implementational benefits of MC methods, yet with better convergence properties. In particular, sparse implementations are suitable for high-dimensional problems and include least-angle regression [18], [19], sparse interpolations [20], [21], and low-rank tensor decompositions [22]–[25]. All these methods, including sparse ones, are parametric techniques, i.e., they require to define a priori the form of the predictor. Therefore, the cost of model training increases with the dimensionality of the problem in terms of input design variables.

A powerful alternative in this regard is the class of non-parametric machine learning methods, for which the model complexity is transparent to the problem dimensionality, but it is rather related to the number of available training data. Examples include neural networks [26], support-vector machine (SVM) [27], least-square support-vector machine (LS-SVM) [28], and Gaussian process regression (GPR) [29], also known as Kriging [30]. The nonparametric feature makes these methods even more attractive for high-dimensional problems. In particular, enhanced variants of GPR/Kriging were specifically proposed to address the "curse of dimensionality" for systems with a large number of inputs [31]–[36]. Applications of machine learning to circuit design include behavioral macromodeling [37]–[39], sensitivity analysis [40], reliability analysis [41], yield analysis and (Bayesian) design optimization [42]–[45], reliability-based optimization [46], [47], as well as UQ [48]–[50].

Since UQ is essentially a probabilistic task, it is useful to assess the accuracy of the predicted statistics by assigning confidence bounds to them. For example, when performing a classical MC analysis, the standard deviation of the MC estimates of the mean and the variance can be analytically computed. However, when using surrogate models, it is difficult to account for the error introduced by the model itself, as information on the model uncertainty is usually unavailable. In practice, when performing UQ, the surrogate model is implicitly assumed to be arbitrarily accurate. For instance, analytical estimates of the first two statistical moments (mean and variance) can be obtained using the polynomial chaos

P. Manfredi and R. Trinchero are with the Department of Electronics and Telecommunications, Politecnico di Torino, Turin 10129, Italy (e-mail: paolo.manfredi@polito.it; riccardo.trinchero@polito.it).

expansion approach [12]. Yet, such estimates come with no associated uncertainty.

In this scenario, GPR exhibits the attractive feature that the model inherently carries an estimate of its uncertainty. Indeed, as opposed to most surrogate modeling techniques, the GPR model is *probabilistic*, rather than deterministic, and it is described by a Gaussian process (GP) with a given mean function (also called "trend") and covariance. The former provides the best or more likely prediction, while the latter is typically used to assign (local) confidence bounds to the (deterministic) prediction for a given configuration of the inputs. This property has been exploited to guide adaptive sampling strategies in deterministic modeling problems, in which new samples are selected in the regions exhibiting the largest prediction variance [51]–[53].

In an UQ scenario, the difficulty lies in combining the model uncertainty with the one of the input parameters, as the correlation between different samples must be suitably taken into account. A naive application of GPR to UQ is to sample the model trend in order to obtain an inexpensive estimate of the MC samples [50], as is typically done with any other deterministic surrogate model. However, this causes the loss of the precious information about the model uncertainty.

The aim of this paper is to provide a probabilistic framework for UQ, in which the uncertainty of the input design parameters is propagated to the output variables by also taking into account, in a rigorous way, the inherent uncertainty of the GPR model. The Gaussian nature of the model uncertainty turns out to be particularly convenient to accomplish this task, while any distribution can be assigned to the input design variables. The result is a simulation framework for UQ in which statistical estimates, such as moments and probability distributions of the outputs, are given in terms of expectations and confidence bounds, as is usually done in traditional MC analysis. However, these confidence bounds now simultaneously account for the uncertainty of both finite sampling and model error. This allows assessing the uncertainty of the model predictions due, e.g., to the lack of data and/or non-optimal assumptions on the prior.

To our best knowledge, there has been a very limited attempt to undertake this task in the literature. An example is found in [57], in which the same prior covariance is assumed for all output components, and a separate GPR model is trained for each statistical moment thereof. Moreover, the predictions at different input values are assumed to be conditionally independent, thus resulting in wider confidence bounds. In our paper instead, we express each output variable with a single GPR model, and we obtain statistical information by rigorously propagating uncertainty using statistical properties.

Furthermore, an important limitation of most surrogate modeling techniques is that they readily apply to scalar outputs, whereas their application to the time-dependent responses of a multi-output dynamic system becomes cumbersome. A naive approach is to train an individual model for each variable and time point of interest, which however becomes computationally intractable for large systems. Alternatively, this problem could be directly addressed using deep learning approaches, such as deep neural networks [54], recurrent neural networks [55], or deep networks with long short-term memory [56]. These approaches allows tackling problems with long time-series output data and a high-dimensional input space. However, the training of deep-learning models is in general extremely expensive, as it requires a very large number of observations and the solution of a high-dimensional non-convex optimization problem to estimate the huge amount of weights that describe their complex network topology. Therefore, these methods are not very attractive for applications to UQ, since the cost of the training phase would be comparable to the one of a typical MC analysis. Our goal here is to train an accurate model using the least amount of data.

We overcome this limitation by means of principal component analysis (PCA), which allows compressing the outputs into a much smaller set of reduced variables, for which the training of individual models becomes feasible [58]. The PCA compression exhibits three convenient features, as it allows: 1) to rigorously control the accuracy of the compression; 2) to readily propagate the uncertainty on the compressed variables back to the original outputs, since it is in fact a linear combination of Gaussian random variables; 3) to automatically take into account redundancy and correlation between data, thus scaling extremely favorably when a finer time step or more output variables are considered in the analysis. In [57], the correlation between different outputs is neglected, whereas in a later paper [59] the correlation is represented by a large covariance matrix that is assumed to be factorizable as the product of smaller matrices, which is not necessarily the case.

The advocated approach is validated starting from a trivial illustrative example, and by then considering two real-life designs, namely a power amplifier and digital integrated link. The rest of the paper is organized as follows. Section II introduces the fundamental notions of GPR. Section III discusses the application of GPR to UQ, whereas rigorous statistical estimates and their confidence bounds are provided in Section IV. Section V introduces the PCA compression. An illustrative application example is provided in Section VI, whereas realistic application scenarios are discussed in Section VII. Finally, conclusions are drawn in Section VIII. Throughout the paper, plain letters (x or X) denote scalar variables, lowercase bold letters ($\boldsymbol{x}$) denote vectors, and uppercase bold letters ($\boldsymbol{X}$) denote matrices. The superscript $^{\mathsf{T}}$ denotes the transpose operator.

II. Classical Gaussian Process Regression

Consider a generic system

$$y = \mathcal{M}(\boldsymbol{x}) \tag{1}$$

depending on a set of input parameters $\boldsymbol{x} = (x_1, \ldots, x_d)^{\mathsf{T}}$, which for the moment we assume to be deterministic. In (1), $\mathcal{M} : \mathbb{R}^d \to \mathbb{R}$ is understood to be a function (or "model") that maps, either explicitly or implicitly, a given configuration of the input parameters $\boldsymbol{x}$ to the corresponding system output y. At this time, we also assume that the output of model (1) is a scalar. Later, in Section III, we relax the assumption on the non-stochasticity of $\boldsymbol{x}$, whereas we extend the discussion to multi-output dynamical systems in Section V.

The underlying idea of GPR is to consider the target function y as a *realization* of a GP with a given mean function $\mu(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ and a covariance or kernel function $k(\boldsymbol{x}, \boldsymbol{x}') : \mathbb{R}^{d\times d} \to \mathbb{R}$. We call this GP the *prior*, and we compactly denote it as $\mathcal{GP}\left(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right)$. For the moment, we assume that the prior mean and kernel be fully specified.

Let us now assume that a set of L observations $\{y_l\}_{l=1}^L$ of the actual system (1) be available for some configurations $\{\boldsymbol{x}_l\}_{l=1}^L$ of the input parameters, with $y_l = \mathcal{M}(\boldsymbol{x}_l)\ \forall l$. We denote these data as $\boldsymbol{X} = (\boldsymbol{x}_1^\mathsf{T}, \ldots, \boldsymbol{x}_L^\mathsf{T})^\mathsf{T}$ and $\boldsymbol{y} = (y_1, \ldots, y_L)^\mathsf{T}$, and we use them to "train" the GPR model, i.e., to identify the specific realization of the prior that best fits the available data. This is done by *conditioning* the prior on the observations $(\boldsymbol{X}, \boldsymbol{y})$, thus resulting in a *posterior* GP whose realizations interpolate the training data. The posterior mean and covariance function are therefore found as [29]

$$m(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{x})\boldsymbol{K}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}) \tag{2}$$

and

$$c(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}(\boldsymbol{x})\boldsymbol{K}^{-1}\boldsymbol{k}(\boldsymbol{x}')^\mathsf{T}, \tag{3}$$

respectively. In the above equations:

- $\boldsymbol{\mu} \in \mathbb{R}^L$ is a column vector with entries $\mu_l = \mu(\boldsymbol{x}_l)$;
- $\boldsymbol{K} \in \mathbb{R}^{L\times L}$ is a matrix with entries $K_{lm} = k(\boldsymbol{x}_l, \boldsymbol{x}_m)$;
- $\boldsymbol{k}(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^L$ is a vector function with components $k_l(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_l)$.

With the above definitions, the resulting GPR model of y is

$$y \approx \hat{\mathcal{M}}(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), c(\boldsymbol{x}, \boldsymbol{x}')\right). \tag{4}$$

Therefore, even for a deterministic system (1), the GPR model is a *stochastic*, rather than a deterministic function, as opposed to most surrogate modeling techniques. Because of this, an attractive feature of GPR models is that they implicitly carry information on the model uncertainty. The calculation of this uncertainty is eased by the fact that the distribution of the model is Gaussian, as many notable properties are available for Gaussian random variables.

For example, the prediction mean coincides with the posterior trend $m(\boldsymbol{x})$, which can be regarded as the "most likely" model. However, any function sampled from the GP (4) is a reasonable model, albeit less likely. The model (local) standard deviation is obtained as $\sigma(\boldsymbol{x}) = \sqrt{c(\boldsymbol{x}, \boldsymbol{x})}$, and it is used to provide confidence bounds for the model prediction at a given value of $\boldsymbol{x}$. The standard deviation collapses to zero at the observations [30], and generally reduces by increasing the number of training samples.

In practice, the posterior is evaluated at a discrete set $\boldsymbol{X}_*$ of values of $\boldsymbol{x}$. In this case, (2) and (3) reduce to the mean vector and covariance matrix of a finite set of correlated Gaussian variables, obtained as

$$\boldsymbol{m} = \boldsymbol{\mu}_* + \boldsymbol{K}_*\boldsymbol{K}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}) \tag{5}$$

and

$$\boldsymbol{C} = \boldsymbol{K}_{**} - \boldsymbol{K}_*\boldsymbol{K}^{-1}\boldsymbol{K}_*^\mathsf{T}, \tag{6}$$

respectively, where the meaning of vector $\boldsymbol{\mu}_*$ and matrices $\boldsymbol{K}_*$ and $\boldsymbol{K}_{**}$ is inferred from the definitions in (2) and (3).

It should be noted that the complexity of the GPR model is independent on the dimensionality of the input space d, and it is only related to the number of available training samples, which determines the size of the covariance matrix that is inverted in (5) and (6) to obtain the posterior trend and covariance. Therefore, the GPR model is nonparametric. The calculation of the covariance (6) may be ill-conditioned, leading to a non-positive semi-definite matrix. Nevertheless, this property can be enforced by finding the "nearest" positive semi-define matrix that minimizes the Frobenius norm of the difference [60].

A. Prior Trend

So far, we assumed that both the prior mean and covariance function be available. In practice, this is hardly the case, but several options are widely used as guesses.

For the mean function, it is very common to simply assume $\mu(\boldsymbol{x}) = 0$. It is important to point out that this apparent oversimplification represents by no means an actual limitation since, according to (2), the posterior mean is not confined to be zero [29]. Indeed, a suitable correction of the prior trend is learned in order to yield a posterior trend that is consistent with the observed data. A predefined and fixed function can be alternatively used based on possible prior knowledge of the system output behavior.

Another and more general option is to use a linear combination of predefined basis functions $\boldsymbol{h}(\boldsymbol{x}) = (h_0(\boldsymbol{x}), \ldots, h_P(\boldsymbol{x}))^\mathsf{T}$ with unknown coefficients $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_P)^\mathsf{T}$, i.e., $\mu(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^\mathsf{T}\boldsymbol{\beta}$, which is referred to as "universal Kriging". For example, a PCE is used for the trend in the so-called PCE-Kriging [61]. A special (and quite common) case is to merely use as a trend a non-zero constant, i.e., $\mu(\boldsymbol{x}) = \beta_0$, which is referred to as "ordinary Kriging".

The use of a non-zero prior trend may improve the model interpretability, but it was shown to provide marginal benefit (see, e.g., [61]). Moreover, a non-fixed trend requires the additional estimation of the coefficients $\boldsymbol{\beta}$. In this case, there is an added contribution to the posterior covariance (3), accounting for the additional uncertainty in the parameters of the trend [29], [30]. The derivations presented in the following apply seamlessly, provided that the correct expression is used for the covariance.

B. Kernel Function

Another key ingredient of a GPR model is the prior kernel. Also in this case, several choices are available [29], [30]. The most popular are surely the squared-exponential (Gaussian) kernel

$$k(\boldsymbol{x}, \boldsymbol{x}'|\boldsymbol{\theta}) = \sigma_f^2 \exp\left(-\frac{1}{2}r^2\right), \tag{7}$$

and the Matérn 5/2 kernel

$$k(\boldsymbol{x}, \boldsymbol{x}'|\boldsymbol{\theta}) = \sigma_f^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \exp\left(-\sqrt{5}r\right), \tag{8}$$

where

$$r = \sqrt{\sum_{j=1}^{d} \frac{(x_j - x_j')^2}{\theta_j^2}} \tag{9}$$

and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_d)$ is a set of *hyperparameters* defining the correlation length along each input dimension. Such kernels were shown to provide good generalization properties and to work well with relatively smooth functions [62], as we expect in our applications. As for the prior trend, specific knowledge on the output behavior (e.g., periodicity, non-smoothness, etc.) can guide the selection of more appropriate and specialized kernels [62]. A common simplification is to consider the kernel to be isotropic, i.e., with the same correlation length σ_l for all dimensions ($\theta_j = \sigma_l$, $\forall j = 1, \dots, d$). In this case, r reduces to

$$r = \sqrt{\frac{(\boldsymbol{x}-\boldsymbol{x}')^{\mathsf{T}}(\boldsymbol{x}-\boldsymbol{x}')}{\sigma_l^2}}. \tag{10}$$

The hyperparameters $\boldsymbol{\theta}$ do not need to be specified a priori, but they are usually optimized as part of the training process. The two main strategies to carry out this task are the minimization of either the negative (log) likelihood function over the training data or a cross-validation error [29]. In this work, we rely on the training function available in the MATLAB® Statistics and Machine Learning Toolbox™ [63], which adopts the former approach. Therefore, the optimal hyperparameters are found as [30]

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\left[\log(\det(\boldsymbol{R}(\boldsymbol{\theta}))) + L\log(2\pi\sigma_f^2(\boldsymbol{\theta})) + L\right] \tag{11}$$

where $\boldsymbol{R}(\boldsymbol{\theta})$ is a correlation matrix evaluated at the training points, with entries $R_{lm}(\boldsymbol{\theta}) = k(\boldsymbol{x}_l, \boldsymbol{x}_m|\boldsymbol{\theta})/\sigma_f^2$, whereas

$$\sigma_f^2(\boldsymbol{\theta}) = \frac{1}{L}(\boldsymbol{y}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{R}(\boldsymbol{\theta})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}). \tag{12}$$

Once the hyperparameters have been optimized, (12) is used to calculate the final value of the kernel variance σ_f^2.

It should be noted that, by considering an isotropic kernel, the problem reduces to minimizing a one-dimensional function to find σ_l. Anisotropic kernels are more flexible, yet they require the solution of a high-dimensional optimization problem for the hyperparameter estimation, which may become computationally intensive if the number of input parameters d is large. In the following, we will use an isotropic squared-exponential kernel, unless stated otherwise. In one of the application examples, we will compare its performance against the use of an anisotropic Matérn kernel. From now on, the explicit dependence of the kernel on the hyperparameters is omitted for brevity of notation when there is no ambiguity.

C. Noisy Observations

Sometimes, the training observations might be affected by an error or, differently put, be noisy. In that case, they can be expressed as $y_l = \mathcal{M}(\boldsymbol{x}_l) + \varepsilon$, $\forall l = 1, \dots, L$. It is common to assume that the additive noise ε follows a zero-mean Gaussian distribution, i.e., $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, with σ_n^2 being the noise variance. The outlined method is readily applied to noisy observation by adding a diagonal contribution to matrix $\boldsymbol{K}$, hence replacing it with $\boldsymbol{K} + \sigma_n^2\boldsymbol{I}$ in (2), (3), (5), and (6), where $\boldsymbol{I}$ is the identity matrix of compatible size. Since the noise variance σ_n^2 is usually unknown, it has to be estimated together with the kernel hyperparameters, thereby slightly complicating the optimization problem outlined in Section II-B [30].

Without loss of generality, we assume here that the training observations are noise free, which is reasonable since in our applications they come from computer simulations. The outlined method applies, with minimal modifications, to noisy observations as well, provided that the posterior trend and covariance are corrected as mentioned above.

D. Illustrative Example: Analytical Function

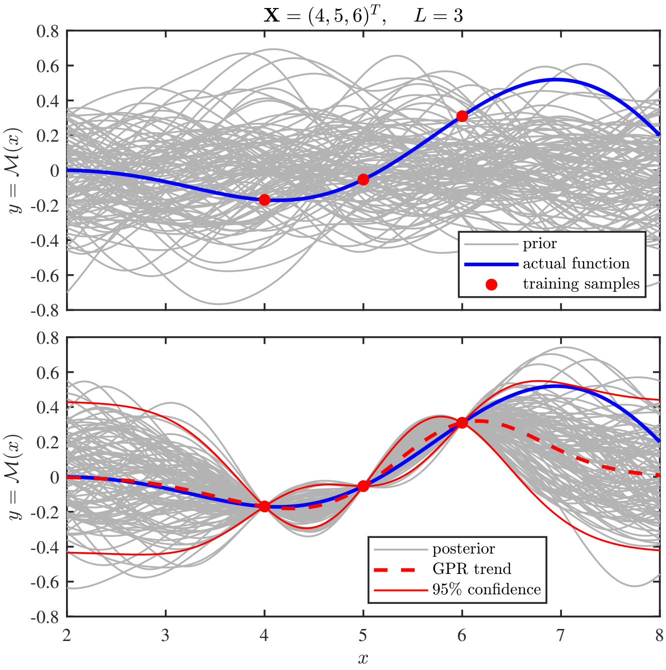


Fig. 1. Illustration of classical GPR. Given a target function (blue line) and some samples thereof (red dots), a prior GP (gray lines in the top panel) is conditioned on the observations, resulting in a posterior GP that interpolates the available data (gray lines in the bottom panel). The dashed and solid red lines are the mean (or trend) and the 95% confidence bounds of the posterior, respectively.

As an illustrative example, we consider the univariate (hence, $d = 1$) function

$$y = \mathcal{M}(x) = e^{-10/x}\left(2\cos(x) + \sin(x)\right), \tag{13}$$

which is deliberately inspired to the response of an under-damped second-order circuit. We consider a zero-mean trend for the prior. With training samples $\boldsymbol{X} = (3, 4, 5)^{\mathsf{T}}$, and corresponding observations $\boldsymbol{y} = (-0.170, -0.053, 0.310)^{\mathsf{T}}$, the kernel hyperparameter is estimated to be $\sigma_l = 0.75$, leading to $\sigma_f = 0.22$. Figure 1 shows, with a blue curve, the actual function (13), and with red dots the training samples. The gray lines in the top panel are some realizations of the prior. The bottom panel shows instead the result of conditioning the prior to interpolate the training samples. The gray lines are now realizations of the posterior, whereas the dashed red line is the model trend (2) and the solid red lines are its 95% confidence bounds, corresponding to $m(\boldsymbol{x}) \pm 1.96 \cdot \sqrt{c(\boldsymbol{x}, \boldsymbol{x})}$. It is interesting to note that the actual function lies within the confidence bounds. It should be noted, however, that these confidence bounds are only an estimate. The actual confidence is subject to the model accuracy and to the assumptions on the prior. We will further elaborate on this later on.

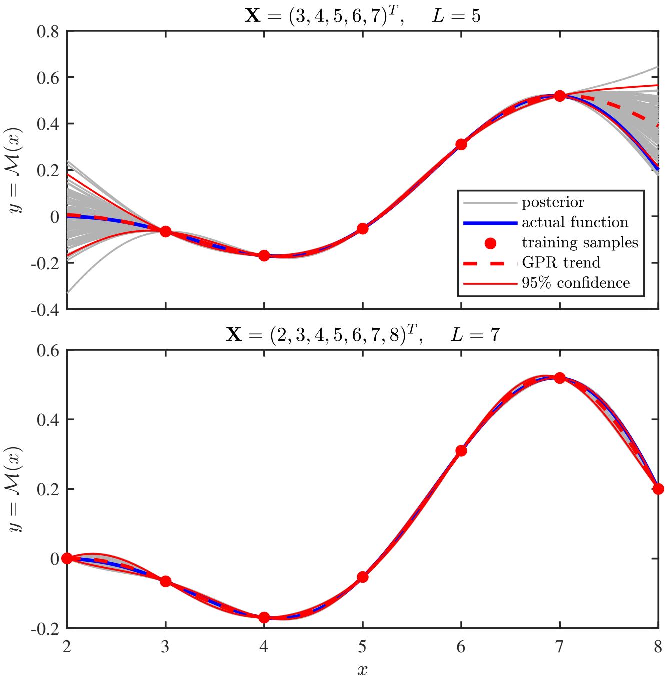


Fig. 2. GPR models of function (13) trained with $L = 5$ (top panel) and $L = 7$ (bottom panel) samples.

By using instead training samples computed for $\boldsymbol{X} = (3,4,5,6,7)^\mathsf{T}$ and $\boldsymbol{X} = (2,3,4,5,6,7,8)^\mathsf{T}$, the optimal kernel parameters become $(\sigma_l, \sigma_f) = (1.57, 0.30)$ and $(1.39, 0.29)$, respectively, resulting in the GPR models shown Fig. 2. A considerable reduction of the model uncertainty, reflected by the progressive narrowing of the confidence bounds, is achieved by increasing the number of training samples.

III. Application to Uncertainty Quantification

We now assume that the input parameters $\boldsymbol{x}$ are uncertain with an associated joint probability density function (PDF) $\rho(\boldsymbol{x})$, and we aim at performing UQ of the output y of system (1). The most common measures of uncertainty are the first two statistical moments (i.e., mean and variance), as well as PDFs. The mean and variance of y are defined as

$$\mu_y = \mathrm{E}\{y\} = \int_{\mathbb{R}^d} \mathcal{M}(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} \tag{14}$$

and

$$\begin{aligned}\sigma_y^2 = \mathrm{Var}\{y\} &= \mathrm{E}\{(y - \mathrm{E}\{y\})^2\}\\ &= \int_{\mathbb{R}^d} \left(\mathcal{M}(\boldsymbol{x}) - \mu_y\right)^2 \rho(\boldsymbol{x})d\boldsymbol{x}\\ &= \mathrm{E}\{y^2\} - (\mathrm{E}\{y\})^2\\ &= \int_{\mathbb{R}^d} \left(\mathcal{M}(\boldsymbol{x})\right)^2 \rho(\boldsymbol{x})d\boldsymbol{x} - \mu_y^2 \end{aligned} \tag{15}$$

respectively. However, the computation of the above integrals quickly becomes prohibitive as the dimensionality d increases, and/or (1) is expensive to compute. Moreover, a closed-form solution for the PDF $f_y(y)$ is available only for the univariate case and if $\mathcal{M}(x)$ is analytical [64].

More conveniently, the above statistical information is estimated by considering a (large) number of random configurations $\{\boldsymbol{x}_i\}_{i=1}^N$ of the uncertain inputs, and by observing the corresponding outputs $\{y_i\}_{i=1}^N$, with $y_i = \mathcal{M}(\boldsymbol{x}_i)\ \forall i$. In this case, the actual moments are estimated using the *sample* mean and variance, i.e.,

$$\mu_y \approx \hat{\mu}_y = \frac{1}{N}\sum_{i=1}^N y_i \tag{16}$$

and

$$\sigma_y^2 \approx \hat{\sigma}_y^2 = \frac{1}{N}\sum_{i=1}^N (y_i - \mu_y)^2 = \frac{1}{N-1}\sum_{i=1}^N \left(y_i - \hat{\mu}_y\right)^2, \tag{17}$$

respectively, whereas the PDF of y can be obtained by binning the samples into a histogram or using kernel density estimates. This is the essence of the MC analysis. It is important to remark that the above estimates are random variables themselves, with a variance that reduces only for large values of N, typically on the order of several thousands. This is why MC also becomes computationally prohibitive when (1) is expensive to evaluate.

In this context, surrogate models come to our rescue, by providing a computationally cheap alternative to the original model. In this process, however, they are often assumed to be error free, and the uncertainty due to the model error is typically neglected. In the case of a GPR model, a trivial and naive approach would be to sample the posterior trend (2) in place of the original model. This would provide the most likely prediction of the MC samples, from which statistical estimates are readily obtained. However, the precious information about the model confidence would unavoidably be lost.

In order to retain this information, we should rather treat the GPR model as a stochastic function and sample many realizations (or "trajectories") of the posterior with a common set of realizations of the input parameters. The process is illustrated in Fig. 3 for the function (13), when the normal distribution $\mathcal{N}(5,1)$ is assigned to x. The top and bottom panels refer to the sampling of the GPR models trained with $L = 3$ samples (cfr. bottom panel of Fig. 1) and $L = 5$ samples (top panel of Fig 2). The trajectories are displayed as lines, whereas each sample is represented by a dot. The blue and red dots are the samples computed by sampling the actual function (blue line) and the GPR trend (dashed red line), respectively. The gray dots are the samples computed for some random realizations of the posterior, shown by the gray lines. A different set of MC samples is obtained for each trajectory. As a result of the Gaussian distribution of both the input and output variables, the samples tend to crowd around the mean value of x, as well as around the posterior trend, consistently with the confidence bounds shown in Figs. 1 and 2. This information can be exploited to calculate many statistical estimates and assess their dispersion, thus obtaining information on their confidence. The dispersion of the estimates (i.e., their uncertainty) is expected to reduce when increasing the number of training samples, consistently

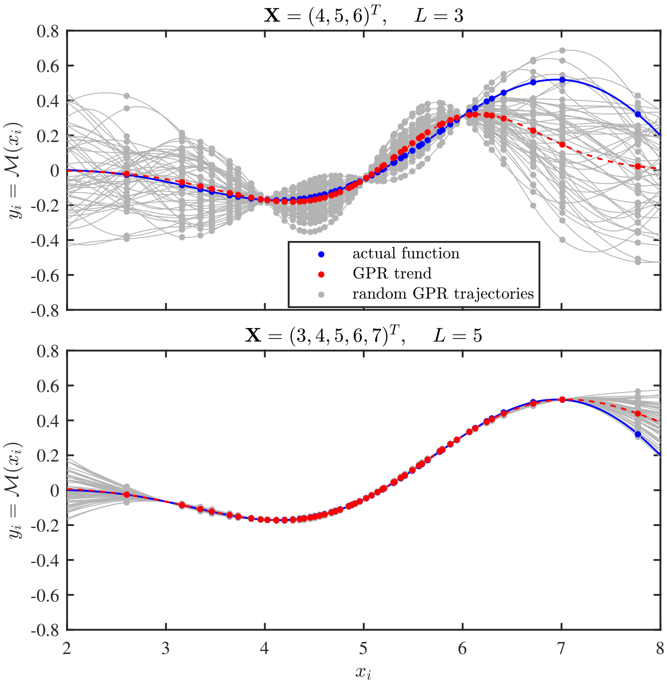


Fig. 3. MC sampling of function (13) and of its GPR models trained with $L = 3$ (top panel) and $L = 5$ (bottom panel) samples. The red and gray dots are the samples computed by evaluating the posterior trend (dashed red line) and other random trajectories (gray lines), respectively. The blue dots are the corresponding samples of the actual function (blue line).

with the reduction in the model uncertainty that is observed in Figs. 1 and 3. The calculation of statistical estimates is discussed in detail in the next section.

IV. Statistical Estimates with GPR Models

This section discusses the calculation of GPR estimates of statistical moments and their uncertainty, starting from analytical derivations and introducing successive approximations.

A. Analytical Estimates

As opposed to classical MC estimates, the UQ of the GPR posterior (4) is complicated by the fact that we are applying a stochastic, rather than a deterministic function to a stochastic variable. Therefore, we have two sources of uncertainty: the input parameters, and the GPR model itself. In order to make this explicit, we express the posterior GP using the Karhunen-Loève transform [65], which allows recasting (4) as

$$\hat{\mathcal{M}}(\boldsymbol{x}) = m(\boldsymbol{x}) + \sum_{q=1}^{\infty} \sqrt{\lambda_q} \xi_q \Psi_q(\boldsymbol{x}) \tag{18}$$

where $\boldsymbol{\xi} = \{\xi_q\}_{q=1}^{\infty}$ is a discrete, though infinite set of independent standard normal random variables, whereas λ_q and $\Psi_q(\boldsymbol{x})$ are the eigenvalues and corresponding eigenfunctions of the posterior covariance $c(\boldsymbol{x}, \boldsymbol{x}')$, satisfying

$$\lambda_q \Psi_q(\boldsymbol{x}) = \int_{\mathbb{R}^d} c(\boldsymbol{x}, \boldsymbol{x}') \Psi_q(\boldsymbol{x}') d\boldsymbol{x}' \tag{19}$$

Hence, the aforementioned two sources of uncertainty are now explicitly identified as the input parameters $\boldsymbol{x}$ and the random variables $\boldsymbol{\xi}$. It should be noted that (18) can be in practice truncated to a finite set of components, as the eigenvalues usually decay very fast.

We can now define the GPR estimates of the mean μ_y as the expectation of (18) over $\boldsymbol{x}$, i.e.,

$$\begin{aligned}\mu_{\mathrm{GPR}} = \mathrm{E}_{\boldsymbol{x}}\left\{\hat{\mathcal{M}}(\boldsymbol{x})\right\} &= \int_{\mathbb{R}^d} m(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} \\ &+ \sum_{q=1}^{\infty} \sqrt{\lambda_q}\xi_q \int_{\mathbb{R}^d} \Psi_q(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\end{aligned} \tag{20}$$

It is important to remark that μ_{GPR} is still a random variable because of the random variables $\boldsymbol{\xi}$. Hence, the best estimate of μ_y is taken as the expectation of (20) over $\boldsymbol{\xi}$. Since $\mathrm{E}\{\xi_q\} = 0$ $\forall q$, this leads to

$$\mu_y \approx \mathrm{E}_{\boldsymbol{\xi}}\{\mu_{\mathrm{GPR}}\} = \int_{\mathbb{R}^d} m(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}, \tag{21}$$

which corresponds to the expected value computed over the GPR trend. Interestingly, it is possible to compute also the variance of μ_{GPR}, i.e.,

$$\begin{aligned}\mathrm{Var}_{\boldsymbol{\xi}}\{\mu_{\mathrm{GPR}}\} &= \mathrm{E}_{\boldsymbol{\xi}}\left\{(\mu_{\mathrm{GPR}} - \mathrm{E}_{\boldsymbol{\xi}}\{\mu_{\mathrm{GPR}}\})^2\right\} \\ &= \mathrm{E}_{\boldsymbol{\xi}}\left\{\left(\sum_{q=1}^{\infty}\sqrt{\lambda_q}\xi_q\int_{\mathbb{R}^d}\Psi_q(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\right)^2\right\} \\ &= \sum_{q=1}^{\infty}\lambda_q\left(\int_{\mathbb{R}^d}\Psi_q(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\right)^2\end{aligned} \tag{22}$$

where we have developed the square of the summation and used the fact that $\mathrm{E}\{\xi_q\xi_k\} = \delta_{qk}$ (i.e., the Kronecker's delta), being the random variables statistically independent.

Similarly, we calculate the GPR estimates of the variance σ_y^2 as

$$\sigma_{\mathrm{GPR}}^2 = \mathrm{E}_{\boldsymbol{x}}\left\{\left(\hat{\mathcal{M}}(\boldsymbol{x})\right)^2\right\} - \left(\mathrm{E}_{\boldsymbol{x}}\left\{\hat{\mathcal{M}}(\boldsymbol{x})\right\}\right)^2, \tag{23}$$

and we take the best estimate as the expectation of σ_{GPR}^2 over $\boldsymbol{\xi}$. Lengthy calculations lead to

$$\begin{aligned}\sigma_y^2 &\approx \mathrm{E}_{\boldsymbol{\xi}}\{\sigma_{\mathrm{GPR}}^2\} \\ &= \int_{\mathbb{R}^d} m^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} - \left(\int_{\mathbb{R}^d} m(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\right)^2 \\ &+ \sum_{q=1}^{\infty}\lambda_q\left[\int_{\mathbb{R}^d}\Psi_q^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} - \left(\int_{\mathbb{R}^d}\Psi_q(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\right)^2\right].\end{aligned} \tag{24}$$

It is interesting to note that the above estimate does *not* coincide with the variance computed along the GPR trend, which is given by only the first two terms in (24). The complexity of dealing with the infinite-dimensional representation (18) discourages the attempt of deriving similar closed-form relations for the variance of σ_{GPR}^2.

In practice, the eigenvelues λ_q and eigenfunctions $\Psi_q(\boldsymbol{x})$ can be effectively approximated by considering a discrete set of $\boldsymbol{x}$-values and calculating the eigendecomposition of the corresponding covariance matrix (6). For low-dimensional problems, the integrals involved in (21), (22), and (24) can be then evaluated using any numerical technique, such as a trapezoidal rule. Nonetheless, these semi-analytical estimates become intractable when d is large.

B. Single Monte Carlo Sampling

A first simplification is to approximate the expectations over $\boldsymbol{x}$ in a "MC way", i.e., by considering a finite number of samples of the input variables, $\{\boldsymbol{x}_i\}_{i=1}^N$, and using sample estimates. In this case, the corresponding samples of the posterior GP can be considered as a finite collection $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_N)^\mathsf{T}$ of N correlated Gaussian variables, with mean vector $\boldsymbol{m}$ and covariance matrix $\boldsymbol{C}$ calculated as in (5) and (6), and whose entries correspond to $m_i = m(\boldsymbol{x}_i)$ and $C_{ij} = c(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\forall i, j = 1, \ldots, N$.

The GPR estimates of the mean are approximated by the sample mean over the available realizations of $\boldsymbol{x}$, i.e.,

$$\mu_{\mathrm{GPR}} \approx \hat{\mu}_{\mathrm{GPR}} = \frac{1}{N}\sum_{i=1}^{N}\hat{\mathcal{M}}(\boldsymbol{x}_i) = \frac{1}{N}\sum_{i=1}^{N}\hat{y}_i \tag{25}$$

The above expression turns out to be a sum of correlated Gaussian variables. As such, it is itself Gaussian, with mean and variance given by [66]

$$\mathrm{E}\left\{\hat{\mu}_{\mathrm{GPR}}\right\} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{E}\left\{\hat{y}_i\right\} = \frac{1}{N}\sum_{i=1}^{N}m(\boldsymbol{x}_i) \tag{26}$$

and

$$\mathrm{Var}\left\{\hat{\mu}_{\mathrm{GPR}}\right\} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}C_{ij} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}c(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{27}$$

Similarly to the results in Section IV-A, the best GPR estimate of μ_y corresponds to the sample mean calculated over the GPR trend. Compared to (21) and (22), the approximation in (26) and (27) is related to considering a finite number of realizations of $\boldsymbol{x}$.

Along the lines of the previous derivations, we introduce the GPR estimates of the variance, which are approximated as the sample variance of the posterior calculated over the realizations of $\boldsymbol{x}$:

$$\begin{aligned}\sigma^2_{\mathrm{GPR}} \approx \hat{\sigma}^2_{\mathrm{GPR}} &= \frac{1}{N-1}\sum_{i=1}^{N}\left(\hat{\mathcal{M}}(\boldsymbol{x}_i) - \mathrm{E}\left\{\hat{\mu}_{\mathrm{GPR}}\right\}\right)^2\\ &= \frac{1}{N-1}\sum_{i=1}^{N}\left(\hat{y}_i - \frac{1}{N}\sum_{i=1}^{N}m(\boldsymbol{x}_i)\right)^2\end{aligned} \tag{28}$$

With reference to Appendix A, $\hat{\sigma}^2_{\mathrm{GPR}}$ turns out to be a quadratic form in $\boldsymbol{\varepsilon} = \hat{\boldsymbol{y}} - \mathrm{E}\left\{\hat{\mu}_{\mathrm{GPR}}\right\}$, with $\boldsymbol{\Lambda} = \frac{1}{N-1}\cdot\mathbf{1}_{N\times N}$, where $\mathbf{1}_{N\times N}$ denotes the $N\times N$ identity matrix. As such, the mean and variance of the variance estimate $\hat{\sigma}^2_{\mathrm{GPR}}$ are found as [66]

$$\begin{aligned}\mathrm{E}\left\{\hat{\sigma}^2_{\mathrm{GPR}}\right\} =& \frac{1}{N-1}\sum_{i=1}^{N}\left(m(\boldsymbol{x}_i) - \mathrm{E}\left\{\hat{\mu}_{\mathrm{GPR}}\right\}\right)^2\\ &+ \frac{1}{N-1}\mathrm{tr}(\boldsymbol{C})\end{aligned} \tag{29}$$

and

$$\begin{aligned}&\mathrm{Var}\left\{\hat{\sigma}^2_{\mathrm{GPR}}\right\}\\ &= \frac{4}{(N-1)^2}(\boldsymbol{m} - \mathrm{E}\left\{\hat{\mu}_{\mathrm{GPR}}\right\})^\mathsf{T}\boldsymbol{C}(\boldsymbol{m} - \mathrm{E}\left\{\hat{\mu}_{\mathrm{GPR}}\right\})\\ &\quad + \frac{2}{(N-1)^2}\mathrm{tr}\left(\boldsymbol{C}^2\right)\end{aligned} \tag{30}$$

respectively. Like in the analytical case (24), the expectation of the variance estimate differs from the variance computed by sampling the GPR trend. Moreover, it should be noted that the distribution of $\hat{\sigma}^2_{\mathrm{GPR}}$ is generally not Gaussian.

Table I provides the mean and variance of (13) estimated with the GPR models trained using $L = 3$, $L = 5$, and $L = 7$ samples, which were shown in Figs. 1 and 2. The analytical estimates and the results of single MC sampling, computed using 1000 samples of the input parameter x, are both provided. The estimates are given in terms of the interval of the expectation ± 1.96 times the standard deviation, corresponding to a 95% confidence for a Gaussian distribution. An exception is the analytical estimate of the variance, for which only the result of the expectation (24) is provided, since no closed-form formula is available for its standard deviation. The exact values computed with (14) and (15) are also provided for comparison. It is remarkable that the exact result is always within the confidence interval, except for the single MC estimates of the mean for the models with $L = 5$ and $L = 7$ training samples, which are (slightly) overestimated. This is due to the finite precision of the MC estimate itself. Moreover, it is once again noted that the confidence interval narrows around the exact value as the number of training samples is increased.

In order to further investigate the confidence of the GPR predictions, we repeat the experiment by randomly draw the training samples, and we check how often the MC estimates actually lie within the predicted GPR confidence bounds. The results, based on 10000 runs each, are reported in Table II. It is noted that the actual figure is quite consistent with the targeted 95% confidence, especially when L is increased. This is reasonable if we consider that the confidence bound is itself an estimate that depends on the model accuracy.

C. Double Monte Carlo Sampling

The estimates introduced in the previous sections only concern the first two statistical moments, while they do not provide information on the PDF $f_y(y)$. Therefore, a feasible approach is to consider also a finite number of posterior trajectories, and to sample them using the same realization $\{\boldsymbol{x}_i\}_{i=1}^N$ of the uncertain inputs. This is equivalent to drawing a finite number of samples from the random vector $\hat{\boldsymbol{y}}$. This allows generating many instances of the PDF, and assess its uncertainty. We refer to this approach as "double MC sampling", as we sample both the input parameters and the posterior. Of course, this approach can also be used to further approximate the estimates of the mean and variance, and in particular to assess the actual distribution of the latter which, as already noted, is in general non-Gaussian.

Figure 4 shows the PDFs of y obtained with four different GPR models. In addition to the models computed with $L = 3$,

TABLE I

ANALYTICAL AND SINGLE MC ESTIMATES COMPUTED FOR THE FUNCTION (13) WITH GPR MODELS TRAINED WITH DIFFERENT NUMBERS OF SAMPLES.

		GPR estimates					
		$L = 3$		$L = 5$		$L = 7$	
Moment	Exact	Analytical	Single MC	Analytical	Single MC	Analytical	Single MC
Mean of (13)	0.0381	$[-0.0123, 0.0492]$	$[-0.0123, 0.0492]$	$[0.0381, 0.0401]$	$[0.0382, 0.0401]$	$[0.0381, 0.0382]$	$[0.0382, 0.0383]$
Variance of (13)	0.0448	0.0376	$[0.0232, 0.0527]$	0.0453	$[0.0447, 0.0460]$	0.0448	$[0.0447, 0.0450]$

TABLE II

ESTIMATE OF THE ACTUAL GPR CONFIDENCE BASED ON MULTIPLE RUNS WITH RANDOM TRAINING SAMPLES (TARGETED CONFIDENCE: 95%).

	$L = 3$	$L = 5$	$L = 7$
Mean	86.3%	97.9%	88.2%
Variance	58.0%	97.5%	91.5%

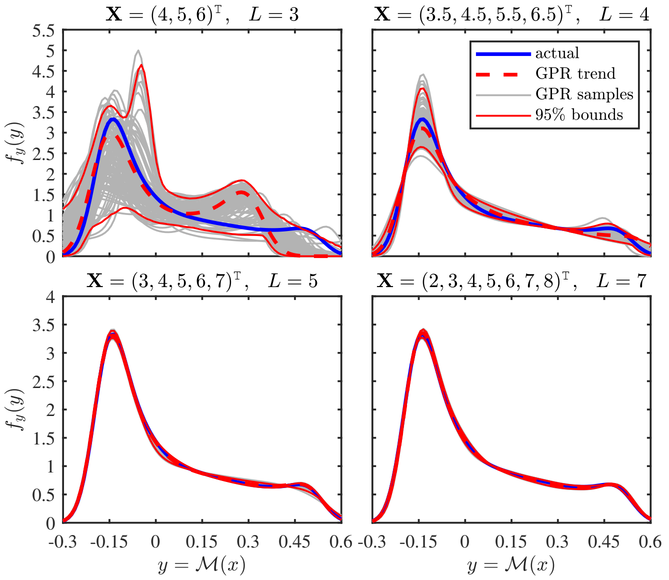


Fig. 4. PDF of (13). The result of direct MC sampling (blue line) is compared against the estimates obtained by sampling the trend of GPR models constructed with different sets of training samples (dashed red lines). The gray lines are the PDFs obtained for some realizations of the posterior trajectories and show the dispersion of the GPR estimate. The solid red lines are the corresponding 95% confidence bounds.

$L = 5$, and $L = 7$ training samples, shown in Figs. 1 and 2, a model with $L = 4$ samples within $[3.5, 6.5]$ is considered (top-right panel). The blue line represents the actual PDF estimated by sampling (13) with 1000 MC samples. The dashed red and the gray lines are the PDFs obtained by sampling, with the same set of MC samples, the GPR trend and some realizations of the posterior trajectories, respectively. Finally, the solid red lines are the 95% confidence bounds of the GPR distributions, obtained by considering 5000 posterior realizations. Also in this case, the reference result mostly lies within the confidence bounds, which reduce by increasing the number of training samples.

V. PRINCIPAL COMPONENT ANALYSIS COMPRESSION

The framework introduced so far applies to a system with an input vector $\boldsymbol{x}$ of arbitrary dimension, but single (scalar) output y. However, in realistic application scenarios, the target function is often a set of multiple responses of a dynamical system. In this section, we apply the PCA compression introduced in [58] to effectively deal with multi-output systems. We show that the linearity of the PCA combines well with the Gaussian model of the surrogate output, making it possible to obtain the GPR models of the original outputs directly from those of the principal components.

To this end, we consider the multi-output system

$$\boldsymbol{y} = \mathcal{M}(t; \boldsymbol{x}) \tag{31}$$

where $\boldsymbol{y} = (y_1, \ldots, y_S)^{\mathsf{T}} \in \mathbb{R}^S$ and $\mathcal{M} : \mathbb{R}^d \to \mathbb{R}^S$. In practice, the system response is evaluated for a discrete set of time points. Since for our analysis it is irrelevant whether data refer to different output components and/or different time points, we assume that $\boldsymbol{y}$ generically denotes the ensemble of available output data for all components and time points of interest. A naive approach would be to train a separate GPR model for each component of $\boldsymbol{y}$, which soon becomes intractable when the goal is to perform UQ of a large number of output variables evaluated at a large number of time points, as the dataset size S would be enormous.

Indeed, the training set now consists of a matrix $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L) \in \mathbb{R}^{S \times L}$, with $\boldsymbol{y}_l = \mathcal{M}(t, \boldsymbol{x}_l)$ $\forall l$ and large S. An effective strategy is to apply a PCA compression [67], which allows approximating $\boldsymbol{Y}$ with a truncated singular value decomposition (SVD) [58]. This leads to

$$\boldsymbol{Y} = \bar{\boldsymbol{y}} + \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{T}} \approx \bar{\boldsymbol{y}} + \tilde{\boldsymbol{U}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{V}}^{\mathsf{T}} = \bar{\boldsymbol{y}} + \tilde{\boldsymbol{U}}\tilde{\boldsymbol{Z}}, \tag{32}$$

where $\bar{\boldsymbol{y}} \in \mathbb{R}^S$ is the columnwise dataset mean

$$\bar{\boldsymbol{y}} = \frac{1}{L}\sum_{l=1}^{L} \boldsymbol{y}_l, \tag{33}$$

whereas $\tilde{\boldsymbol{U}} \in \mathbb{R}^{S \times \tilde{n}}$, $\tilde{\boldsymbol{S}} \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$, $\tilde{\boldsymbol{V}} \in \mathbb{R}^{L \times \tilde{n}}$ are the truncated matrices of an "economy-size" SVD of matrix $\tilde{\boldsymbol{Y}}$ having columns $\tilde{\boldsymbol{y}}_l = \boldsymbol{y}_l - \bar{\boldsymbol{y}}$, for $l = 1, \ldots, L$. The truncation is determined based on a relative threshold on the singular values, which allows to rigorously control the accuracy of the PCA approximation (32). In this work, we set the threshold to 1% of the largest singular value. Since the various outputs of the same system at different time points exhibit a certain amount of interdependency, a substantial compression (i.e., $\tilde{n} \ll S$) is usually achieved.

In this context, matrix

$$\tilde{\boldsymbol{Z}} = \tilde{\boldsymbol{S}}\tilde{\boldsymbol{V}}^{\mathsf{T}} = \tilde{\boldsymbol{U}}^{\mathsf{T}}(\boldsymbol{Y} - \bar{\boldsymbol{y}}) \in \mathbb{R}^{\tilde{n}\times L} \tag{34}$$

can be interpreted as a collection of L training responses of a reduced system

$$\boldsymbol{z} = \mathcal{M}_{\text{PCA}}(\boldsymbol{y}) = \tilde{\boldsymbol{U}}^{\mathsf{T}}(\boldsymbol{y} - \bar{\boldsymbol{y}}) = \tilde{\boldsymbol{U}}^{\mathsf{T}}(\mathcal{M}(t;\boldsymbol{x}) - \bar{\boldsymbol{y}}) \tag{35}$$

with only $\tilde{n}$ output components, where $\mathcal{M}_{\text{PCA}} : \mathbb{R}^S \to \mathbb{R}^{\tilde{n}}$. Hence, each of these “principal components” can now be modeled effectively with an individual GPR surrogate, yielding

$$z_n \approx \tilde{\mathcal{M}}_n(\boldsymbol{x}) \sim \mathcal{GP}(m_n(\boldsymbol{x}), c_n(\boldsymbol{x},\boldsymbol{x}')|\boldsymbol{\theta}_n), \tag{36}$$

for $n = 1, \ldots, \tilde{n}$. Since the principal components are zero-meaned by construction, it is particularly reasonable to consider a prior with $\mu(\boldsymbol{x}) = 0$ for the model (36).

From the reduced output $\boldsymbol{z}$, the original output $\boldsymbol{y}$ is recovered by inverting (35):

$$\boldsymbol{y} = \mathcal{M}(t;\boldsymbol{x}) \approx \mathcal{M}_{\text{PCA}}^{-1}(\boldsymbol{z}) = \bar{\boldsymbol{y}} + \tilde{\boldsymbol{U}}\boldsymbol{z} \tag{37}$$

Therefore, using the properties for linear combinations of random variables, the GPR model for the s-th component of $\boldsymbol{y}$ is obtained directly from the model of the principal components as [66]

$$\begin{aligned} y_s \approx \hat{\mathcal{M}}_s(\boldsymbol{x}) &= \bar{y}_s + \sum_{n=1}^{\tilde{n}} U_{sn}\tilde{\mathcal{M}}_n(\boldsymbol{x}) \qquad (38)\\ &\sim \mathcal{GP}\left(\bar{y}_s + \sum_{n=1}^{\tilde{n}} U_{sn}m_n(\boldsymbol{x}), \sum_{n=1}^{\tilde{n}} U_{sn}^2 c_n(\boldsymbol{x},\boldsymbol{x}')\,|\boldsymbol{\theta}_n\right) \end{aligned}$$

$\forall s = 1, \ldots, S$, and allows computing statistical estimates as described in Sections IV-A and IV-B. Alternatively, one can directly sample the posteriors (36) and recover samples of the original outputs via (37), for use in a “double MC analysis” as described in Section IV-C.

VI. Illustrative Example: RLC Circuit

Before applying the proposed framework to real-life application scenarios, we first illustrate the performance of PCA compression based on the trivial case of a parallel RLC circuit excited by a deterministic dc current source I_0. According to the modified nodal analysis (MNA) formulation [68], the system is described by the set of ordinary differential equations

$$\begin{pmatrix} C & 0 \\ 0 & -L \end{pmatrix} \frac{d}{dt}\begin{pmatrix} v_C(t) \\ i_L(t) \end{pmatrix} = \begin{pmatrix} 1/R & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} v_C(t) \\ i_L(t) \end{pmatrix} + \begin{pmatrix} I_0 \\ 0 \end{pmatrix} \tag{39}$$

where v_C is the voltage across all elements and i_L is the current flowing through the inductor.

For the ease of visualization, we consider again the case of a single uncertain parameter x, affecting however all three passive components simultaneously. This could be the case, for example, of the operating temperature. In order to stress the proposed method, we assume a ±60% variability of the elements around the nominal value, which we express as

$$\begin{aligned} R &= \bar{R}(1 + 0.6x)\\ L &= \bar{L}(1 + 0.6x)\\ C &= \bar{C}(1 + 0.6x) \end{aligned}$$

with $x \in [-1, 1]$. For the simulation, we consider $\bar{R} = 10\ \Omega$, $\bar{L} = 100$ nH, $\bar{C} = 1$ nF, and $I_0 = 5$ A. Under the above assumptions, the circuit exhibits both over- and under-damped behavior, depending on the value of x.

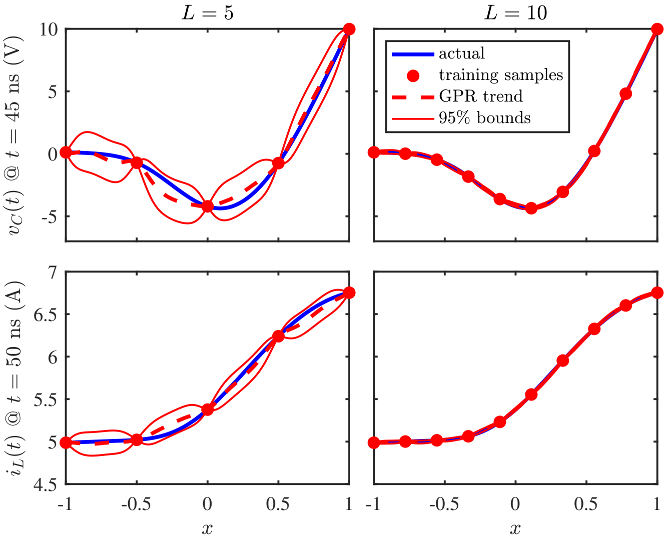


Fig. 5. Capacitor voltage at $t = 45$ ns (top panels) and inductor current at $t = 50$ ns (bottom panels) as a function of parameter x. Blue lines: actual value; red markers: training samples; dashed red lines: mean GPR prediction (trend); solid red lines: 95% confidence bounds of GPR prediction. Left and right panels refer to GPR models trained with $L = 5$ and $L = 10$ samples, respectively.

We evaluate the step response by solving (39) with null initial conditions for 501 equally-spaced time points between [0, 150] ns, leading to an output size of $S = 1002$. Figure 5 shows, with blue lines, the parametric variation, as a function of x, of the capacitor voltage at $t = 45$ ns (top panels) and of the inductor current at $t = 50$ ns (bottom panels).

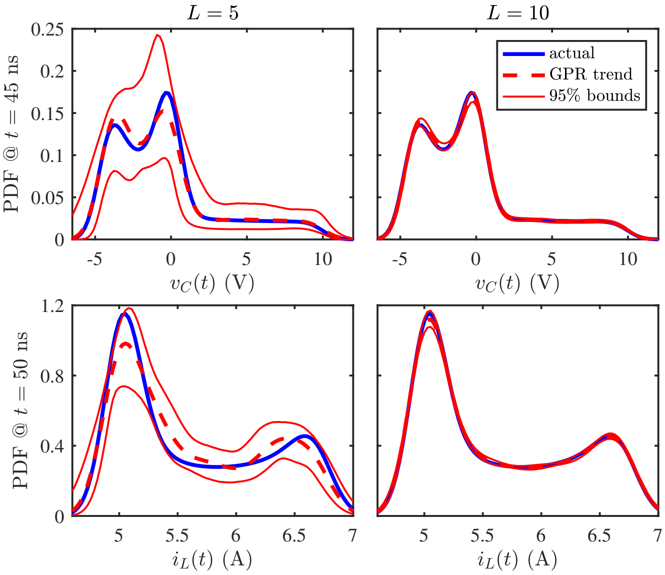


Fig. 6. PDF of the capacitor voltage and inductor current for a uniform variability of x. The result from MC analysis (blue line) is compared against the mean GPR prediction of the MC samples (dashed red line). The solid red lines represent the 95% confidence bounds of the GPR distributions. Panels refer to the same time instants and GPR models as in Fig. 5.

Next, we apply GPR by using $L = 5$ and $L = 10$ training responses, computed for equally-spaced values of x between $[-1, +1]$ and indicated by the red dots in Fig. 5. Specifically, we apply PCA compression (35) to these responses, which leads to $\tilde{n} = 4$ for the dataset with $L = 5$ samples and to $\tilde{n} = 7$ for the one with $L = 10$ samples, and we train individual GPR models for the principal components. It is important to remark that these $\tilde{n}$ GPR models comprise the information of both the capacitor voltage and the inductor current at any of the 501 simulated time points. The dashed red lines in Fig. 5 show the prediction of the parametric variation of the capacitor voltage and inductor current provided by the GPR trend, recovered from the trend of the compressed variables according to (38). Moreover, the solid red lines are the 95% confidence bounds of the model. Once again, it is shown that the actual output value is well within the confidence bounds, which substantially reduce by increasing the number of training samples.

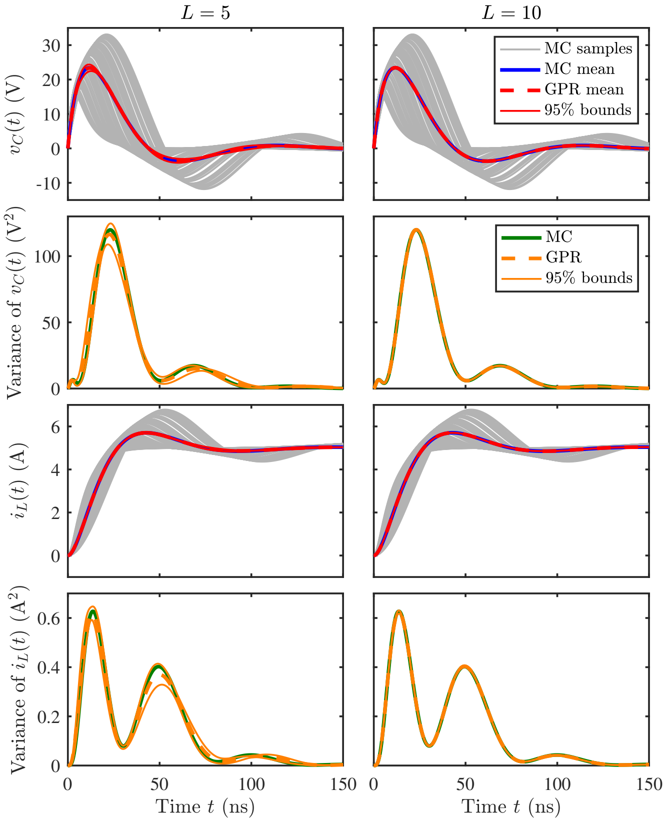


Fig. 7. Step response of the capacitor voltage and inductor current. Panels in the first and third rows show a subset of MC samples (gray lines), the mean of the MC samples (blue lines), as well as the GPR estimate of the mean (dashed red line) and its 95% confidence bounds (solid red lines). Panels in the second and fourth rows show the variance of the MC samples (green lines), the corresponding GPR estimate (dashed orange lines), and its 95% confidence bounds (solid orange lines). Left and right panels refer to GPR models trained with $L = 5$ and $L = 10$ samples, respectively.

Furthermore, Fig. 6 shows the PDFs of the same variables when a uniform distribution is ascribed to x. The distribution of 1000 MC simulations of the original system (39) (blue line) is compared against the result obtained by predicting the value of the MC samples with the GPR trend (dashed red lines). This is what is normally done when exploiting GPR as a deterministic surrogate model. It can be seen that the prediction based on $L = 5$ training samples (left panels) is quite off, especially for the inductor current. However, if we take into account the uncertainty of the GPR model, we obtain the 95% confidence bounds shown by the solid red lines, which are indeed rather large and include the actual PDF. By increasing the number of training samples to $L = 10$ (right panels), the confidence bounds reduce significantly, and the GPR prediction compares well with the MC result.

Finally, Fig. 7 shows the entire transient response of the capacitor voltage and inductor current. The gray lines in the panels of the first and third rows are a subset of responses from the MC analysis, whereas the blue line is the mean of the MC samples. The dashed and solid red line show instead the GPR estimate of the mean and its 95% confidence bounds, respectively. The panels in the second and forth rows refer to the variance instead. Specifically, the green line is the variance of the MC samples, the dashed orange line is the corresponding GPR prediction, and the solid orange lines are its 95% confidence bounds. Left and right panels refer to the GPR models trained with $L = 5$ and $L = 10$ samples, respectively. For the model trained with $L = 5$ samples, relatively large confidence bounds are observed especially for the variance. Nevertheless, the prediction compares well with the MC estimate. As already noted, the uncertainty of the GPR model with $L = 10$ training samples is much lower, thereby leading to narrow confidence bounds that indicate a very high accuracy of the GPR statistical estimates.

VII. Application Examples

In this section, the advocated GPR-based framework for UQ is applied to two realistic designs, namely a low-noise amplifier and a digital electronic link.

A. Low-Noise Amplifier

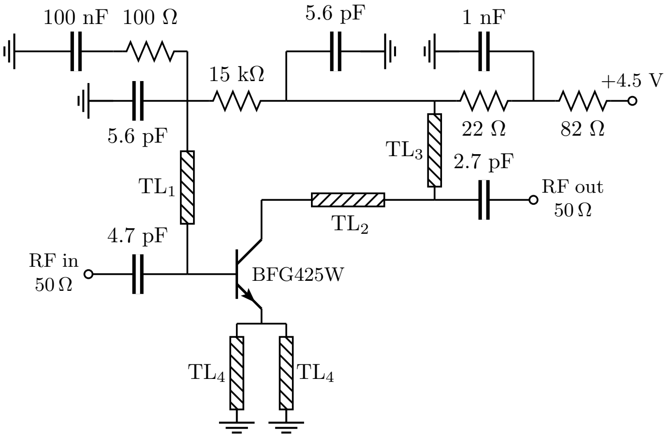


Fig. 8. Schematic of the 2-GHz low-noise amplifier.

The first application example considers the low-noise amplifier with the schematic shown in Fig. 8. The amplifier is designed for 2-GHz operation and to exhibit a gain of 16 dB when operating at a collector current of 5 mA [69]. For the BFG425W bipolar junction transistor (BJT), a standard SPICE model available by the vendor is used. The uncertainty is provided by $d = 21$ parameters, namely the forward current gain of the BJT, some of its parasitics, as well as all the external resistors and capacitors shown in Fig. 8. All these parameters are ascribed a Gaussian distribution with a 10% relative standard deviation around their nominal values.

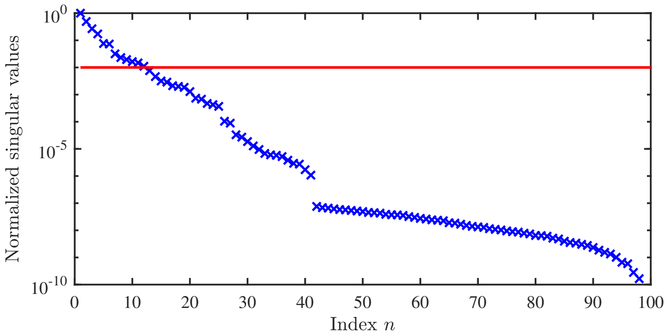


Fig. 9. Normalized singular values (markers) of the training dataset for the low-noise amplifier test case. The red line indicates the 1% threshold for the PCA truncation.

The amplifier is simulated with the HSPICE harmonic balance solver to find the steady-state output power at 201 time points for a +10-dBm input tone of 2 GHz. Hence, $S = 201$ in this case. Reference results are computed with a MC analysis that considers 1000 samples generated according to a LHS design. Moreover, training responses for the GPR are calculated for $L = 100$ parameter samples from a Sobol sequence [11], [70]. An inverse probability transform [71] is applied to obtain samples for the Gaussian distribution, starting from the canonical uniform sequence.

Figure 9 shows the normalized singular values resulting from the SVD of the training dataset. It is found that the singular values drop below the 1% threshold (red line) for $\tilde{n} = 12$. A compression rate of 94% is thus achieved. Higher compression rates are achieved for larger problems, as will be shown by the next application example.

Figure 10 provides in the top panel a subset of MC samples (gray lines) of the output power, as well as the average power estimated from the MC samples (blue line) and with the GPR model (dashed red line). The solid red lines are the 95% confidence bounds of the GPR estimate. The bottom panel provides, with a green line, the MC estimate of the variance, and with solid and orange lines the corresponding GPR estimate and its 95% confidence bounds, respectively. Very accurate estimates, as confirmed by the tightness of the confidence bounds, are obtained with only 100 training samples.

Finally, Fig. 11 shows the PDF of the output power at time $t = 0.18$ ns (top panel) and $t = 0.52$ ns (bottom panel). The distribution of the MC samples (blue lines) is compared against the distribution of the same samples predicted with the GPR trend (dashed red line). The solid red lines are instead the 95% confidence bounds of the GPR prediction. Also in this case, the confidence bounds are relatively tight, and the MC estimate lies within them.

For the sake of comparison, the Fig. 11 further shows the predictions obtained with a LS-SVM model using a squared-exponential kernel (dotted green lines), and with an adaptive sparse PCE (dash-dotted orange lines). These two models are trained by means of the LS-SVMlab [72] and UQLab [73] toolboxes, respectively, leveraging the same observations as used for the GPR surrogate. It is interesting to note that the LS-SVM and the GPR trend very provide similar results. Indeed, it is possible to demonstrate that the mathematical formulation of the posterior mean of a GPR with a noise variance $\sigma_n^2 \neq 0$ and the LS-SVM regression are in fact equivalent [74]. On the other hand, for the specific case in which $\sigma_n^2 = 0$, the posterior mean estimated via the GPR turns out to be equivalent to a LS-SVM regression in which the regularizer term is neglected (i.e., the hyperparameter $\gamma \to \infty$ [49]). The PCE distribution exhibits instead a larger discrepancy, although it is also within the GPR confidence bounds. This comparison shows that, for this example, the GPR surrogate provides similar results compared to state-of-the-art surrogate models, while providing the additional information on the prediction uncertainty.

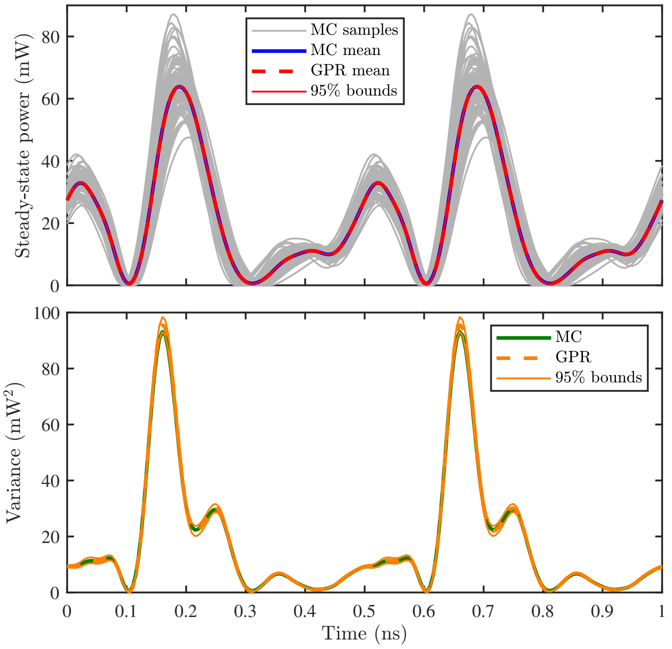


Fig. 10. Steady-state output power for the low-noise amplifier of Fig. 8. Top panel: subset of MC samples (gray lines), MC mean (blue line), GPR estimate of the mean and its 95% confidence bounds (solid and dashed red lines, respectively). Bottom panel: MC variance (green line), GPR estimate of the variance (dashed orange line) and its 95% confidence bounds (solid orange lines).

B. Digital Electronic Link

The second application example refers to the electronic link investigated in [19]. It describes a node-to-node communication link between a driver and a receiver, including a number of distributed and lumped discontinuities such as transmission lines and vias. We consider $d = 26$ uncertain parameters, namely all independent electrical, geometrical, and material parameters, which are ascribed a Gaussian distribution with a 10% relative standard deviation. The Reader is referred to [19] for the schematic, component values, and further information.

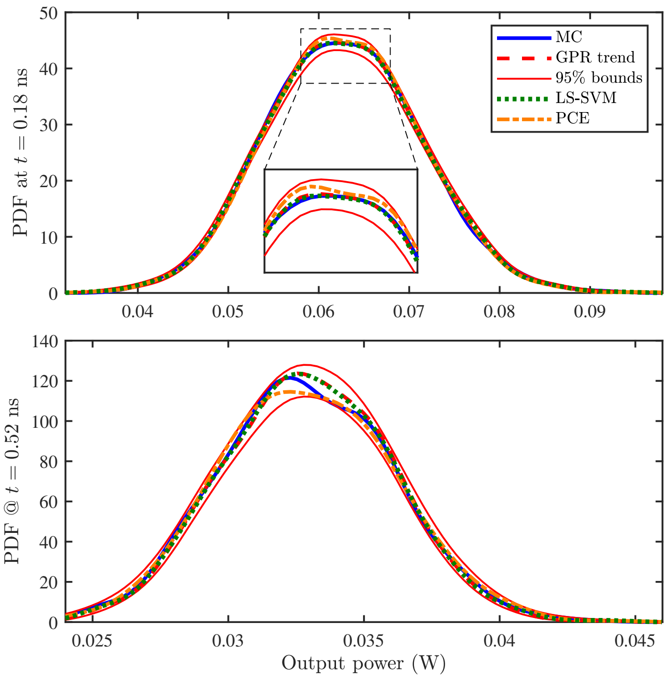


Fig. 11. PDF of the steady-state output power at two different time instants. Blue line: PDF of the MC samples; dashed red line: PDF computed by sampling the GPR trend; solid red lines: 95% bounds of the distributions obtained from the GPR model; dotted green and dash-dotted orange lines: distributions obtained with a LS-SVM and a sparse PCE model, respectively.

For the simulation, a behavioral macromodel [75] of a 133-MHz DDR memory with supply voltage of $V_{DD} = 1.8$ V is connected at the transmitter side with the inclusion of the power rail network and package parasitics, and it is used to transmit a pseudo-random bit sequence. Five outputs are considered, namely four crosstalk voltages at the two intermediate coupled transmission-line sections and the voltage at the receiver side. These outputs are observed at 4001 time points over a window of 200 ns. Therefore, the output size in this case is $S = 20005$.

Reference results are generated by means of a MC simulation with 1000 samples drawn according to a LHS strategy. The training responses for the GPR model are instead computed for $L = 300$ Gaussian-distributed samples of the uncertain parameters, generated with a Sobol sequence. The PCA compression reduces this training dataset to $\tilde{n} = 37$ components only, thereby achieving a compression rate of 99.8%.

Figure 12 shows the received voltage, limited to a 16-bit window for the sake of readability. In the top panel, the gray lines are a subset of samples from the MC simulation. The blue line is the average of the received voltage computed from the MC samples, whereas the dashed and solid red lines are the GPR mean and its 95% confidence bounds, respectively. The bounds are tight enough to be indistinguishable from the prediction itself. The bottom panel shows instead the variance of the voltage computed from the MC samples (green line) as well as the GPR prediction and its 95% confidence bounds (dashed and solid orange lines, respectively). Also in this case, the bounds are indistinguishable from the prediction. A close-up around a local maximum shows that the prediction compares well with the MC result, which is enclosed by the confidence bounds. Similar results are provided in Fig. 13 for the four crosstalk voltages, limited to a 6-bit window. The left y-axis refers to voltage values and their mean, whereas the right y-axis refers to their variance. Also in this case, very narrow bounds, encompassing the MC results, are found.

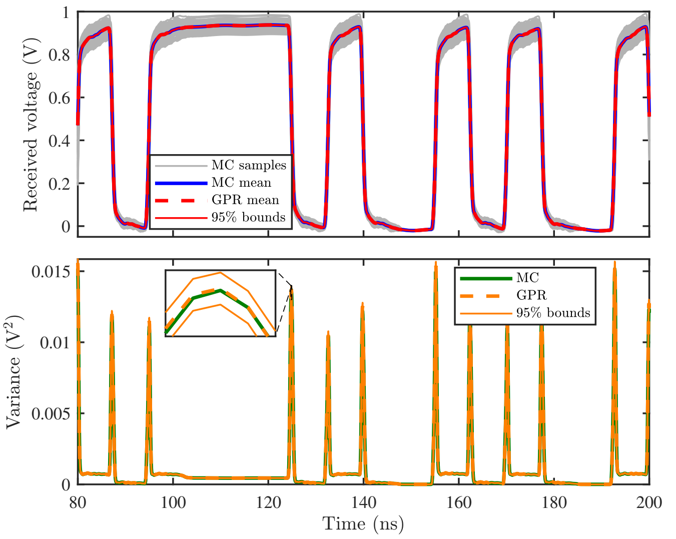


Fig. 12. Voltage at the receiver side of the electronic link (limited to a 16-bit window). In the top panel, the gray lines are a subset of MC samples, the blue line is the average of the MC samples, whereas the dashed and solid red lines are the GPR prediction of the average and its 95% confidence bounds, respectively. In the bottom panel, the green line is the variance of the MC samples, whereas the dashed and solid orange lines are the GPR prediction of the variance and its 95% confidence bounds, respectively.

Figure 14 provides a comparison on the PDFs computed for each of the five output voltages at a significant time point. The reference distribution of the MC samples is shown with a blue line. The solid red lines are the 95% confidence bounds of the prediction obtained with the GPR model. Furthermore, to assess the impact of the prior kernel, the dashed green lines show the confidence bounds obtained by using an anisotropic Matérn 5/2 kernel. It is found that the anisotropic kernel achieves a better prediction, with even tighter confidence bounds compared to the isotropic Gaussian kernel.

In order to provide a more quantitative comparison in this regard, two figures of merit are introduced. The first is the confidence width (CW), and measures the cumulative width of the confidence interval of the PDF:

$$CW = \int_{\mathbb{R}} \left(q_H(y) - q_L(y) \right) dy, \tag{40}$$

where q_H and q_L denote the upper and lower quantiles, respectively. A larger CW indicates a wider confidence interval, and hence a more uncertain prediction. The second figure of merit is the root-mean-square error (RSME) between the PDF ρ_{GPR} predicted by the GPR surrogate and the reference MC distribution ρ_{MC}:

$$RMSE = \sqrt{\int_{\mathbb{R}} \left(\rho_{\text{GPR}}(y) - \rho_{\text{MC}}(y) \right)^2 dy}. \tag{41}$$

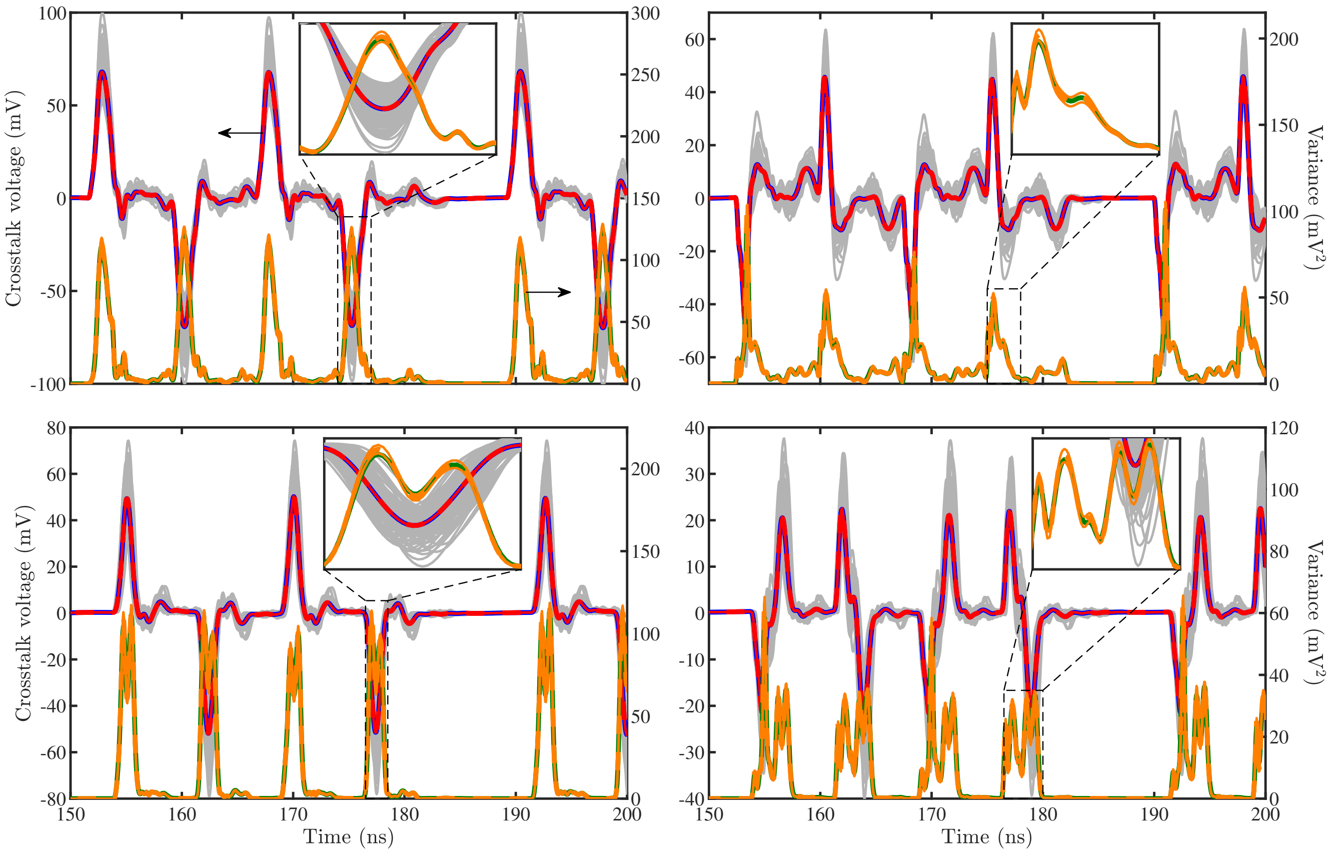


Fig. 13. Crosstalk voltages at the transmission-line sections of the electronic link (limited to a 6-bit window). The same color scheme as in Fig. 12 is used for the MC samples, as well as for the MC and GPR statistical estimates.

The results are collected in Table III. The figures show that the GPR model with anisotropic Matérn kernel is roughly twice more accurate in both measures, which is reasonable given its higher complexity in terms of degrees of freedom. However, it is important to mention that the training requires about 8 s for each of the 37 principal components, leading to a total, non-negligible training time of 311.4 s, compared to a mere 3.9 s required by the GPR model with isotropic kernel.

TABLE III
PERFORMANCE COMPARISON BETWEEN AN ISOTROPIC GAUSSIAN KERNEL AND AN ANISOTRIPIC MATÉRN KERNEL, IN TERMS OF ACCURACY AND TRAINING EFFICIENCY.

Kernel →	isotropic Gaussian		anisotropic Matérn 5/2	
Quantity ↓	CW	RMSE	CW	RMSE
PDF of crosstalk voltage #1 @ 160.4 ns	0.1530	0.1804	0.0848	0.1045
PDF of crosstalk voltage #2 @ 139.8 ns	0.1516	0.4482	0.1192	0.2292
PDF of crosstalk voltage #3 @ 162.0 ns	0.1413	0.2418	0.0730	0.1085
PDF of crosstalk voltage #4 @ 173.0 ns	0.1664	0.4679	0.1292	0.2251
PDF of received voltage #1 @ 130.2 ns	0.0842	0.0700	0.0417	0.0425
Training time	3.9 s		5 min 11 s	

VIII. Conclusions

This paper introduced a probabilistic framework, based on GPR, for the UQ of electronic circuits. As opposed to traditional surrogate modeling techniques, the advocated method additionally provides confidence bounds for the predicted statistical information. This is achieved by propagating the uncertainty from the input parameters to the output variables by also taking into account the inherent uncertainty of the GPR model. Furthermore, the use of PCA compression allows to effectively deal with problems characterized by a large number of outputs.

The proposed method was applied to both illustrative and real-life application examples, namely a low-noise amplifier and a digital communication link, each with a relatively large number of uncertain input parameters (21 and 26, respectively). It was shown that similar results are obtained compared to state-of-the-art techniques such as PCE and LS-SVM, which however do not provide information on the prediction confidence. Moreover, the use of an isotropic kernel provides a viable solution with good accuracy and limited training time, whereas anisotropic kernels allows to further improve the accuracy at the price of a non-negligible training cost. Plans for future work include the investigation of adaptive training strategies and of alternatives for the trend and kernel functions of the prior.

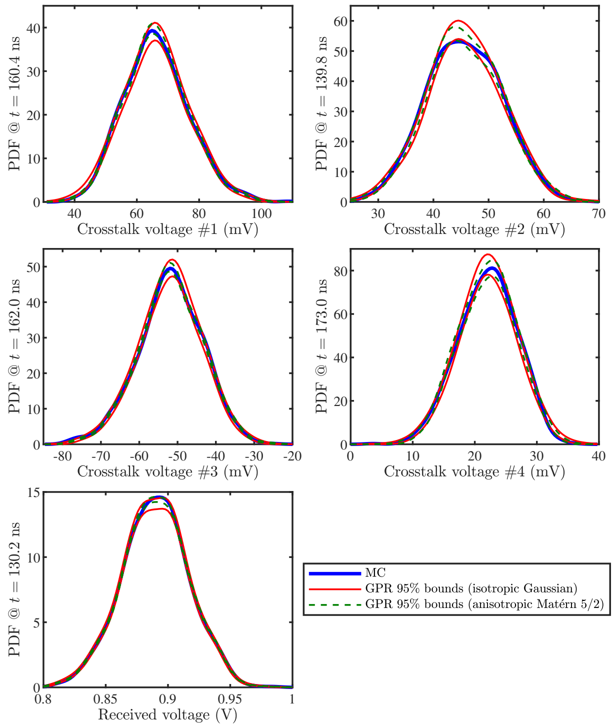


Fig. 14. PDFs of the five output voltages at different time points. The MC distribution (solid blue line) is compared with the 95% confidence bounds of the GPR predictions obtained with an isotropic squared-exponential kernel (solid red lines) and with an anisotropic Matérn 5/2 kernel (dashed green lines).

Appendix

Given a vector $\boldsymbol{\varepsilon}$ of N random variables with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, and an N-dimensional symmetric matrix $\boldsymbol{\Lambda}$, the scalar quantity

$$Q = \boldsymbol{\varepsilon}^{\mathsf{T}} \boldsymbol{\Lambda} \boldsymbol{\varepsilon} \tag{42}$$

is a "quadratic form" in $\boldsymbol{\varepsilon}$ [66]. The expected value of Q is

$$\mathrm{E}\{Q\} = \mathrm{tr}(\boldsymbol{\Lambda}\boldsymbol{\Sigma}) + \boldsymbol{\mu}^{\mathsf{T}} \boldsymbol{\Lambda} \boldsymbol{\mu}. \tag{43}$$

Moreover, if the random variables $\boldsymbol{\varepsilon}$ are normally distributed, the variance of Q is

$$\mathrm{Var}\{Q\} = 2\,\mathrm{tr}\left((\boldsymbol{\Lambda}\boldsymbol{\Sigma})^2\right) + 4\,\boldsymbol{\mu}^{\mathsf{T}} \boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{\Lambda}\boldsymbol{\mu}. \tag{44}$$

References

[1] S. Mukhopadhyay, "A generic data-driven nonparametric framework for variability analysis of integrated circuits in nanometer technologies," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 28, no. 7, pp. 1038–1046, Jul. 2009.

[2] J. Yao, Z. Ye, and Y. Wang, "Importance boundary sampling for SRAM yield analysis with multiple failure regions," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 33, no. 3, pp. 384–396, Mar. 2014.

[3] A. Lange, C. Sohrmann, R. Jancke, J. Haase, B. Cheng, A. Asenov, and U. Schlichtmann, "Multivariate modeling of variability supporting non-Gaussian and correlated parameters," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 35, no. 2, pp. 197–210, Feb. 2016.

[4] I. Ukhov, P. Eles, and Z. Peng, "Probabilistic analysis of power and temperature under process variation for electronic system design," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 33, no. 6, pp. 931–944, Jun. 2014.

[5] I. Ukhov, P. Eles, and Z. Peng, "Probabilistic analysis of electronic systems via adaptive hierarchical interpolation," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 36, no. 11, pp. 1883–1896, Nov. 2017.

[6] O. Lahiouel, M. H. Zaki, and S. Tahar, "Accelerated and reliable analog circuits yield analysis using SMT solving techniques," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 37, no. 3, pp. 517–530, Mar. 2018.

[7] S. Mohanachandran Nair, R. Bishnoi, M. S. Golanbari, F. Oboril, F. Hameed, and M. B. Tahoori, "VAET-STT: Variation aware STT-MRAM analysis and design space exploration tool," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 37, no. 7, pp. 1396–1407, Jul. 2018.

[8] H. Amrouch, G. Pahwa, A. D. Gaidhane, C. K. Dabhi, F. Klemme, O. Prakash, and Y. Singh Chauhan, "Impact of variability on processor performance in negative capacitance FinFET technology," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 9, pp. 3127–3137, Sep. 2020.

[9] Z. Gao, J. Tao, Y. Su, D. Zhou, X. Zeng, and X. Li, "Efficient rare failure analysis over multiple corners via correlated Bayesian inference," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 39, no. 10, pp. 2029–2041, Oct. 2020.

[10] R. Spence and R. S. Soin, *Tolerance Design of Electronic Circuits*. London, U.K.: Imperial College Press, 1997.

[11] A. Singhee and R. A. Rutenbar, "Why quasi-Monte Carlo is better than Monte Carlo or Latin hypercube sampling for statistical circuit analysis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 29, no. 11, pp. 1763–1776, Nov. 2010.

[12] D. Xiu and G. E. Karniadakis, "The Wiener-Askey polynomial chaos for stochastic differential equations," *SIAM J. Sci. Computation*, vol. 24, no. 2, pp. 619—644, 2002.

[13] 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. 30:1–21, Feb. 2018.

[14] S. Vrudhula, J. M. Wang, and P. Ghanta, "Hermite polynomial based interconnect analysis in the presence of process variations," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 25, no. 10, pp. 2001–2011, Oct. 2006.

[15] N. Mi, S.X.-D. Tan, Y. Cai, and X. Hong, "Fast variational analysis of on-chip power grids by stochastic extended Krylov subspace method," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 27, no. 11, pp. 1996–2006, Nov. 2008.

[16] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel, and L. Daniel, "Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 32, no. 10, pp. 1533–1545, Oct. 2013.

[17] P. Manfredi, R. Trinchero, and D. Vande Ginste, "A perturbative stochastic Galerkin method for the uncertainty quantification of linear circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 9, pp. 2993–3006, Sep. 2020.

[18] 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.

[19] 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.

[20] P. Manfredi, D. Vande Ginste, D. De Zutter, and F.G. Canavero, "Generalized decoupled polynomial chaos for nonlinear circuits with many random parameters," *IEEE Microw. Wireless Compon. Lett.*, vol. 25, no. 8, pp. 505–507, Aug. 2015.

[21] M. Ahadi and S. Roy, "Sparse linear regression (SPLINER) approach for efficient multidimensional uncertainty quantification of high-speed circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 35, no. 10, pp. 1640–1652, Oct. 2016.

[22] Z. Zhang, X. Yang, I. V. Oseledets, G. E. Karniadakis, and L. Daniel, "Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 34, no. 1, pp. 63–76, Jan. 2015.

[23] Z. Zhang, K. Batselier, H. Liu, L. Daniel, and N. Wong, "Tensor computation: a new framework for high-dimensional problems in EDA," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 36, no. 4, pp. 521–536, Apr. 2017.

[24] Z. Zhang, T. 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.

[25] C. Cui and Z. Zhang, "High-dimensional uncertainty quantification of electronic and photonic IC with non-Gaussian correlated process variations," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 39, no. 8, pp. 1649–1661, Aug. 2020.

[26] S. Haykin, *Neural Networks and Learning Machines*, Third Upper Saddle River, NJ, USA: Pearson Education, 2009.

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

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

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

[30] C. Lataniotis, D. Wicaksono, S. Marelli, and B. Sudret, *UQLab User Manual: Kriging (Gaussian Process Modeling)*, UQLab-V1.4-105, 2021.

[31] S. Ulaganathan, I. Couckuyt, T. Dhaene, J. Degroote, and E. Laermans, "High dimensional Kriging metamodelling utilising gradient information," *Appl. Math. Modelling*, vol. 40, no. 9–10, pp. 5256–5270, May 2016.

[32] M. A. Bouhlel, N. Bartoli, A. Otsmane, and J. Morlier, "Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction," *Struct. Multidisc. Optim.*, vol. 53, pp. 935–952, May 2016.

[33] M. A. Bouhlel and J. R. R. A. Martins, "Gradient-enhanced kriging for high-dimensional problems," *Eng. Comput.*, vol. 35, pp. 157–173, Jan. 2019.

[34] L. Chen, H. Qiu,L. Gao, C. Jiang, and Z. Yang, "A screening-based gradient-enhanced Kriging modeling method for high-dimensional problems," *Appl. Math. Modelling*, vol. 69, pp. 15–31, May 2019.

[35] K. Lee, H. Cho, and I. Lee, "Variable selection using Gaussian process regression-based metrics for high-dimensional model approximation with limited data," *Struct. Multidisc. Optim.*, vol. 59, pp. 1439–1454, May 2019.

[36] Y. Zhou and Z. Lu, "An enhanced Kriging surrogate modeling technique for high-dimensional problems," *Mech. Syst. Signal Process.*, vol. 140, pp. 106687, Jun. 2020.

[37] H. Yu, H. Chalamalasetty, and M. Swaminathan, "Modeling of voltage-controlled oscillators including I/O behavior using augmented neural networks," *IEEE Access*, vol. 7, pp. 38973–38982, 2019.

[38] F. E. Rangel-Patiño, J. E. Rayas-Sánchez, A. Viveros-Wacher, J. L. Chávez-Hurtado, E. A. Vega-Ochoa, and N. Hakim, "Post-silicon receiver equalization metamodeling by artificial neural networks," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 38, no. 4, pp. 733–740, Apr. 2019.

[39] B. Li, B. Jiao, C. Chou, R. Mayder and P. Franzon, "Self-evolution cascade deep learning model for high-speed receiver adaptation," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 10, no. 6, pp. 1043–1053, Jun. 2020.

[40] H. Ma, E. 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.

[41] H. Lin and P. Li, "Circuit performance classification with active learning guided sampling for support vector machines," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 34, no. 9, pp. 1467–1480, Sep. 2015.

[42] J. E. Rayas-Sánchez and V. Gutierrez-Ayala, "EM-based Monte Carlo analysis and yield prediction of microwave circuits using linear-input neural-output space mapping," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 12, pp. 4528–4537, Dec. 2006.

[43] S. J. Park, B. Bae, J. Kim, and M. Swaminathan, "Application of machine learning for optimization of 3-D integrated circuits and systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 6, pp. 1856–1865, Jun. 2017.

[44] H. M. Torun, M. Swaminathan, A. Kavungal Davis, and M. L. F. Bellaredj, "A global Bayesian optimization algorithm and its application to integrated system design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 4, pp. 792–802, Apr. 2018.

[45] M. Wang, W. Lv, F. Yang, C. Yan, W. Cai, D. Zhou, and X. Zeng, "Efficient yield optimization for analog and SRAM circuits via Gaussian process regression and adaptive yield estimation," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 37, no. 10, pp. 1929–1942, Oct. 2018.

[46] V. Dubourg, B. Sudret, and J. M. Bourinet, "Reliability-based design optimization using kriging surrogates and subset simulation," *Struct. Multidisc. Optim.*, vol. 44, pp. 673–690, Nov. 2011.

[47] M. Moustapha, B. Sudret, J. M. Bourinet, and B. Guillaume, "Quantile-based optimization under uncertainties using adaptive Kriging surrogate models," *Struct. Multidisc. Optim.*, vol. 54, pp. 1403–1421, Dec. 2016.

[48] 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.

[49] 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.

[50] 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.*, vol. 62, no. 5, pp. 1755–1762, Oct. 2020.

[51] A. Keprate, R. M. C. Ratnayake, and S. Sankararaman, "Adaptive Gaussian process regression as an alternative to FEM for prediction of stress intensity factor to assess fatigue degradation in offshore pipeline," *Int. J. Pres. Ves. Pip.*, vol. 153, pp. 45–58, Jun. 2017.

[52] M. Yang and Z. Xiao, "POD-based surrogate modeling of transitional flows using an adaptive sampling in Gaussian process," *Int. J. Heat Fluid Flow*, vol. 84, pp. 108596, Aug. 2020.

[53] J. N. Fuhg, A. Fau, and U. Nackenhorst, "State-of-the-art and comparative review of adaptive sampling methods for kriging," *Arch. Computat. Methods Eng.*, vol. 28, pp. 2689–2747, Jun. 2021.

[54] 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 Tech.*, vol. 67, no. 10, pp. 4140–4155, Oct. 2019.

[55] 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.

[56] H. Hagmar, L. Tong, R. Eriksson, and L. A. Tuan, "Voltage instability prediction using a deep recurrent neural network," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 17–27, Jan. 2021.

[57] I. Bilionis and N. Zabaras, "Multi-output local Gaussian process regression: Applications to uncertainty quantification," *J. Comput. Physics*, vol. 231, no. 17, pp. 5718–5746, Jul. 2012.

[58] 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.

[59] I. Bilionis, N. Zabaras, B. A. Konomi, and G. Lin, "Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification," *J. Comput. Physics*, vol. 241, pp. 212–239, May 2013.

[60] J. D'Errico, "nearestSPD", *MATLAB Central File Exchange*. https://www.mathworks.com/matlabcentral/fileexchange/42885-nearestspd (Retrieved October 19, 2020).

[61] R. Schöbi, B. Sudret, and J. Wiart, "Polynomial-chaos-based Kriging," *Int. J. Uncertainty Quantification*, vol. 5, no. 2, pp. 171–193, 2015.

[62] F. Lindsten, T. B. Schön, A. Svensson, and N. Wahlström, *Probabilistic modeling – linear regression & Gaussian processes*, Uppsale: Uppsala Univ., 2017 [Online]. Available: http://www.it.uu.se/edu/course/homepage/sml/literature/probabilistic_modeling_compendium.pdf

[63] *Statistics and Machine Learning Toolbox*, version 11.7, The MathWorks, Inc., Natick, MA, USA.

[64] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991.

[65] M. Loève, *Probability Theory*, 4th ed. New York, NY, USA: Springer-Verlag, 1977.

[66] A. C. Rencher and G. B. Schaalje, *Linear Models in Statistics*, 2nd ed. Hoboken, NJ, USA: Wiley, 2008.

[67] 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.

[68] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, Jun. 1975.

[69] T. Buss, "2 GHz low noise amplifier with the BFG425W," Philips Semiconductors B.V., Nijmegen, The Netherlands, Appl. Note RNRT45-96-B-773, Nov. 1996.

[70] I. M. Sobol, "The distribution of points in a cube and the approximate evaluation of integrals," *Zh. Vychisl. Mat. i Mat. Fiz.*, vol. 7, 1967. [in Russian]

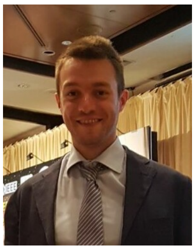
[71] L. Devroye, *Non-Uniform Random Variate Generation*. New York, NY, USA: Springer-Verlag, 1986.

[72] LS-SVMlab, version 1.8; Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven: Leuven, Belgium, 2011. Available online: http://www.esat.kuleuven.be/sista/lssvmlab/

[73] S. Marelli and B. Sudret, "UQLab: A framework for uncertainty quantification in Matlab," in *Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management*, Liverpool, United Kingdom, Apr. 2014, pp. 2554–2563.

[74] J. Matías, A. Vaamonde, J. Taboada, and W. González-Manteiga, "Support vector machines and gradient boosting for graphical estimation of a slate deposit," *Stoch. Envir. Res. Risk Ass.*, vol. 18, pp. 309–323, Oct. 2004.

[75] 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.



Paolo Manfredi (S'10–M'14–SM'18) received the M.Sc. degree in electronic engineering from the Politecnico di Torino, Torino, 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 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 Associate 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 uncertainty quantification, surrogate modeling, signal integrity, and electromagnetic compatibility.

Dr. Manfredi was a recipient of an Outstanding Young Scientist Award at the 2018 Joint IEEE International Symposium on Electromagnetic Compatibility & 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, a 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. He is currently serving as an Associate Editor for the IEEE JOURNAL ON MULTISCALE AND MULTIPHYSICS COMPUTATIONAL TECHNIQUES and the *International Journal of Circuit Theory and Applications*, and as an Academic Editor for *Mathematical Problems in Engineering*.

Riccardo Trinchero (M'16) received the M.Sc. and the Ph.D. degrees in electronics and communication engineering from Politecnico di Torino, Torino, Italy, in 2011 and 2015, respectively.

He is currently an Assistant Professor 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.