POLITECNICO DI TORINO
Repository ISTITUZIONALE

Nonparametric formulation of polynomial chaos expansion based on least-square support-vector machines

Original
Nonparametric formulation of polynomial chaos expansion based on least-square support-vector machines / Manfredi, Paolo; Trinchero, Riccardo. - In: ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE. - ISSN 0952-1976. - STAMPA. - 133:Part C(2024), pp. 1-20. [10.1016/j.engappai.2024.108182]

Availability:
This version is available at: 11583/2987735 since: 2024-04-11T13:07:19Z

Publisher:
Elsevier

Published
DOI:10.1016/j.engappai.2024.108182

Terms of use:

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

Publisher copyright

(Article begins on next page)

30 April 2024

Engineering Applications of Artificial Intelligence 133 (2024) 108182

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Research paper

Nonparametric formulation of polynomial chaos expansion based on least-square support-vector machines

Paolo Manfredi *, Riccardo Trinchero

Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi 24, Turin 10129, Italy

ARTICLE INFO

Keywords:
Kernel
Least-square support-vector machines
Machine learning
Polynomial chaos
Uncertainty quantification

ABSTRACT

This paper introduces an innovative data-driven approach to uncertainty quantification (UQ) in complex engineering designs based on polynomial chaos expansion (PCE) and least-square support-vector machine (LSSVM). While PCE is a prevalent UQ method, its reliance on a predefined model form poses limitations in the model complexity, training accuracy, and efficiency in high-dimensional settings. To overcome this, we propose a nonparametric reformulation that draws equivalence with the kernel-based LSSVM. Leveraging special implicit kernels based on Hermite or Legendre polynomials, our method achieves accurate predictions with reduced training data. The model is trained through the dual space LSSVM formulation, thereby exploiting the so-called "kernel trick", while PCE coefficients up to an arbitrary order are efficiently obtained post-training. Validation encompasses a diverse range of scenarios, including standard benchmarks and real-life applications in electrical engineering involving up to 26 independent random input parameters. The proposed "PCE-LSSVM" method exhibits superior performance over traditional PCE implementations. It achieves up to 7 times smaller root-mean-square error and up to 10 times lower dispersion across various training datasets, showcasing superior performance even with a significantly reduced number of training samples. Notably, this approach is 2 to 50 times more efficient compared to alternative approaches such as ordinary least square regression or Gaussian quadratures. Overall, PCE-LSSVM emerges as a powerful and efficient tool for accurate UQ in complex engineering systems.

1. Introduction

Uncertainty quantification (UQ) is becoming of paramount importance in many domains of science to assess how complex systems react to random perturbations of their parameters, thus supporting design optimization and decision making. A (definitely non-exhaustive) list of applications includes the modeling of weather predictions (Moosavi et al., 2021), water quality (Freni and Mannina, 2010), water usage (Pang and O'Neill, 2018), energy production systems (Kim et al., 2019), the energy performance of buildings (Srivastav et al., 2013), as well as the design and safety analysis of nuclear reactors (Avramova and Ivanov, 2010) and structural health monitoring (Lorenzoni et al., 2016). Moreover, in modern mass production electronics, manufacturing tolerances and process variations play a key limiting factor to evergrowing miniaturization and frequency scaling and must suitably taken into account during the design phase (Dietrich and Haase, 2011; Weng et al., 2015).

In this context, polynomial chaos expansion (PCE) (Xiu and Karniadakis, 2002) has been widely adopted as a robust and well-consolidated tool for UQ and still represents a very active field of research (Liu and Jiang, 2023; Kantarakias and Papadakis, 2023; Bürkner et al., 2023; Zhang and Ni, 2023; García-Merino et al., 2023; Vauchel et al., 2023; Lee and Rahman, 2023; Yang et al., 2023). The PCE leverages special classes of orthogonal polynomials to approximate the desired quantity of interest (QoI). The popularity of the PCE stems from the fact that statistical information, like moments and sensitivity indices, is analytically derived from the model coefficients (Sudret, 2008). Indeed, as opposed to other surrogate modeling tools, the method is specifically designed to deal with uncertain inputs. The PCE method has been extensively demonstrated to outperform brute force Monte Carlo (MC) methods in terms of trade-off between accuracy and efficiency, especially for relatively smooth problems that depend on a moderate number of uncertain parameters.

Several approaches exist for the calculation of the PCE coefficients. Most of them are general, as they work on the set of governing equations, and therefore they are readily applied in any domain (Xiu, 2009). The stochastic Galerkin method applies projections to the original set of stochastic equations to arrive at an augmented but deterministic

* Correspondence to: Department of Electronics and Telecommunications, Politecnico di Torino, Turin 10129, Italy.
E-mail address: paolo.manfredi@polito.it (P. Manfredi).

https://doi.org/10.1016/j.engappai.2024.108182
Received 1 August 2023; Received in revised form 21 February 2024; Accepted 26 February 2024
0952-1976/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

set of equations in the unknown coefficients (Ghanem and Spanos, 1991). This method is considered to be the most accurate. However, it is intrusive, in that it requires to access and modify the governing equations, and hence it is hardly applicable in conjunction with existing numerical tools, especially proprietary ones, and/or to nonlinear problems. Pseudo-spectral collocation methods approximate the projection integrals by means of suitable quadrature rules, e.g., Gauss or sparse-grid schemes (Xiu, 2007; Xiu and Hesthaven, 2005). In this regard, they only need to collect samples of the QoI evaluated for specific configurations of the input parameters. Their main limitation is that the number of quadrature nodes increases dramatically with the expansion order and the number of input dimensions, a problem that is mitigated only in part by using sparse strategies. Finally, regression-based approaches fit the model coefficients via a suitable regression scheme (Hadigol and Doostan, 2018). While ordinary least square (OLS) strategies are hardly applicable because of the large number of samples they require to overdetermine the system, sparse approaches are able to work with a limited amount of data and identify a subset of most important coefficients. A popular example is the least-angle regression (LAR) (Blatman and Sudret, 2011), which rapidly became a popular state-of-the-art technique due to its general applicability and capability to work with limited data. In general, one of the main limitations of the PCE is that it is a parametric method, meaning that the form and the size of the model are determined upfront by the predefined (maximum) expansion order and truncation scheme, even for sparse methods. The achievable accuracy is also limited upfront by the selected expansion order.

In recent years, the burgeoning of machine learning methods has paved the way to their application to UQ tasks (Psaros et al., 2023). For instance, neural networks (Shin and Choi, 2023; Zhu et al., 2023; Qi and Harlim, 2023; Zhang and Shafieezadeh, 2023; Garg and Chakraborty, 2023; Antil et al., 2023) as well as kernel-based methods like support vector machines (He et al., 2019; Trinchero et al., 2018) and Gaussian process regression (Bilionis and Zabaras, 2012; Bhattacharyya, 2022; Chahar and Mukhopadhyay, 2023; Chang and Zeng, 2023; Peng et al., 2023; Li et al., 2023) have been investigated for UQ problems. Specifically, kernel methods are nonparametric as they do not exhibit a predefined model form. Their complexity scales well with the input dimensionality, while in fact being mainly determined by the amount of available training data. In most cases, the learning capability is virtually arbitrary, provided that a sufficient amount of training data is provided. In particular, the least-square formulation of support-vector machines (Suykens et al., 2002) exhibits superior generalization performance compared to competing methods, and the performance is not significantly affected by the dimensionality of the input data (dos Santos et al., 2012). The least-square support-vector machine (LSSVM) regression has been applied, e.g., to the identification of thermal processes (dos Santos et al., 2012), to the prediction of energy consumption (Kaytez et al., 2015), and to the modeling of the activity of HIV-1 protease inhibitors (Cui and Yan, 2009). As opposed to the PCE, machine learning methods are general purpose techniques that can reproduce the input–output relation of a complex simulation environment, thus lacking interpretability in the UQ scenario. Indeed, they are used as mere surrogates of the expensive simulator to predict a large number of samples of the QoI, from which statistical information is numerically extracted in a MC-like analysis.

This paper bridges the gap between the PCE and kernel-based machine learning methods by providing a link between the former and the LSSVM regression. In particular, it is shown that the PCE and the *primal space* LSSVM formulation are in fact equivalent, provided that the classical set of orthogonal polynomials is used as basis functions in the feature space. Furthermore, two special kernels are introduced that reproduce the inner product of an infinite sequence of Hermite or Legendre polynomials. Thanks to these special kernels, the model training is performed in the equivalent dual space formulation by exploiting the so-called "kernel trick", while the PCE coefficients are retrieved via back-projection onto the primal feature space. This is possible because the feature space functions are known, a situation that rarely holds with standard kernels, and makes the training of the PCE model virtually transparent to the input dimensionality, expansion order, and number of basis functions. Indeed, the PCE coefficients are analytically retrieved in post-processing for each individual basis function and up to an *arbitrary* order, without the need to compute the complete basis at once. The novel contribution put forward in this paper is twofold: on the one hand, we present an elegant theoretical framework that links the PCE and LSSVM methods, in which the former is cast as a primal space formulation of the latter, while the training is performed in the dual space. On the other hand, we show that the advocated approach achieves better accuracy compared to the state-of-the-art LAR method by using a substantially smaller number of training samples. Specifically, we assess the performance as follows: we quantify the accuracy based on both the root-mean-square error (RMSE) and coefficient of determination; we evaluate the reproducibility based on the dispersion of the above-mentioned metrics across different training datasets; we determine the efficiency based on the number of training data and computational resources required.

The remainder of this paper is organized as follows. Sections 2 and 3 provide a brief overview of the PCE and LSSVM methods, as needed for the subsequent developments. Section 4 first introduces a link between the two methods based on an explicit definition of the kernel, truncated to a finite expansion order. This is an intermediate step that is useful to understand the final result. Then, in Section 5, an implicit definition of the kernel is introduced for the case of Hermite and Legendre expansions. The method is illustrated based on a simple one-dimensional function in Section 6. It is then applied to challenging application test cases belonging to the electrical engineering domain in Section 7 and to classical benchmark functions in Section 8. The outcomes of these analyses are summarized and discussed in Section 9. Finally, conclusions are drawn in Section 10.

2. Polynomial chaos expansion

This section introduces the PCE, which is the first ingredient of the proposed methodology. In particular, we review the basic properties, the orthogonal basis functions, and the state-of-the-art solutions for the calculation of the expansion coefficients, as needed for the subsequent developments and to properly compare the proposed technique with the available approaches.

Consider a set of d uncertain parameters $\boldsymbol{x} = (x_1, \dots, x_d)$. We further assume the uncertain parameters to be independent and, for the ease of notation, to have the same distribution. The PCE seeks to approximate a target function $y = \mathcal{M}(\boldsymbol{x})$, mapping a configuration $\boldsymbol{x}$ of the input parameters to the corresponding output y, with a polynomial model in the form of Xiu and Karniadakis (2002), Xiu (2009)

$$y \approx \hat{y} = \mathcal{M}_{\text{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\kappa} \in \mathcal{K}} c_{\boldsymbol{\kappa}} \psi_{\boldsymbol{\kappa}}(\boldsymbol{x}), \tag{1}$$

where $\psi_{\boldsymbol{\kappa}}$ are multivariate polynomials. The basis functions are constructed as the product of univariate orthonormal polynomials, i.e.,

$$\psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) = \prod_{j=1}^{d} \zeta_{\kappa_j}(x_j) \tag{2}$$

according to the multi-index $\boldsymbol{\kappa} = (\kappa_1, \dots, \kappa_d) \in \mathbb{N}^d$. Since $\deg(\zeta_k) = k$, the multi-index $\boldsymbol{\kappa}$ defines the degree of the basis function in each dimension.

2.1. Basic properties

The univariate polynomials satisfy the orthogonality condition

$$\langle \zeta_k, \zeta_m \rangle = \int_{\mathbb{R}} \zeta_k(x) \zeta_m(x) w(x) dx = \delta_{km}, \tag{3}$$

where $w(x)$ is the probability density function of the each individual uncertain parameter x. The orthogonality readily extends to the multivariate bases because of their construction via (2).

One of the key aspects of the PCE is the definition of the set of multi-indices $\mathcal{K} \subset \mathbb{N}^d$, which in turns defines the truncation of the expansion. The most common choices are based on bounding some norm of the multi-indices by a maximum degree p, e.g., $\|\boldsymbol{\kappa}\|_\infty \leq p$ (tensor product truncation), $\|\boldsymbol{\kappa}\|_1 \leq p$ (total degree truncation), or $\|\boldsymbol{\kappa}\|_u \leq p$, with $0 < u < 1$ (hyperbolic truncation), leading to increasingly sparser expansions (Blatman and Sudret, 2011).

The PCE (1) can be rewritten with a scalar indexing as

$$\mathcal{M}_{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{k=0}^{K} c_k \psi_k(\boldsymbol{x}), \tag{4}$$

where the scalar index k maps to the elements of $\mathcal{K}$, which are assumed to be ordered according to a given rule (typically, graded lexicographic ordering). Hence, $K = |\mathcal{K}| - 1$. Note that a zero-based indexing is preferred, since the first term is always of order zero. Throughout this paper, we will use either notation as convenient.

One of the peculiar properties of PCEs is that, for any L^2 function, the approximation error tends to zero if the expansion order p is increased (Xiu, 2009), i.e.,

$$\|y - \hat{y}\|_{L^2}^2 = \lim_{K \to \infty} \int_{\mathbb{R}^d} \left(\mathcal{M}(\boldsymbol{x}) - \mathcal{M}_{\mathrm{PCE}}(\boldsymbol{x})\right)^2 w(\boldsymbol{x}) d\boldsymbol{x} = 0 \tag{5}$$

which is equivalent to letting $p \to \infty$. Since the norm is defined based on the inner product (3), a function is L^2 if it has finite variance. Moreover, thanks to the orthogonality of the polynomials, statistical moments and sensitivity information are obtained analytically from the PCE coefficients. For example, the mean and the variance of y are obtained as

$$\mathrm{E}\{y\} \approx \mathrm{E}\{\hat{y}\} = c_0 \tag{6}$$

and

$$\mathrm{Var}\{y\} \approx \mathrm{Var}\{\hat{y}\} = \sum_{k=1}^{K} c_k^2, \tag{7}$$

respectively. Sobol's sensitivity indices, describing the contribution of individual inputs to the output variance, are obtained by summing the squares of appropriate subsets of the coefficients (Sudret, 2008). Because of the aforementioned statistical properties, the PCE has been widely adopted for uncertainty quantification problems over the recent years (Najm, 2009; Kaintura et al., 2018; Lüthen et al., 2021).

2.2. Orthogonal polynomials

According to the Wiener–Askey scheme (Xiu and Karniadakis, 2002), there exist well-known bases of orthogonal polynomials that satisfy (3) for standard probability distributions. For example, Hermite, Legendre, and Jacobi polynomials are orthogonal for the Gaussian, uniform, and beta distributions, respectively. In general, it is possible to numerically compute orthogonal polynomials for arbitrary and possibly correlated distributions (Cui and Zhang, 2018a,b). However, for the most general case of a correlated and non-Gaussian distribution, the multivariate polynomials are no longer separable as in (2). In the following sections, we focus the attention on Hermite and Legendre polynomials, which are the ones associated to the popular Gaussian and uniform distributions.

2.2.1. Hermite polynomials

We consider a Gaussian distributed random parameter x. We further assume that x is normalized so that $x \sim \mathcal{N}(0,1)$. Note that this can be always achieved by proper translation and scaling. Hence, the probability distribution of x is

$$w(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \tag{8}$$

The *probabilist's* Hermite polynomials are orthogonal based on the above measure, and they are generated by the three-term recurrence relation (Gradshteyn and Ryzhik, 2014)

$$He_{k+1}(x) = xHe_k(x) - kHe_{k-1}(x), \tag{9}$$

for $k \geq 0$, with $He_0(x) = 1$ and $He_{-1}(x) = 0$. The first five Hermite polynomials are therefore

$$He_0(x) = 1$$
$$He_1(x) = x$$
$$He_2(x) = x^2 - 1$$
$$He_3(x) = x^3 - 3x$$
$$He_4(x) = x^4 - 6x^2 + 3.$$

Moreover, since

$$\langle He_k, He_k \rangle = \|He_k\|_{L^2}^2 = k!, \tag{10}$$

the *orthonormal* Hermite polynomials, which we shall use for the PCE (1), are defined as

$$\zeta_k(x) = \frac{He_k(x)}{\sqrt{k!}}, \tag{11}$$

the first five being

$$\zeta_0(x) = 1$$
$$\zeta_1(x) = x$$
$$\zeta_2(x) = (x^2 - 1)/\sqrt{2}$$
$$\zeta_3(x) = (x^3 - 3x)/\sqrt{6}$$
$$\zeta_4(x) = (x^4 - 6x^2 + 3)/\sqrt{24}.$$

2.2.2. Legendre polynomials

We now consider a uniformly distributed random parameter x. We further assume that x is normalized so that $x \sim \mathcal{U}(-1,1)$, which can be always achieved by proper translation and scaling. Hence, the probability distribution of x is

$$w(x) = \begin{cases} \frac{1}{2} & |x| < 1 \\ 0 & |x| > 1 \end{cases}. \tag{12}$$

The orthogonal polynomials w.r.t. the above measure are the Legendre polynomials, which are generated by the three-term recurrence relation (Doman, 2015)

$$P_{k+1}(x) = \frac{2k+1}{k+1} x P_k(x) - \frac{k}{k+1} P_{k-1}(x), \tag{13}$$

for $k \geq 0$, with $P_0(x) = 1$ and $P_{-1}(x) = 0$. The first five Legendre polynomials are therefore

$$P_0(x) = 1$$
$$P_1(x) = x$$
$$P_2(x) = \frac{3}{2}x^2 - \frac{1}{2}$$
$$P_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x$$
$$P_4(x) = \frac{35}{8}x^4 - \frac{30}{8}x^2 + \frac{3}{8}.$$

Since

$$\langle P_k, P_k \rangle = \|P_k\|_{L^2}^2 = \frac{1}{2k+1}, \tag{14}$$

the orthonormal Legendre polynomials are defined as

$$\zeta_k(x) = \sqrt{2k+1}\, P_k(x), \tag{15}$$

the first five being

$$\zeta_0(x) = 1$$
$$\zeta_1(x) = \sqrt{3}\, x$$

$$\zeta_2(x) = \sqrt{5}\left(\frac{3}{2}x^2 - \frac{1}{2}\right)$$
$$\zeta_3(x) = \sqrt{7}\left(\frac{5}{2}x^3 - \frac{3}{2}x\right)$$
$$\zeta_4(x) = \sqrt{9}\left(\frac{35}{8}x^4 - \frac{30}{8}x^2 + \frac{3}{8}\right).$$

2.3. Calculation of the PCE coefficients

Owing to the orthogonality of the basis polynomials, the PCE coefficients are rigorously computed as

$$c_k = \langle y, \zeta_k \rangle = \int_{\mathbb{R}^d} \mathcal{M}(\boldsymbol{x})\zeta_k(\boldsymbol{x})w(\boldsymbol{x})d\boldsymbol{x}. \tag{16}$$

In practice, the above integral is difficult to evaluate in high-dimensional settings and/or when the target function $\mathcal{M}$ is expensive to evaluate. Hence, various numerical techniques were devised for the calculation of the coefficients (Xiu, 2009; Marelli et al., 2022).

2.3.1. Quadrature-based projection methods

Pseudo-spectral collocation methods approximate the integral (16) by means of Gauss quadratures, possibly combined with sparse (e.g., Smolyak') grids (Xiu, 2007; Xiu and Hesthaven, 2005). In either case, the PCE coefficients are calculated as (Marelli et al., 2022)

$$c_k \approx \sum_{q=1}^{Q} y_q \psi_k(\boldsymbol{x}_q) w_q, \tag{17}$$

which is an approximation of (16) based on a given quadrature rule with Q nodes $\boldsymbol{x}_q$ and corresponding weights w_q, whereas the $y_q = \mathcal{M}(\boldsymbol{x}_q)$ are the output observations computed at the quadrature nodes. Albeit usually very accurate, this approach bears two limitations:

1. The number of quadrature nodes is typically (much) larger than the number of PCE coefficients, i.e., $Q \gg K+1$, even when sparse grids are used. This is often not affordable in complex designs that require time-consuming simulations to compute the required output observations.
2. It requires to evaluate the output observations at specified points, which is not always possible, e.g., when data are available from previous experiments that cannot be easily reperformed. Therefore, this approach is not fully "data-driven".

As an example, a third-order ($p = 3$) and ten-dimensional ($d = 10$) PCE expansion with total degree truncation features $K + 1 = 286$ terms and requires $Q = 4^{10} \approx 10^6$ samples for the projection with a full Gauss quadrature rule, and a still rather large number of $Q = 1771$ samples with a Smolyak' grid.

2.3.2. Least-square regression method

Regression-based approaches use samples of the original function to build an overdetermined problem that is then solved in the least-square sense (Hadigol and Doostan, 2018), i.e.,

$$\boldsymbol{c} = (c_0, \dots, c_K)^{\mathsf{T}} = \arg\min \mathrm{E}\left\{\left(\sum_{k=0}^{K} c_k \psi_k(\boldsymbol{x}) - \mathcal{M}(\boldsymbol{x})\right)^2\right\}, \tag{18}$$

where $\boldsymbol{c}$ is the vector of PCE coefficients. This leads to the well-known OLS solution

$$\boldsymbol{c} = \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{y}, \tag{19}$$

where $\boldsymbol{A}$ is a matrix with entries $A_{lk} = \psi_k(\boldsymbol{x}_l)$ and $\boldsymbol{y} = (y_1, \dots, y_L)$ is the vector of observations, with $y_l = \mathcal{M}(\boldsymbol{x}_l)$ for $l = 1, \dots, L$. As opposed to quadrature rules, the sampling points are not predetermined, and they are typically drawn according to the input distribution. In this case, the requirement is $L > K$ in order for the regression problem to be overdetermined. Hence, this method is more parsimonious in terms of data. Nonetheless, the calculation is less rigorous and prone to larger errors related to the specific set of training data.

2.3.3. Sparse regression methods

Sparse regression methods, such as least-angle regression (LAR) (Blatman and Sudret, 2011) or orthogonal matching pursuit (Baptista et al., 2019), are able to identify a sparse subset of basis functions and calculate the corresponding coefficients by relaxing the requirement of the system to be overdetermined w.r.t. the original number of unknowns. Compared to (18), LAR uses a sparse subset of expansion terms $\mathcal{A} \subseteq \mathcal{K}$ and adds a penalty on the 1-norm of the coefficient vector, i.e., it solves

$$\boldsymbol{c} = \underset{\boldsymbol{c}\in\mathbb{R}^{|\mathcal{A}|}}{\arg\min}\, \mathrm{E}\left\{\left(\sum_{k=0}^{K} c_k \psi_k(\boldsymbol{x}) - \mathcal{M}(\boldsymbol{x})\right)^2\right\} + \lambda\|\boldsymbol{c}\|_1, \tag{20}$$

where $\|\boldsymbol{c}\|_1 = \sum_{k\in\mathcal{A}} |c_k|$. The penalty is a regularization term that favors low-rank solutions (Marelli et al., 2022). The LAR algorithm iteratively selects regressors from a candidate set, which is initialized to the complete set of the full-blown PCE, based on the correlation to the current residual. Contrary to both the quadrature and OLS methods, the LAR does not require the number of available samples L to be greater than the (total) number of PCE terms $K + 1$. Because of this feature, this method can be considered to be fully data-driven, as it tries to fit the best possible model given the available data. However, it should be noted the number of identifiable coefficients is limited by the relation $\max|\mathcal{A}| = \min(K + 1, L - 1)$. Therefore, the number of non-zero coefficients identified by LAR is bounded by the total amount of available data.

2.3.4. Stochastic Galerkin method

All the aforementioned methods are non-intrusive, meaning that they only require some evaluations of the actual function $\mathcal{M}(\boldsymbol{x})$ to calculate the coefficients. On the contrary, the stochastic Galerkin method (Ghanem and Spanos, 1991) operates on a (linear) system of equations by performing Galerkin projections to arrive at an augmented and coupled system of equations in the unknown coefficients, from which the dependence on the uncertain parameters is removed so that it can be solved deterministically. It is intrusive in that it requires access to the system of equations generating the model output y.

2.4. Limitations of the state-of-the-art methods

The main limitation of the stochastic Galerkin method is that it requires access to the underlying model equations to be modified by means of Galerkin projections. Therefore, it is hardly applicable to complex engineering designs, especially to those that require the use of proprietary software for their simulation. Moreover, the application of the Galerkin method to nonlinear systems is highly challenging. Projection methods and classical regression-based approaches share the same limitation of requiring a number of samples that is larger than the number of unknown coefficients to be estimated. While the latter may work with fewer samples, they are also less accurate. In this framework, the sparse regression method based on LAR turns out to be the most comparable to the proposed PCE-LSSVM method, and in general to machine learning techniques. Therefore, it shall be used as a reference throughout this paper.

Furthermore, for virtually any state-of-the-art method, the main limitation of the PCE is that it is a *parametric* model, i.e., its form (and hence, also the maximum achievable accuracy) is determined a priori, and the full set of basis functions must be developed in order to solve for the coefficients. Since the number of basis functions tends to grow exponentially, even for sparser truncation schemes, the classical PCE method becomes impractical in high dimensions and/or for problems that require large expansion orders.

To overcome the abovementioned limitations, we introduce in the following sections a novel nonparametric formulation of the PCE, in which the model training is cast as a LSSVM problem.

3. Least-square support-vector machines

In this section, we outline the second ingredient, i.e., the LSSVM method and, particularly, the main features that allow drawing the sought-for analogy with the PCE.

The LSSVM method, in its *primal space* formulation, seeks to find a model in the form of Suykens et al. (2002)

$$y \approx \hat{y} = \mathcal{M}_{\text{LSSVM}}(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{\varphi}(\boldsymbol{x}) \rangle + b = \sum_{k=1}^{K} w_k \varphi_k(\boldsymbol{x}) + b, \tag{21}$$

where $\boldsymbol{\varphi}(\boldsymbol{x}) = (\varphi_1(\boldsymbol{x}), \dots, \varphi_K(\boldsymbol{x}))$ is a vector of basis functions (not necessarily orthogonal). The model (21) is a parametric model with feature space $\boldsymbol{\varphi} : \mathbb{R}^d \to \mathbb{R}^K$ and is notably similar to (1), with the zero-order term explicitly represented by b.

The primal space model (21) is equivalently cast in the *dual space* formulation, which reads

$$\mathcal{M}_{\text{LSSVM}}(\boldsymbol{x}) = \sum_{l=1}^{L} \alpha_l k(\boldsymbol{x}, \boldsymbol{x}_l) + b, \tag{22}$$

where $\{\boldsymbol{x}_l\}_{l=1}^{L}$ is a set of "training samples" of the input parameters $\boldsymbol{x}$ and

$$k(\boldsymbol{x}, \boldsymbol{x}') = \left\langle \boldsymbol{\varphi}(\boldsymbol{x}), \boldsymbol{\varphi}(\boldsymbol{x}') \right\rangle = \sum_{k=1}^{K} \varphi_k(\boldsymbol{x}) \varphi_k(\boldsymbol{x}') \tag{23}$$

is a kernel function. In fact, the dual space model is a linear combination of kernel functions centered at the training samples. The coefficients $\boldsymbol{\alpha}$ and b are computed by solving the linear system

$$\begin{pmatrix} \boldsymbol{\Omega} + \boldsymbol{I}_L/\gamma & \boldsymbol{1}_L \\ \boldsymbol{1}_L^{\mathsf{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ b \end{pmatrix} = \begin{pmatrix} \boldsymbol{y} \\ 0 \end{pmatrix}, \tag{24}$$

where:

- $\boldsymbol{y} = (y_1, \dots, y_L)^{\mathsf{T}}$ is the vector of observations evaluated at the training samples, i.e., $y_l = \mathcal{M}(\boldsymbol{x}_l)$ for $l = 1, \dots, L$;
- $\boldsymbol{I}_L$ is the $L \times L$ identity matrix and $\boldsymbol{1}_L = (1, \dots, 1)^{\mathsf{T}} \in \mathbb{R}^L$ is a column vector of ones;
- γ is a regularization hyperparameter, which is optimized by minimizing some error metric, e.g., a cross-validation error;
- $\boldsymbol{\Omega}$ is a $L \times L$ Gram matrix with entries

$$\Omega_{lm} = k(\boldsymbol{x}_l, \boldsymbol{x}_m) = \sum_{k=1}^{K} \varphi_k(\boldsymbol{x}_l) \varphi_k(\boldsymbol{x}_m). \tag{25}$$

Moreover, the coefficients $\boldsymbol{\alpha}$ satisfy

$$\sum_{l=1}^{L} \alpha_l = 0. \tag{26}$$

Once the dual space coefficients $\boldsymbol{\alpha}$ are computed, the primal space coefficients $\boldsymbol{w}$ are obtained as

$$w_k = \sum_{l=1}^{L} \alpha_l \varphi_k(\boldsymbol{x}_l), \tag{27}$$

i.e., as a linear combination of the basis functions evaluated at the training samples.

There are two features that make the LSSVM model attractive. First of all, the dual space formulation (22) involves $L + 1$ terms, which are obtained by solving the $(L + 1) \times (L + 1)$ linear system (24), instead of $K + 1$ terms as in the primal space (21) or PCE (1) models. Hence, the complexity of the dual space model is determined by the available training data, which in general we want to be smaller than the size K of the feature space. Most importantly, the inner products in the kernel definition (23) do not need to be computed explicitly, since any (implicit) function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ that satisfies Mercer's condition, i.e.,

$$\iint_{\mathbb{R}^d \times \mathbb{R}^d} g(\boldsymbol{x}) k(\boldsymbol{x}, \boldsymbol{x}') g(\boldsymbol{x}') d\boldsymbol{x} d\boldsymbol{x}' \geq 0, \quad \forall g(\boldsymbol{x}) \in L^2, \tag{28}$$

is a valid kernel for the LSSVM (Mohri et al., 2018). Hence, both the kernel matrix $\boldsymbol{\Omega}$ in (24) and the model predictions in (22) can be computed by mere function evaluations, which is the so-called the "kernel trick". Popular implicit kernels are, e.g., the polynomial kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = (c + \boldsymbol{x}^{\mathsf{T}} \boldsymbol{x}')^p \tag{29}$$

and the squared-exponential (or RBF) kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2} \right), \tag{30}$$

where c, p, and σ are hyperparameters that are typically optimized as part of the training process. On the other hand, a kernel can be always built explicitly as in (23) or, more generally, as (Mohri et al., 2018)

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{k=1}^{K} \lambda_k \varphi_k(\boldsymbol{x}) \varphi_k(\boldsymbol{x}'), \tag{31}$$

which satisfies (28) provided that $\lambda_k > 0$, $\forall k$.

4. Nonparametric polynomial chaos formulation: Explicit kernel definition

This section introduces a first, preliminary formulation of the proposed nonparametric PCE method. The formulation leverages the explicit definition of the kernel function as the inner product of the feature space functions. While not being computationally efficient, we deem this intermediate step useful to understand the final framework.

It is argued that the primal space LSSVM model (21) is virtually identical to the PCE model (1) if the same set of orthonormal polynomials $(\psi_1(\boldsymbol{x}), \dots, \psi_K(\boldsymbol{x}))$ is taken as basis functions in the feature space. The kernel function is then built explicitly according to (23), leading to

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{k=1}^{K} \psi_k(\boldsymbol{x}) \psi_k(\boldsymbol{x}'). \tag{32}$$

With the above definitions, the primal space coefficients w_k in (21), obtained via (27), correspond to the PCE coefficients c_k for $k > 0$, whereas the zero-order coefficient c_0 corresponds to the bias term b. Hence, a nonparametric PCE model can be computed with the steps outlined in Algorithm 1.

It should be noted that the predictions $\boldsymbol{y}^*$ for an ensemble of points $\{\boldsymbol{x}_i^*\}_{i=1}^{N}$ can be compactly obtained as

$$\boldsymbol{y}^* = \boldsymbol{\Omega}^* \boldsymbol{\alpha} + b, \tag{33}$$

where $\boldsymbol{\Omega}_* \in \mathbb{R}^{N \times L}$ is a matrix with entries

$$\Omega_{il}^* = k(\boldsymbol{x}_i^*, \boldsymbol{x}_l) \tag{34}$$

for $i = 1, \dots, N$ and $l = 1, \dots, L$.

Hermite polynomials were already utilized in the kernel construction to improve the performance in support-vector machine classifiers (Moghaddam and Hamidzadeh, 2016). Moreover, a similar nonparametric formulation of the PCE was introduced in Cheng et al. (2017), Cheng and Lu (2018) to compute Sobol's sensitivity indices through support-vector regression, whose formulation is similar to the LSSVM primal space one. However, at present, this formulation still bears two important limitations:

1. The modeling accuracy is limited by the truncation of the PCE, i.e., by the maximum order of the polynomials that are used in constructing the kernel. Obviously, the same limitation affects also the classical PCE formulation (1).
2. The explicit kernel definition in (32) requires to build and evaluate all the K basis functions, which can make the evaluation of the kernel extremely intensive if L and/or K are large.

Algorithm 1: Nonparametric PCE

```
Generate a set of input training samples {x_l}_{l=1}^L;
for l = 1 to L do
    Collect corresponding observations: y_l ← M(x_l);
end
Build the L × L kernel matrix Ω according to (25), using φ_k = ψ_k, ∀k;
Find the coefficients α and b by solving the LSSVM system (24), possibly using cross-validation to optimise the hyperparameter γ;
Use (22) to compute predictions;
Get the first PCE coefficient: c_0 ← b;
for k = 1 to K do
    Retrieve the PCE coefficients c_k via (27), using φ_k = ψ_k;
end
```

Indeed, as already noted, the main advantage of kernel methods is that they can avoid the explicit calculation of the inner products using the kernel trick, i.e., by means of an implicit function evaluation. In other words, it is desirable to find an implicit kernel function that allows building matrix $\boldsymbol{\Omega}$ without going through the explicit generation of the basis functions. This is discussed in the next section.

5. Implicit kernel definition

In this section, we introduce implicit kernel definitions for Hermite and Legendre polynomials, as needed when the input parameters have Gaussian or uniform distribution, respectively. These implicit functions allow avoiding the explicit computation of the kernel as the inner product of a finite and potentially large number of feature space functions, thereby dramatically improving both the accuracy and the computational efficiency.

5.1. Implicit kernel for Hermite polynomials

Let us introduce the probability version (Kibble, 1945) of Mehler kernel in one dimension (Mehler, 1866), which reads

$$E(x,x'|\rho) = \frac{1}{\sqrt{1-\rho^2}} \exp\left(\frac{-\rho^2(x^2+x'^2)-2\rho x x'}{2(1-\rho^2)}\right) = \sum_{k=0}^{\infty} \frac{\rho^k}{k!} He_k(x) He_k(x'), \tag{35}$$

for $0 < \rho < 1$. By recalling (11), the kernel is rewritten as

$$E(x,x'|\rho) = \sum_{k=0}^{\infty} \rho^k \zeta_k(x)\zeta_k(x'). \tag{36}$$

Hence, the Mehler kernel allows for an implicit calculation of (31), with $\lambda_k = \rho^k$ and the orthonormal Hermite polynomials as basis functions in the feature space. Notably, for $\rho = 1$, it would correspond to the kernel of a one-dimensional Hermite PCE with an infinite number of terms! Unfortunately, the Mehler kernel has obviously a singularity for $\rho = 1$. Nevertheless, using $\rho < 1$ in (36) is equivalent to further rescaling the basis functions, which makes them no longer orthonormal. Indeed, by introducing the rescaled Hermite polynomials

$$\tilde{\zeta}_k(x) = \sqrt{\rho^k}\zeta_k(x) = \sqrt{\frac{\rho^k}{k!}} He_k(x), \tag{37}$$

we can further express (36) as

$$E(x,x'|\rho) = \sum_{k=0}^{\infty} \tilde{\zeta}_k(x)\tilde{\zeta}_k(x'). \tag{38}$$

Therefore, (35) can be used as an implicit LSSVM kernel, whose feature space functions are defined in (37). Once the dual space coefficients are obtained by solving (24), the primal space coefficients are obtained via (27). This leads to

$$\tilde{w}_k = \sum_{l=1}^{L} \alpha_l \tilde{\zeta}_k(x_l) = \sqrt{\rho^k} \sum_{l=1}^{L} \alpha_l \zeta_k(x_l), \tag{39}$$

where a tilde was used to highlight that the $\tilde{w}_k$ are the coefficients of the rescaled Hermite polynomials, and not of the standard orthonormal ones. The primal space LSSVM model therefore reads

$$\mathcal{M}_{\text{LSSVM}}(x) \approx \sum_{k=1}^{K} \tilde{w}_k \tilde{\zeta}_k(x) + b = \sum_{k=1}^{K} \sqrt{\rho^k}\tilde{w}_k \zeta_k(x) + b, \tag{40}$$

which, for a finite number of terms, is an approximation of the dual space model. It should be noted that there is no limitation on the value of K, and the number of terms in the primal space model can be increased at will. From (40), we obtain the standard PCE coefficients as $c_0 = b$ and

$$c_k = \sqrt{\rho^k}\tilde{w}_k = \rho^k \sum_{l=1}^{L} \alpha_l \zeta_k(x_l), \tag{41}$$

for $k > 1$. Hence, once the dual space coefficients $\boldsymbol{\alpha}$ are determined, the PCE coefficients are individually and inexpensively obtained as a linear combination of the corresponding basis polynomials evaluated at the training samples. It should be noted that, as opposed to the explicit definition in (32), the Mehler kernel also embeds the zero-order basis function $\zeta_0 = 1$. However, the corresponding coefficient c_0 cannot be reproduced with (41), which would lead to zero because of (26). This is reasonable since, as noted before, the zero-order term is actually provided by the bias term b.

For the multivariate case, the kernel is readily constructed as the product of univariate kernels (Vapnik, 1998)

$$k(\boldsymbol{x},\boldsymbol{x}'|\boldsymbol{\rho}) = \prod_{j=1}^{d} E(x_j, x'_j|\rho_j), \tag{42}$$

where $\boldsymbol{\rho} = (\rho_1, \dots, \rho_d)$. In general, the kernel parameters can differ for each dimension, thereby making the kernel *anisotropic*. The above kernel is readily shown to embed all basis function of an infinite series of multivariate Hermite polynomials. When such a kernel is used in a multivariate problem, the calculation of the coefficients in (41) generalizes to

$$c_{\boldsymbol{\kappa}} = \left(\prod_{j=1}^{d} \rho_j^{\kappa_j}\right) \sum_{l=1}^{L} \alpha_l \psi_{\boldsymbol{\kappa}}(\boldsymbol{x}_l), \tag{43}$$

for $\boldsymbol{\kappa} \in \mathcal{K}\backslash\{\mathbf{0}\}$, where the multi-index notation has been used for convenience.

The outlined nonparametric PCE formulation has several advantages compared to the state of the art. First of all, it does not directly suffer from the "curse of dimensionality", since there is no need to explicit build all basis functions simultaneously. Indeed, the PCE coefficients are retrieved in post processing using (43), by considering one basis function at a time and up to an arbitrary order, since the kernel inherently embeds all possible basis functions. The considered basis functions could be even selected adaptively. Furthermore, the formulation is equivalent to using a PCE with an infinite number of terms. This means that the accuracy is not limited a priori by a predefined expansion order. Rather, the model can represent any L^2 function with an arbitrary accuracy, provided that a sufficient number of training samples is used. The explicit kernel formulation (32) tends to approach the implicit one as the expansion order is increased. Finally, it is worth mentioning that in principle any value of $\rho \in (0,1)$ can be used. In fact, this is another kernel hyperparameter to be optimized in the training phase.

5.2. Implicit kernel for Legendre polynomials

A similar implicit kernel is found also for Legendre polynomials. Starting from the univariate and non-normalized Legendre polynomials (13), the kernel is defined as (Prudnikov et al., 1990; Anli and Gungor, 2007)

$$E(x,x'|\rho)=\frac{2K(u)}{\pi\sqrt{a-b}}=\sum_{k=0}^{\infty}\rho^k P_k(x)P_k(x'), \tag{44}$$

for $-1\le x,x'\le 1$ and $0<\rho<1$, where

$$u=\sqrt{\frac{2b}{b-a}},\quad a=1-2xx'\rho+\rho^2,\quad b=-2\rho\sqrt{1-x^2}\sqrt{1-x'^2} \tag{45}$$

and $K(u)$ is the complete elliptic integral of the first kind:

$$K(u)=\int_0^1\frac{dt}{\sqrt{(1-t^2)(1-u^2t^2)}}. \tag{46}$$

In (44), ρ plays a similar role as in (35), and it is therefore considered as a hyperparameter. Following the above definitions, we introduce the rescaled Legendre polynomials

$$\tilde{\xi}_k(x)=\sqrt{\rho^k}P_k(x). \tag{47}$$

Combining (44) and (47) with the dual space LSSVM formulation, the primal space coefficients are obtained as

$$\tilde{w}_k=\sum_{l=1}^{L}\alpha_l\tilde{\xi}_k(x_l)=\sqrt{\rho^k}\sum_{l=1}^{L}\alpha_l P_k(x_l). \tag{48}$$

Recalling (15), the primal space model is expressed in terms of the orthonormal Legendre polynomials as

$$\begin{aligned}\mathcal{M}_{\text{LSSVM}}(x)&\approx\sum_{k=1}^{K}\tilde{w}_k\tilde{\xi}_k(x)+b\\&=\sum_{k=1}^{K}\sqrt{\rho^k}\tilde{w}_k P_k(x)+b=\sum_{k=1}^{K}\sqrt{\frac{\rho^k}{2k+1}}\tilde{w}_k\zeta_k(x)+b,\end{aligned} \tag{49}$$

which is a finite-order approximation of the dual space model.

The multivariate kernel is constructed from the univariate kernel (44) as in (42). After solving the dual space LSSVM problem, the PCE coefficients for the multivariate orthonormal Legendre polynomials are retrieved as $c_0=b$ and

$$c_{\boldsymbol{\kappa}}=\left(\prod_{j=1}^{d}\frac{\rho_j^{\kappa_j}}{\sqrt{2\kappa_j+1}}\right)\sum_{l=1}^{L}\alpha_l\psi_{\boldsymbol{\kappa}}(\boldsymbol{x}_l), \tag{50}$$

for $\boldsymbol{\kappa}\in\mathcal{K}\backslash\{\mathbf{0}\}$.

5.3. Implementation

The proposed method is implemented in MATLAB. A Bayesian optimizer (Močkus, 1989; Garnett, 2023) is invoked to tune the hyperparameters, i.e., $\boldsymbol{\rho}\in(0,1)^d$ and $\gamma>0$, based on leave-out-out cross-validation error. This corresponds to a k-fold cross-validation with $k=L$ folds, i.e., as many as the number of available training samples. At each iteration of the optimizer, L models are built based on $L-1$ training samples and used to predict the observation at the left-out sample. This strategy is viable, as the training of a single LSSVM model is relatively quick. For large training datasets, employing cross-validation with a lower number of folds may represent a good tradeoff in favor of training efficiency.

Since the benefit of anisotropic kernels is found to be marginal, isotropic kernels will be used in all simulations, i.e., the same ρ is used for each input dimension. The availability of implicit kernels for Hermite and Legendre polynomials suggests that similar kernels may be found also for other orthogonal polynomials from the Wiener–Askey scheme.

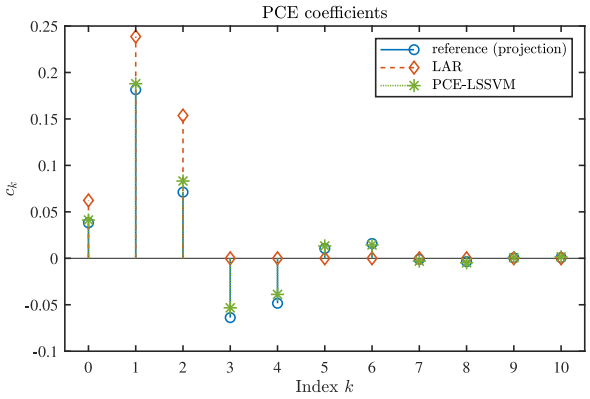

Fig. 1. Coefficients of a tenth-order Hermite PCE of function (51). Blue circles: reference results computed by projection; red diamonds and green asterisks: coefficients computed with the LAR and the proposed PCE-LSSVM formulation, respectively, based on $L=5$ training samples.

6. Illustrative example

Before considering realistic application examples, the advocated method is first illustrated based on an analytical one-dimensional function:

$$y=\mathcal{M}(x)=e^{-10/x}\left(2\cos(x)+\sin(x)\right). \tag{51}$$

The following error metrics are introduced to assess and compare the performance of the various methods:

- The RMSE between the target value and the model prediction for N randomly generated input samples, defined as

$$\text{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}. \tag{52}$$

  The RMSE has several advantageous features like handling the penalization of smaller errors, sharing the same units and scale as the QoI thereby being more interpretable, and being less sensitive to outliers (Bajaj, 2023).
- The coefficient of determination (R^2) over the same dataset, defined as

$$R^2=1-\frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{N}(y_i-\bar{y}_i)^2}, \tag{53}$$

  where

$$\bar{y}=\frac{1}{N}\sum_{i=1}^{N}y_i \tag{54}$$

  is the dataset mean. The coefficient of determination describes how much of the total variation in the QoI is explained by the variation in the model. If the sum of the squared error between the model predictions and the reference is small, R^2 will be close to 1 (ideal case), meaning that the model is able to capture nearly 100% of the variance in the QoI. Conversely, if the model is not able to capture any variance in the QoI, R^2 will be close to 0 or even negative (Bajaj, 2023).

In all simulations for this example, an independent test dataset with $N=10000$ samples is considered. Since the target function is analytical and one dimensional, the reference PCE coefficients are computed via an accurate numerical integration of (16).

6.1. Gaussian variability

We start by considering the case of a Gaussian distribution of the input parameter, with $x\sim\mathcal{N}(5,1)$. The reference coefficients of a

Table 1
Error metrics for the LAR and PCE-LSSVM methods for five independent runs with $L = 5$ training samples.

	run #1		run #2		run #3		run #4		run #5	
	LAR	PCE-LSSVM	LAR	PCE-LSSVM	LAR	PCE-LSSVM	LAR	PCE-LSSVM	LAR	PCE-LSSVM
RMSE	1.280×10^{-1}	$\mathbf{1.595 \times 10^{-2}}$	1.318×10^{-1}	$\mathbf{2.026 \times 10^{-2}}$	1.243×10^{-1}	$\mathbf{1.579 \times 10^{-2}}$	2.117×10^{-1}	$\mathbf{3.121 \times 10^{-2}}$	1.260×10^{-1}	$\mathbf{2.062 \times 10^{-2}}$
R^2	0.6343	**0.9944**	0.6125	**0.9909**	0.6553	**0.9948**	0	**0.9787**	0.6458	**0.9905**

tenth-order Hermite PCE are obtained by numerically computing the projection integral (16). The results are shown by the blue circles in the stem plot of Fig. 1. Next, we compute the same coefficients using a state-of-the-art sparse method based on LAR (Blatman and Sudret, 2011) and the proposed PCE-LSSVM formulation. For the former, the PCE module in the UQLab toolbox is used (Marelli et al., 2022). We choose LAR as a reference since it is one of the few PCE-based methods that is able to work with "sparse" data and compute PCE coefficients even when the amount of training samples is smaller than the total number of PCE terms, which is what we are targeting with the proposed PCE-LSSVM method. Other methods, such as OLS regression or quadrature-based approaches, would require a larger amount of training data and therefore would not be directly comparable. Indeed, for both methods, the same and rather small set of $L = 5$ training samples is considered, drawn according to the distribution of x. Owing to the Gaussian distribution of the input parameter x, the Mehler kernel is used in the PCE-LSSVM. The PCE coefficients are then retrieved from the dual space coefficients as outlined in Section 5. The results of the two approaches are shown in Fig. 1 by the red and green markers, respectively. It is observed that, despite the very low number of training samples, which is less than half of the coefficients, the values obtained via the PCE-LSSVM are much closer to the reference ones. Because of the low number of training samples, the coefficients for $k > 2$ estimated by the LAR are actually zero, and the UQLab model is in fact of order two.

Table 1 provides the error metrics achieved by the two methods. Since the performance may depend on the specific realization of the training samples, especially given the small number considered, the results are reported for five independent runs. For all the runs, the proposed PCE-LSSVM method substantially outperforms the LAR, achieving RMSE values that are almost one order of magnitude lower. It is important to mention that the RMSE and R^2 values of the PCE-LSSVM method are computed based on the primal space PCE model, which is truncated to order 10. Nonetheless, the difference w.r.t. the dual space model, which uses a kernel with an infinite number of expansion terms, is in this case negligible, meaning that the truncation error for a tenth-order PCE is marginal. This is also highlighted by the small magnitude of the last two coefficients, i.e., $c_9 = 6.2289 \times 10^{-4}$ and $c_{10} = 1.5422 \times 10^{-3}$, which provide a less than 0.01% contribution to the variance according to (7). Indeed, the difference between the primal space and dual space predictions or, equivalently, the drop in the coefficients' magnitude can be used as reasonable criteria to determine the PCE truncation when converting from the dual space to the primal space model.

Fig. 2 shows the results computed based on $L = 20$ training samples. The left panel compares the PCE coefficients. Now, two different maximum expansion orders are considered for the LAR, namely $p = 3$ (purple square markers) and $p = 10$ (red diamond markers). For the former, the LAR is able to provide an estimate for all coefficients. For the latter, only two additional coefficients are identified compared to the case with $L = 5$ samples, thereby yielding a fourth-order expansion. It is noted that the preselected expansion order affects the results. This is because the LAR is a parametric method whose output depends on the particular form of the regressors. Typically, in regression-based methods, the coefficients of a lower-order expansion are estimated with higher accuracy for a given training set size. Nevertheless, the error is still rather large for both expansion orders. On the contrary, the PCE-LSSVM coefficients now perfectly match the reference ones. Moreover, since they are computed in post-processing and individually for each basis function, the truncation order can be tuned at will, with no need to retrain the model. This highlights another important advantage in the operation of the proposed method.

The right panel of Fig. 2 shows instead the parametric approximation of the target function provided by the various models. The predictions of the third- and tenth- (or, in fact, fourth-) order models computed with UQLab (dashed purple and red lines, respectively) are compared with the approximation provided by the third-order primal space PEC-LSSVM model (dash-dotted yellow line) and by the dual space model (dotted green line). The tenth-order primal space PCE-LSSVM model is omitted since it is very close to the dual space one. The comparison shows that the accuracy of the dual space model, which is based on an infinite expansion, is mainly determined by the training samples. On the contrary, the accuracy of the primal space model is limited by the truncation. Overall, with 20 training samples, the dual space PCE-LSSVM models achieves excellent accuracy. Even when considering a low-order truncation in the primal space, however, the results are still more accurate than the corresponding ones obtained with the LAR, as shown by the accuracy of the coefficients. The superior performance is further confirmed by the error metrics reported in Table 2. It is noted that, for a given expansion order, the primal space PCE-LSSVM model provides higher accuracy compared to the LAR. Moreover, it is also confirmed that the tenth-order primal space model and the dual space one provide similar accuracy.

6.2. Uniform variability

The simulations are repeated by now assigning a uniform variability to the input parameter, with $x \sim \mathcal{U}(1, 9)$. Therefore, the Legendre kernel (44) is used. The coefficients of a tenth-order Legendre PCE are computed with both the LAR and the proposed PCE-LSSVM formulation by using $L = 10$ training samples. The results are provided in the left panel of Fig. 3 (red diamonds and green asterisks) and compared against the reference result obtained by projection (blue circles). In this case, the LAR returns only two null coefficients, i.e., c_2 and c_9. However, a significant error is obtained even for the non-zero coefficients, whereas the results achieved with the PCE-LSSVM are in excellent agreement with the actual coefficients. The superior accuracy of the PCE-LSSVM model is further appreciated in the right panel of Fig. 3, which compares the parametric approximation provided by the UQLab and PCE-LSSVM models. Despite an overall good accuracy in the center of the interval, the UQLab model (dashed red line) exhibits a large error towards the extrema of the interval. The PCE-LSSVM model is instead very accurate everywhere and, also in this case, negligible difference is found between the primal space and dual space models (dash-dotted yellow and dotted green lines, respectively).

Table 3 collects the relevant figures concerning the accuracy of the LAR and PCE-LSSVM methods for five independent runs. For this test case with uniform variability, it is once again confirmed that the PCE-LSSVM approach yields a more accurate model for a given expansion order and training dataset. In particular, the RMSE is about two orders of magnitude lower compared to the LAR result.

6.3. Implicit vs explicit kernels

As already noted, using implicit kernels provides two major advantages:

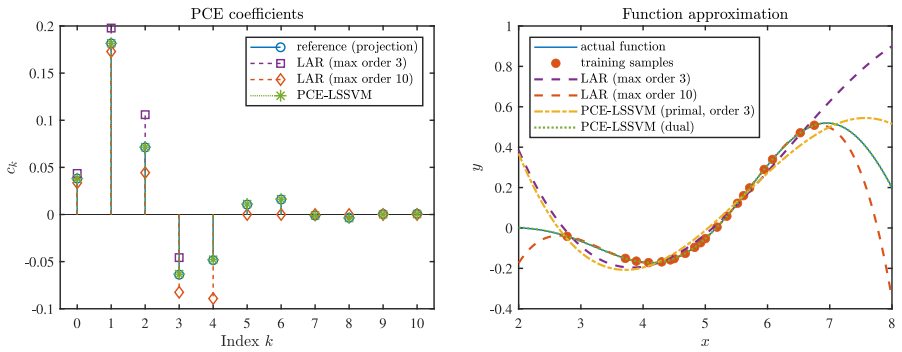

Fig. 2. Hermite PCE coefficients of function (51) (left) computed for different orders with $L = 20$ training samples and corresponding function approximation (right). Blue: reference; purple and red: models computed by means of LAR with maximum order $p = 3$ and $p = 10$, respectively; green: model obtained with the proposed PCE-LSSVM formulation. In the right panel, the dotted green and dash-dotted yellow curves correspond to the PCE-LSSVM dual space model and primal space model truncated to order three, respectively.

Table 2
Error metrics for the LAR and PCE-LSSVM models trained with $L = 20$ samples.

order	LAR		PCE-LSSVM (primal space)		PCE-LSSVM (dual space)
	$p = 3$	$p = 10$	$p = 3$	$p = 10$	–
RMSE	6.767×10^{-2}	5.341×10^{-2}	5.266×10^{-2}	9.920×10^{-4}	$\mathbf{9.496 \times 10^{-4}}$
R^2	0.8978	0.9363	0.9381	1.0000	**1.0000**

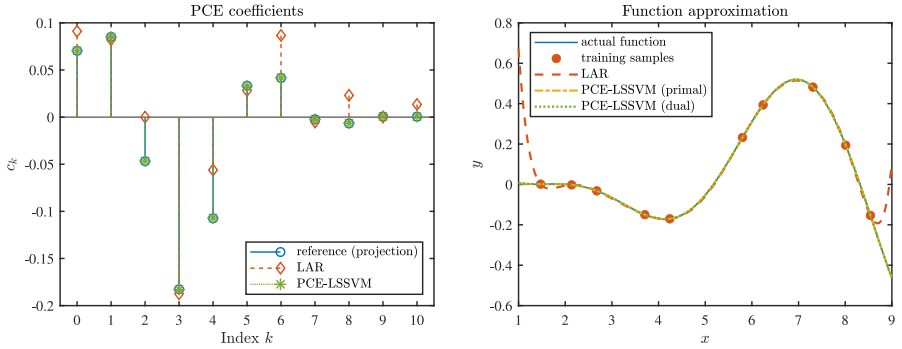

Fig. 3. Left panel: coefficients of a tenth-order Legendre PCE of function (51) computed with $L = 10$ training samples. The coefficients computed by means of LAR (red diamonds) and the proposed PCE-LSSVM formulation (green asterisks) are compared to the exact coefficients obtained by projection (blue circles). Right panel: corresponding function approximation. Solid blue line: actual function; dashed red line: UQLab model based on LAR; dash-dotted yellow and dotted green line: primal and dual space PCE-LSSVM models, respectively.

Table 3
Error metrics for the LAR and PCE-LSSVM methods for five independent runs with $L = 10$ training samples.

	run #1		run #2		run #3		run #4		run #5	
	LAR	PCE-LSSVM	LAR	PCE-LSSVM	LAR	PCE-LSSVM	LAR	PCE-LSSVM	LAR	PCE-LSSVM
RMSE	9.165×10^{-2}	$\mathbf{7.641 \times 10^{-4}}$	1.004×10^{-1}	$\mathbf{3.708 \times 10^{-3}}$	1.004×10^{-1}	$\mathbf{9.624 \times 10^{-4}}$	1.543×10^{-2}	$\mathbf{8.918 \times 10^{-4}}$	7.736×10^{-2}	$\mathbf{1.205 \times 10^{-3}}$
R^2	0.8532	**1.0000**	0.8238	**0.9998**	0.8237	**1.0000**	0.9958	**1.0000**	0.8954	**1.000**

1. As seen from the dual space LSSVM formulation (22), the order of the model is the same as the order of the kernel. Since implicit kernels have an infinite dimensionality, the achievable accuracy is inherently higher compared to explicit kernels, which are bounded by a predefined expansion order.
2. We do not need to form the inner product (32) explicitly, which would become intractable for high-dimensional problems with thousands of basis functions.

In this section, we compare the performance between explicit and implicit kernels in the examples of the previous two sections. For the explicit kernels, the same order as for the target PCE is considered, i.e., $p = 10$. It should be noted that the conversion from the dual space to the primal space is in this case exact. Table 4 compares the error metrics obtained with the LAR method and with the LSSVM formulation using both explicit and implicit kernels. The results for implicit kernels refer to the primal space (PCE) model although, as already noted, the difference w.r.t. the dual space model is negligible. It is observed that, while being generally better than LAR, the formulation using explicit kernels is substantially less accurate than the same formulation using implicit kernels. Therefore, it will not be considered further in the remainder of the paper.

7. Application examples: Circuit-level and full-wave simulations of electronic designs

In this section, the proposed technique is applied to the uncertainty quantification of electronic designs. Owing to the ever-growing miniaturization, the impact of manufacturing processes on multiple design parameters must be taken into account for a robust design. The structures are typically described by either circuit-level equations (relating voltages and currents) or electromagnetic equations (relating the electric and magnetic fields). Since the quantity of interest (QoI)

Table 4
Error metrics obtained for five independent runs with the LAR and PCE-LSSVM method, the latter implemented with both explicit and implicit kernels.

Run	Metric	Gaussian distribution						Uniform distribution		
		$L = 5$			$L = 20$			$L = 10$		
		LAR	PCE-LSSVM		LAR	PCE-LSSVM		LAR	PCE-LSSVM	
			Explicit	Implicit		Explicit	Implicit		Explicit	Implicit
#1	RMSE	1.312×10^{-1}	7.505×10^{-2}	$\mathbf{2.131 \times 10^{-2}}$	2.828×10^{-2}	1.460×10^{-3}	$\mathbf{8.189 \times 10^{-4}}$	1.1626×10^{-2}	6.0808×10^{-3}	$\mathbf{1.0021 \times 10^{-3}}$
	R^2	0.6159	0.8743	**0.9899**	0.9822	1.000	**1.000**	0.9976	0.9994	**1.000**
#2	RMSE	1.133×10^{-1}	6.345×10^{-2}	$\mathbf{1.993 \times 10^{-2}}$	5.728×10^{-2}	1.343×10^{-3}	$\mathbf{9.252 \times 10^{-4}}$	1.6403×10^{-2}	5.0544×10^{-2}	$\mathbf{9.0075 \times 10^{-4}}$
	R^2	0.7136	0.9101	**0.9911**	0.9268	1.000	**1.000**	0.9953	0.9553	**1.000**
#3	RMSE	1.189×10^{-1}	6.061×10^{-2}	$\mathbf{1.850 \times 10^{-2}}$	3.377×10^{-2}	1.509×10^{-3}	$\mathbf{7.441 \times 10^{-4}}$	9.6497×10^{-2}	7.2710×10^{-2}	$\mathbf{1.5899 \times 10^{-3}}$
	R^2	0.6843	0.9180	**0.9924**	0.9746	1.000	**1.000**	0.8372	0.9076	**1.000**
#4	RMSE	3.155×10^{-1}	2.017×10^{-1}	$\mathbf{1.410 \times 10^{-2}}$	3.315×10^{-2}	1.517×10^{-3}	$\mathbf{9.031 \times 10^{-4}}$	1.0751×10^{-2}	2.2609×10^{-3}	$\mathbf{7.3802 \times 10^{-4}}$
	R^2	−1.2223	0.0924	**0.9956**	0.9755	1.000	**1.000**	0.9980	0.9999	**1.000**
#5	RMSE	3.276×10^{-1}	7.677×10^{-2}	$\mathbf{2.186 \times 10^{-2}}$	2.793×10^{-3}	1.455×10^{-3}	$\mathbf{1.043 \times 10^{-3}}$	8.1370×10^{-2}	7.9287×10^{-2}	$\mathbf{6.3323 \times 10^{-4}}$
	R^2	−1.3956	0.8685	**0.9893**	0.9826	1.000	**1.000**	0.8843	0.8901	**1.000**

Table 5
Nominal geometrical parameters for the transmission line sections in the network of Fig. 4.

Parameter	w_1	w_2	w_3	d_1	d_2	d_3	g	h	t_1	t_2	t_3
Nominal value	150 μm	130 μm	170 μm	100 μm	140 μm	70 μm	150 μm	200 μm	30 μm	20 μm	40 μm

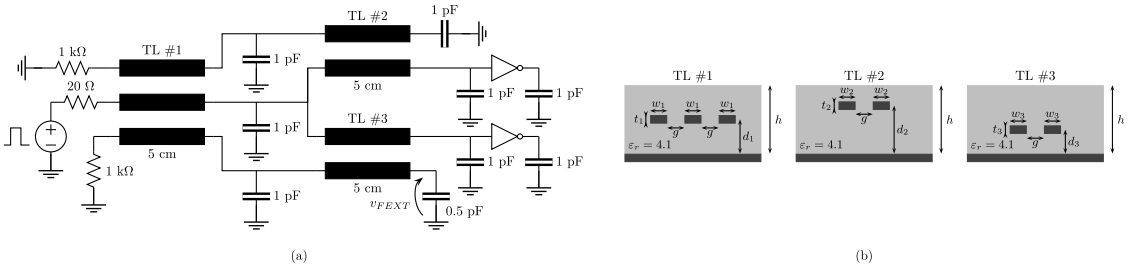

Fig. 4. Schematic of the multicondutor transmission line network for the first application test case (a) and cross-sectional geometry of the coupled transmission line sections (b).

is often a function of time or frequency, and multiple QoIs may be present, particular care has to be taken in handling such large datasets. A naive approach would be to train a separate model for each output and time or frequency point of interest. However, this strategy turns out to be highly inefficient, since the training cost is usually low, yet not negligible. An effective approach based on dimensionality reduction via principal component analysis (PCA) was proposed in Manfredi and Trinchero (2020) and shall be adopted in this paper. The strategy is readily applied in conjunction with any data-driven surrogate modeling technique (e.g., Bhattacharyya (2022), Pulch and Youssef (2020)). Further details will be provided later on, as needed. Moreover, for frequency-domain simulations, the QoIs are complex-valued. In this case, the real and the imaginary parts are handled separately, and the squared magnitude of the difference is taken in the calculation of the error metrics (52) and (53).

7.1. Test case #1: Crosstalk in a multiconductor transmission line network

The first application example considers the multiconductor transmission line network of Fig. 4, which consists of a connection of lumped RC elements, distributed transmission line sections, and nonlinear loads (inverters) (Ahadi and Roy, 2016). The nominal values of the geometrical parameters of the transmission lines are provided in Table 5. We consider three different simulation scenarios with an increasing number of uncertain parameters. We investigate the voltage coupling from the source on the left to the 0.5-pF termination on the bottom right, which is termed "far-end crosstalk" (v_{FEXT}). The excitation is a pulse with an amplitude of 5 V, a duration of 1 ns, and rise/fall times of 100 ps. The simulations are carried out using a Simulation Program with Integrated Circuit Emphasis (SPICE) (Synopsis Inc., 2009), an industry standard for circuit-level simulations (Vladimirescu, 1994). For each scenario, reference results are generated via the MC simulation of an independent test dataset with $N = 5000$ random configurations of the uncertain parameters, drawn according to a Latin hypercube scheme.

7.1.1. Scenario (a): Maximum crosstalk

In this first scenario, the variability is caused by six independent geometrical parameters of the transmission lines, namely the trace widths w_j and their distances d_j from the ground plane (with $j = 1, 2, 3$). An independent Gaussian variation with a relative standard deviation of 10% is assumed. The variation is considered to be the same for all the traces and distances within the same section. The gap between the traces g, the substrate thickness h, and the trace thicknesses t_1, t_2, t_3, are considered to be deterministic parameters. We take the maximum crosstalk voltage occurring over time as the QoI. This is particularly challenging, since the location and the sign of the maximum crosstalk level may vary non-smoothly with the uncertain parameters.

The UQLab toolbox and the proposed PCE-LSSVM formulation are used to compute a third-order Hermite PCE leveraging a total degree truncation, which leads to 84 basis functions. For the latter, an isotropic Mehler kernel is used. Although with the PCE-LSSVM method the expansion order can be increased at will without re-training the model, we fix it to $p = 3$ for a direct comparison with LAR, which requires to define the expansion order a priori. Both models are trained using an increasing number of $L = \{10, 20, 30\}$ training samples. For each parameter configuration, a transient simulation is performed and the maximum of the absolute value of crosstalk over time is taken as the QoI.

The results are collected in Table 6, which provides the mean value and the standard deviation of the QoI predicted by the various methods, as well as the RMSE between the surrogate models and the MC samples. It is noted that, for all training set sizes, the PCE-LSSVM method achieves better accuracy than LAR in terms of RMSE. Moreover,

Table 6
Mean value, standard deviation, and RMSE versus MC obtained with LAR and PCE-LSSVM methods for three different training datasets with increasing number of samples.

	MC	$L = 10$		$L = 20$		$L = 30$	
		LAR	PCE-LSSVM	LAR	PCE-LSSVM	LAR	PCE-LSSVM
Mean (V)	0.4761	0.4842	**0.4787**	**0.4755**	0.4768	**0.4759**	0.4757
Std (V)	0.0549	0.0583	**0.0571**	0.0518	**0.0541**	**0.0545**	0.0542
RMSE (V)	–	5.4466×10^{-2}	$\mathbf{7.6828 \times 10^{-3}}$	9.2424×10^{-3}	$\mathbf{5.1324 \times 10^{-3}}$	9.8703×10^{-3}	$\mathbf{4.8779 \times 10^{-3}}$

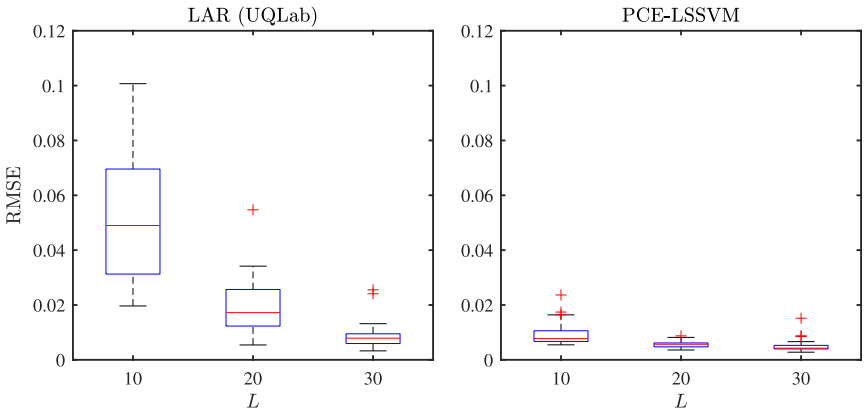

Fig. 5. RMSE obtained with the LAR (left) and PCE-LSSVM (right) methods over 50 independent runs and different training set sizes.

it also provides more accurate or comparable estimates of the mean and standard deviation for the training datasets with $L = 10$ and $L = 20$ samples. However, the convergence rate seems to be lower, and LAR provides a more accurate estimate with the largest dataset.

For a more robust assessment of the performance in terms of accuracy and convergence, the analysis is repeated for 50 independent runs for each training set size, and the results for the RMSE are provided in Fig. 5. The two boxplots allows appreciating the dispersion of the error achieved by the two methods across the various runs and for the different training set sizes. In particular, the red line indicates the median, the bottom and top edges are the 25th and 75th percentiles, respectively, whereas the whiskers indicate the minimum and maximum values. The red crosses are outliers. This analysis shows that, for this test case, the advocated PCE-LSSVM formulation significantly outperforms the LAR, achieving both a lower median error (indicated by the lower red bar) and a lower dispersion across different training datasets (indicated by the narrower box size).

7.1.2. Scenario (b): Transient crosstalk

We now consider the entire transient crosstalk as the QoI, computed at $M = 5001$ equally-spaced time points over a window of 50 ns. Moreover, in this second scenario, the variability is provided by the trace widths w_1, w_2, w_3, and the trace gap g, which are now considered to vary independently also within the same section, leading to a total of 11 independent uncertain parameters. The same nominal values as in the first test scenario are considered, again with a Gaussian distribution and a 10% standard deviation.

The naive modeling of the crosstalk at each time point as a separate output would require to train 5001 surrogates, which is impractical. Rather, we build a $M \times L$ training dataset $\boldsymbol{Y}$ and we apply PCA (Manfredi and Trinchero, 2020), leading to a compressed dataset

$$\tilde{\boldsymbol{Z}} = \tilde{\boldsymbol{U}}^{\mathsf{T}}(\boldsymbol{Y} - \bar{\boldsymbol{Y}}), \tag{55}$$

where $\bar{\boldsymbol{Y}}$ is the dataset mean over the L training samples and $\tilde{\boldsymbol{U}}$ is a $M \times \tilde{n}$ matrix with the first $\tilde{n}$ left-singular vectors of $\boldsymbol{Y} - \bar{\boldsymbol{Y}}$. A truncation on the singular values σ_n is considered, such that

$$\frac{\sigma_n}{\sigma_1} < \delta \tag{56}$$

for $\forall n > \tilde{n}$. A threshold of $\delta = 10^{-2}$ is used in this paper. Hence, the dataset $\tilde{\boldsymbol{Z}}$ is of size $\tilde{n} \times L$, typically with $\tilde{n} \ll M$, and only $\tilde{n}$ surrogates need to be trained. Predictions for the original outputs are obtained by reverting (55), leading to a prediction dataset

$$\hat{\boldsymbol{Y}} = \bar{\boldsymbol{Y}} + \tilde{\boldsymbol{U}}\,\hat{\tilde{\boldsymbol{Z}}} \tag{57}$$

starting from matrix $\hat{\tilde{\boldsymbol{Z}}}$, which contains the surrogate predictions for the $\tilde{n}$ principal components.

Three training datasets with $L = \{15, 30, 110\}$ are considered, which are compressed to 13, 15, and 18 principal components, respectively, thereby achieving compression factor of over 99.6% in all cases. These datasets are used to train a third-order Hermite PCE with total degree truncation, which features up to 364 terms. Once again, we aim at using an amount of training data that is much smaller than the number of terms in the full-blown PCE, which makes LAR one of the only viable solutions for the calculation of reference coefficients.

Fig. 6 collects the results, restricted to a window of 15 ns for the sake of readability. For each training dataset, the top panels show the standard deviation of the transient crosstalk computed from the MC samples (solid blue line), with the UQLab surrogate based on LAR (dashed red line), and with the primal space PCE-LSSVM model (dotted green line). The bottom panels provide instead the RMSE. The plots indicate that the standard deviation obtained with the proposed PCE-LSSVM formulation better matches the reference result from the MC analysis, especially when the number of training samples is small. This is also confirmed by the (much) smaller RMSE, particularly in the initial part of the transient. For the largest training dataset, the two approaches achieve a similar accuracy. Therefore, also for this example, it is confirmed that the proposed method yields a more accurate model compared to the LAR when the number of training samples is small.

The above conclusion is further corroborated by Fig. 7, showing the probability density function (PDF) of the crosstalk voltage at 1.7 ns (top panels) and 3.2 ns (bottom panels) obtained with the three training datasets. These were selected as two time points at which the crosstalk voltage exhibits significant variability. It is observed that the PDFs obtained from the PCE-LSSVM model trained with 15 samples are remarkably accurate and do not substantially differ from the ones obtained by using 110 training samples, and they are in good agreement with the reference MC distribution. On the contrary, the UQLab model attains a similar accuracy only with the largest training dataset.

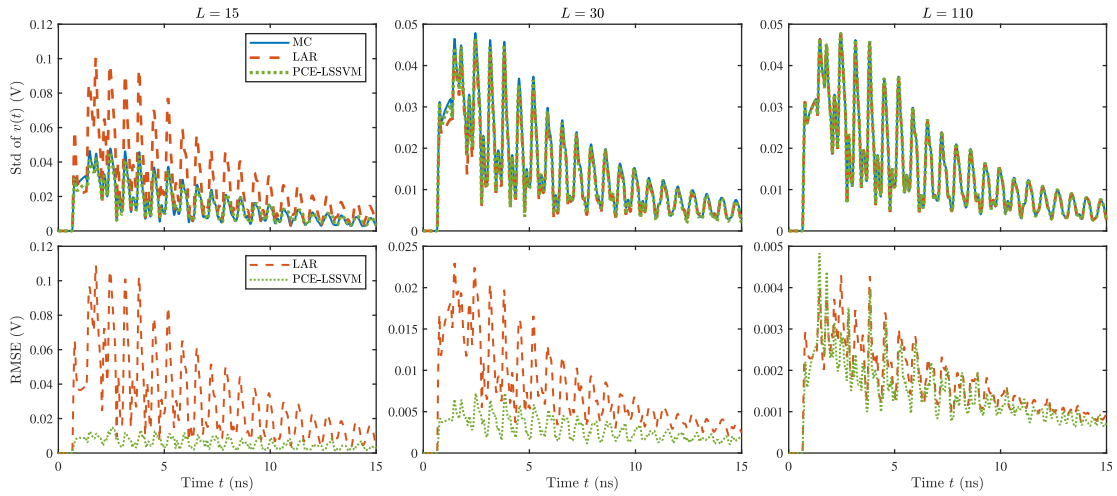

Fig. 6. Standard deviation (top panels) and RMSE (bottom panels) obtained with the PCE surrogates trained with LAR (dashed red lines) and PCE-LSSVM (dotted green lines) using datasets of increasing size. In the top panels, the solid blue line is the reference standard deviation from the MC samples.

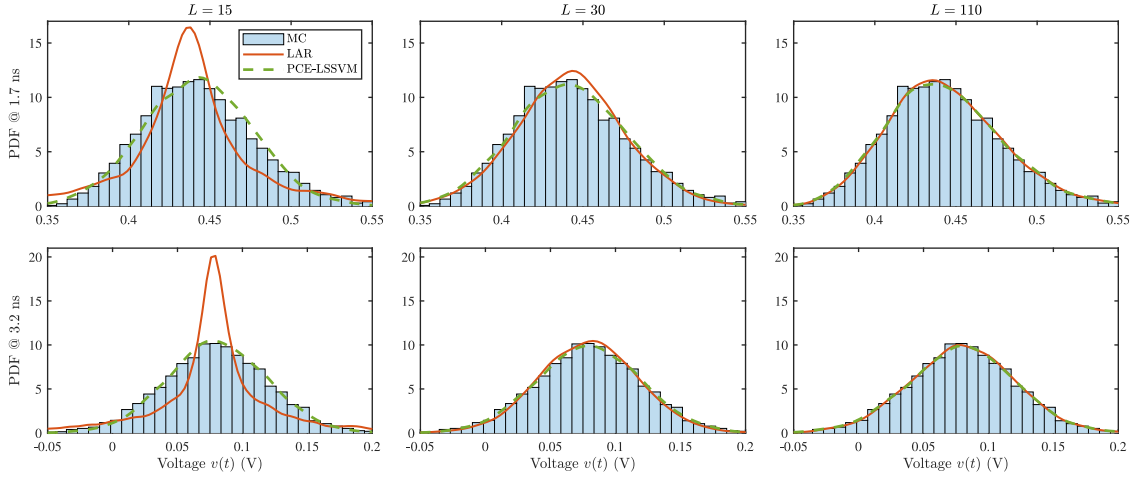

Fig. 7. PDF of the crosstalk voltage at 1.7 ns (top panels) and 3.2 ns (bottom panels). The distribution of the MC samples (blue bars) is compared against the PDF obtained from the PCE-based surrogates trained with the LAR (solid red line) and PCE-LSSVM (dashed green line) using $L = 15$ (left), $L = 30$ (center), and $L = 110$ samples (right).

7.1.3. Scenario (c): Maximum crosstalk in high-dimensional settings

In order to push the proposed method to high-dimensional settings, we further increase the number of independent uncertain parameters to 26. Now all the geometrical parameters indicated in Fig. 4 and Table 5, i.e., the trace widths w_1, w_2, w_3, the trace thicknesses t_1, t_2, t_3, the trace distances from the ground plane d_1, d_2, d_3, the gap between the traces g, and the substrate thickness h, vary with a Gaussian distribution and a 10% relative standard deviation. The variation is independent for each trace and section with the exception of the substrate thickness h, for which a common variation is assumed for all the three sections. This is reasonable since all transmission lines lie on the same substrate. Therefore, in addition to the substrate thickness, there are 11 independent uncertain parameters in the first transmission line section and 7 in both the second and third sections. As in the first scenario, we focus the analysis on the maximum crosstalk over time. We use both LAR and the proposed PCE-LSSVM to compute a third-order Hermite PCE, which now features up to 3654 terms, based on three training datasets with $L = \{40, 60, 80\}$ samples only.

Fig. 8 compares the accuracy in terms of RMSE obtained with the LAR and the PCE-LSSVM based on 50 independent runs. Also for this scenario, the proposed PCE-LSSVM achieves a lower RMSE compared to LAR. Specifically, the upper whiskers of the PCE-LSSVM boxes are lower than the red lines of the LAR boxes, indicating that the largest error obtained with the advocated technique is lower than the median error of the reference LAR method. This example confirms the good performance of the proposed method also in a high-dimensional scenario.

7.2. Test case #2: Microstrip line with ground plane discontinuity

The second test case deals with the microstrip transmission line with ground plane discontinuity depicted in Fig. 9 (Manfredi and Grivet-Talocia, 2021). Such a ground plane discontinuity affects the return path of the current propagating along the line, causing a significant degradation of the signal integrity. The electrical performance is characterized by means of the scattering parameters S_{11} (return loss) and S_{21} (insertion loss), which are complex-valued and frequency-dependent quantities that describe the signal reflection and transmission along the line, respectively. For such a passive structure, their magnitude is limited between 0 and 1, respectively meaning zero and total reflection/transmission. The magnitude is often expressed in decibels (dB), therefore ranging between $-\infty$ and 0 dB. The scattering parameters are computed by means of electromagnetic full-wave simulations at 1859 equally-spaced frequency points from dc to 10 GHz, performed via the CST Studio Suite® software from Dassault Systemès (Dassault Systemès, 2022).

The variability is provided by the length and position of the slot in the ground plane, which are ascribed a Gaussian distribution with a

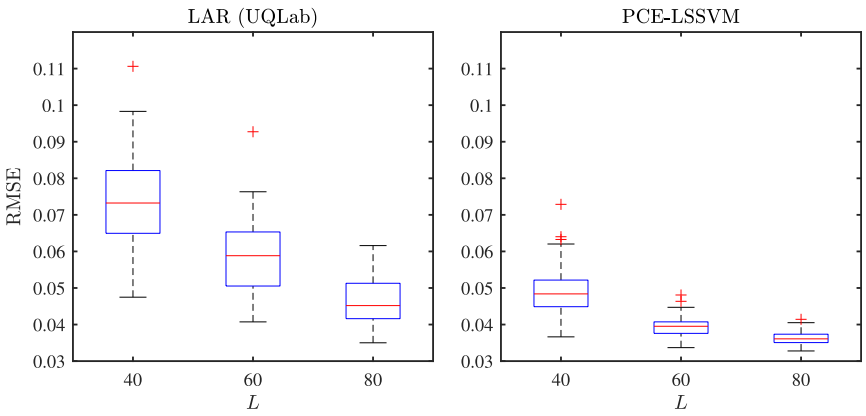

Fig. 8. RMSE obtained with the LAR (left) and PCE-LSSVM (right) methods over 50 independent runs and different training set sizes for the scenario with 26 independent uncertain parameters.

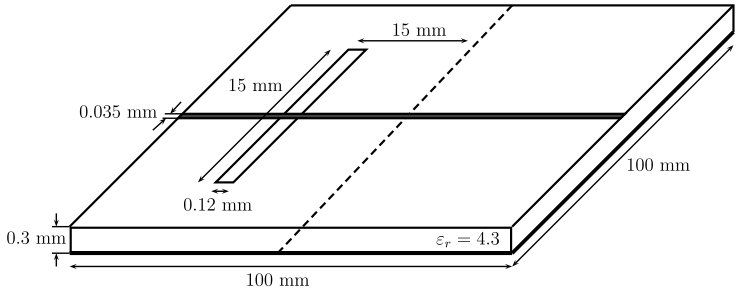

Fig. 9. Geometry of the microstrip transmission line with ground plane discontinuity.

10% relative standard deviation. This example is particularly challenging because the large variability of the scattering parameters at high frequency requires high expansion orders, unless a rational model is used (Manfredi and Grivet-Talocia, 2021). Therefore, a Hermite PCE of order $p = 10$ is trained using $L = 50$ samples. As before, PCA is used to compress the training dataset and limit the amount of surrogate models to build. The original training dataset, consisting of 50 samples of the response computed at 1859 frequency points for each of the four scattering parameters S_{ij} (with, $i, j = 1, 2$), is compressed to a mere 50 samples of 15 principal components only. Reference results are generated with a MC analysis based on a test dataset with $N = 1000$ configurations of the uncertain parameters.

Fig. 10 shows the results for the return loss (left panels) and insertion loss (right panels). The top panels provide a subset of MC responses (solid gray lines) as well as the average response computed from the MC samples (solid blue line), from the PCE model trained with LAR (dashed red lines), and with the primal space PCE-LSSVM model (dotted green lines). A large variability is observed above 6 GHz, especially in the insertion loss. This indicates that the transmission of information is significantly affected around that frequency. In the higher frequency range, a larger error on the average is also established for the LAR model compared to the PCE-LSSVM one. The superior accuracy of the PCE-LSSVM model is further confirmed by the bottom panels, which provide the variance and the RMSE. A much lower error is obtained with the PCE-LSSVM in the entire frequency range.

Fig. 11 shows the PDF of the magnitude of S_{11} (left) and S_{21} (right), computed at the frequencies of 6.2 GHz (top panels) and 7.3 GHz (bottom panels). These values are selected as two frequencies at which the scattering parameters exhibit a large variability (cfr. Fig. 10). In this case, the distribution of the MC samples (solid blue lines) is compared against the results of two PCE models, one with order $p = 3$ and trained with $L = 20$ samples and one with order $p = 10$ and trained with $L = 50$ samples. The former requires to train a surrogate for 12 principal components, in place of the 15 surrogates required by the larger dataset, whereas the latter is the same model as considered in Fig. 10. The comparisons highlight once more that the PCE-LSSVM model (dashed green lines) achieves a much higher accuracy compared to the LAR model with the same order and training dataset (solid red lines). In particular, a lower discrepancy is observed between the two PCE-LSSVM models, with the tenth-order one being in excellent agreement with the reference MC distribution. This is consistent with the results of Manfredi and Grivet-Talocia (2021), which showed that a single PCE of low order provides an inaccurate model at high frequency.

Table 7
Nominal parameters for the transmission line sections and BJT in the schematic of Fig. 12.

Parameter	Nominal value
Microstrip trace widths w_1, w_2, w_3	0.25 mm
Microstrip trace width w_4	0.8 mm
Forward current gain	145
Base-emitter depletion capacitance	0.3109 pF
Base-collector depletion capacitance	0.1377 pF
Collector substrate capacitance	0.6675 pF
Collector inductance	1.1 nH
Base inductance	1.1 nH
Emitter inductance	0.25 nH
Base-emitter substrate resistance	25 Ohm
Base-emitter bonding pad capacitance	145 fF
Collector–emitter substrate resistance	19 Ohm
Collector–emitter bonding pad capacitance	145 fF

7.3. Test case #3: Low noise amplifier

The third and last application test case considers the low noise amplifier (LNA) whose schematic is illustrated in Fig. 12 (Buss, 1996). The LNA makes use of a bipolar junction transistor (BJT) and is designed to amplify an electrical signal at the frequency of 2 GHz with a nominal gain of 16 dB when operating with a current of 5 mA at the collector

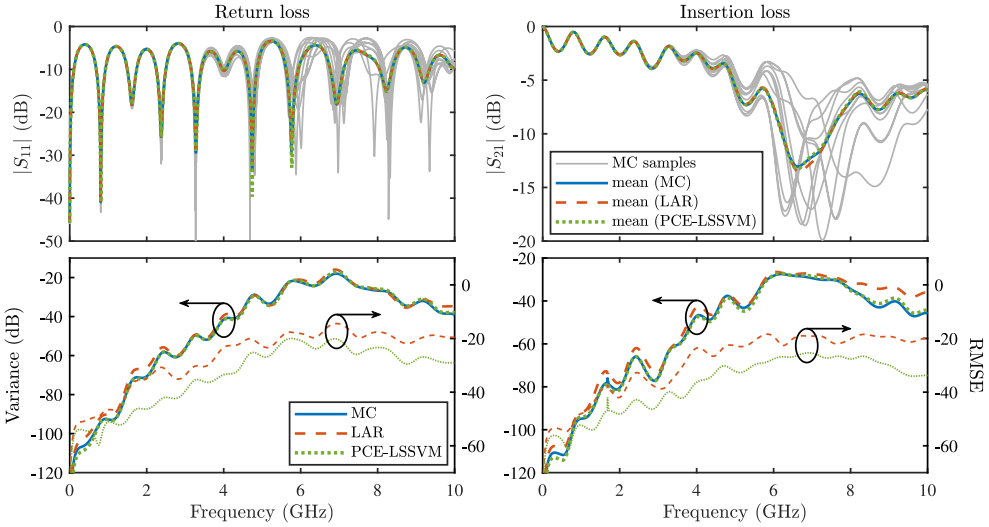

Fig. 10. Variability analysis of the return loss (left panels) and insertion loss (right panels) of the microstrip line with ground plane discontinuity. The top panels show a subset of responses from the MC analysis (solid gray lines) as well as the average response computed from the MC samples (solid blue line) and with the LAR and PCE-LSSVM models (dashed red and dotted green lines, respectively). The bottom panels provide the RMSE instead, using the same color scheme.

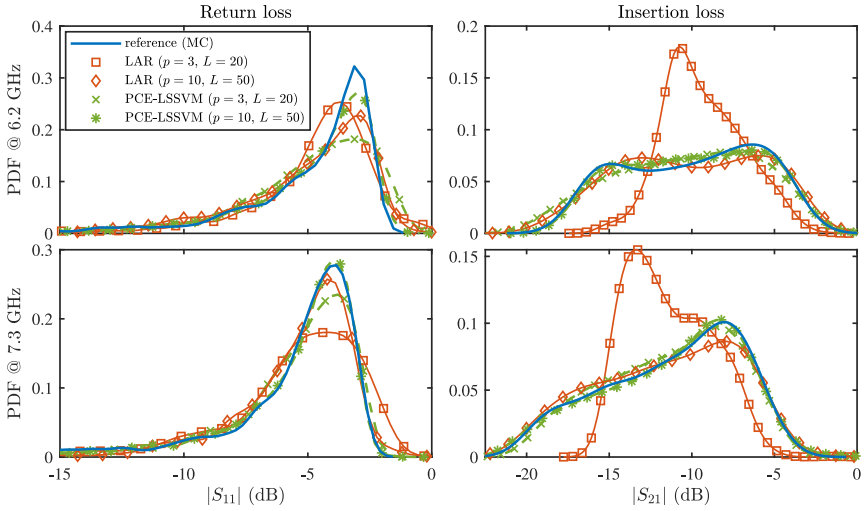

Fig. 11. PDF of the magnitude of the return loss (left panels) and insertion loss (right panels) computed at 6.2 GHz (top panels) and 7.3 GHz (bottom panels). Solid blue lines: reference distribution from the MC samples; solid red lines: PDF obtained from the PCE model of order $p = 3$ (square markers) and $p = 10$ (diamond markers) trained with LAR; dashed green lines with cross and diamond markers: PDF obtained from the same PCE models trained with the PCE-LSSVM formulation.

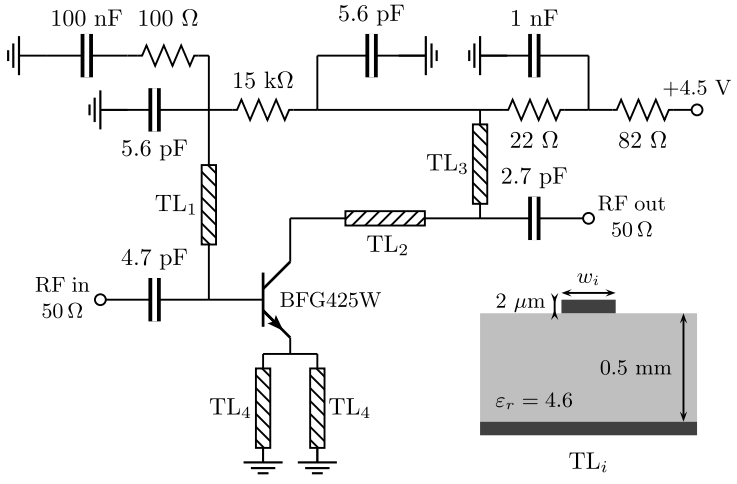

Fig. 12. Schematic of the low noise amplifier.

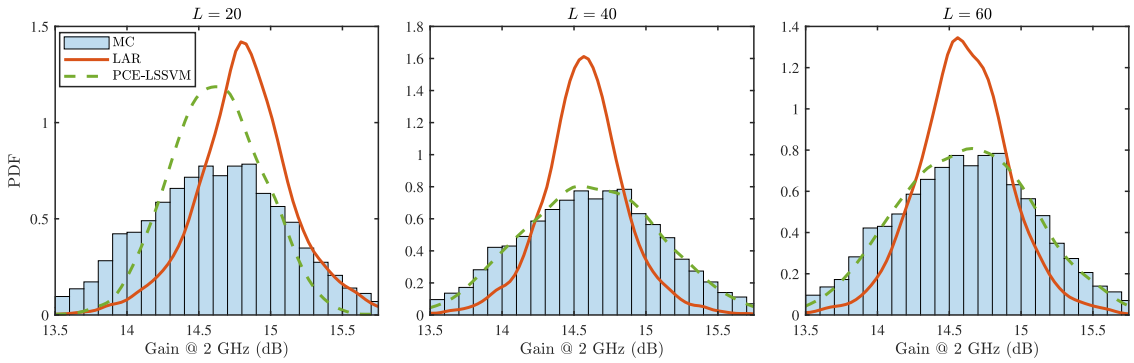

Fig. 13. PDF of the LNA gain at 2 GHz. The distribution of the MC samples (blue bars) is compared against the PDF obtained from the PCE model with coefficients computed by means of LAR (solid red line) and PCE-LSSVM (dashed green line) using $L = 20$ (left), $L = 40$ (center), and $L = 60$ samples (right).

terminal. Specifically, the BJT with part number BFG425W produced by NXP Semiconductors N.V. is used for the design. The circuit is simulated in HSPICE to calculate the insertion loss (i.e., the scattering parameter S_{21}) between the radio frequency (RF) input and output ports via a small-signal analysis. The magnitude of S_{21} at the operating frequency (i.e., 2 GHz) corresponds to the gain. The SPICE model of the transistor is available from the vendor website (N.V., 2012).

In this example, the variability is provided by 25 independent uncertain parameters, which are summarized below:

- The resistance of the four resistors shown in the schematic of Fig. 12;
- The capacitance of the six capacitors in the schematic of Fig. 12;
- The widths of the four microstrip transmission line sections (i.e., TL_i, with $i = 1, \dots, 4$);
- 11 parameters within the BJT model, which are listed in Table 7.

The nominal values of the resistors and capacitors are indicated in Fig. 12, whereas the nominal value of the uncertain transmission line and BJT parameters are indicated in Table 7. The Reader is referred to the datasheet and the SPICE model of the BJT for the definition of the internal parameters of the transistor. All parameters have a uniform distribution within ±20% around the nominal value. Both LAR and the proposed PCE-LSSVM method are used to construct a third-order Legendre PCE, which features up to 3276 terms. Three training datasets with $L = \{20, 40, 60\}$ samples are considered. Reference results are generated via the MC simulation of an independent test dataset with $N = 5000$ samples.

Fig. 13 compares the PDFs obtained with LAR (solid red lines) and PCE-LSSVM (dashed green lines) against the reference distribution from the MC simulation (blue histogram). For a given training set size, the distribution obtained with the proposed method agrees better with the reference. In particular, with both $L = 40$ and $L = 60$ samples, the distribution of the PCE-LSSVM model is in excellent agreement with the MC one.

The excellent accuracy achieved with a small training dataset by the PCE-LSSVM method is further confirmed by the boxplots in Fig. 14. It is once again established that the advocated technique achieves a lower error for a given training set size as well as a lower dispersion across different training datasets. In particular, the median error obtained by the PCE-LSSVM is two to three times lower than the one achieved with the LAR.

8. Benchmark functions

In this section, we consider popular benchmark functions for stochastic methods to further validate the proposed PCE-LSSVM method. These functions are typically designed to stress the features of the tested methods. In particular, we introduce the following test functions of increasing dimensionality.

8.1. Ishigami function

The Ishigami function (Ishigami and Homma, 1990) is a popular three-dimensional benchmark for UQ and sensitivity analysis methods (cfr., e.g., Schobi et al., 2015). It is defined as

$$y = \mathcal{M}(\boldsymbol{x}) = \sin(x_1) + 7\sin^2(x_2) + 0.1x_3^4 \sin(x_1), \tag{58}$$

with $x_i \sim \mathcal{U}(-\pi, \pi)$, for $i = 1, 2, 3$. Given the high nonlinearity, a large expansion order is necessary for PCE-based methods (cfr. Blatman and Sudret, 2011). Therefore, we train a tenth-order Legendre expansion for this function.

8.2. Sobol' function

The Sobol' function (Soboĺ, 1993) is eight-dimensional with uniform inputs:

$$y = \mathcal{M}(\boldsymbol{x}) = \prod_{i=1}^{8} \frac{|4x_i - 2| + c_i}{1 + c_i}, \tag{59}$$

where $x_i = \mathcal{U}(0, 1)$, for $i = 1, \dots, 8$, and $\boldsymbol{c} = (1, 2, 5, 10, 20, 50, 100, 500)$. It is a non-smooth function around $x_i = 0.5$, which we aim to model with a third-order Legendre expansion.

8.3. O'Hagan function

The O'Hagan function was defined in Oakley and O'Hagan (2004) as a 15-dimensional benchmark with Gaussian inputs:

$$y = \mathcal{M}(\boldsymbol{x}) = \boldsymbol{a}_1^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{a}_2^{\mathsf{T}}\sin(\boldsymbol{x}) + \boldsymbol{a}_3^{\mathsf{T}}\cos(\boldsymbol{x}) + \boldsymbol{x}^{\mathsf{T}}\boldsymbol{Q}\boldsymbol{x}, \tag{60}$$

where $x_i \sim \mathcal{N}(0, 1)$, for $i = 1, \dots, 15$. We refer to Oakley and O'Hagan (2004) for the definition of vectors $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$ and matrix $\boldsymbol{Q}$. We model the O'Hagan function using a third-order Hermite PCE.

8.4. Morris function

The Morris function is a 20-dimensional benchmark with uniform inputs that was originally defined in Morris (1991). Since the original definition featured some random coefficients, we consider here the modified function introduced in Schobi et al. (2015), i.e.,

$$y = \mathcal{M}(\boldsymbol{x}) = \sum_{i=1}^{20} \beta_i w_i + \sum_{i<j}^{20} \beta_{ij} w_i w_j + \sum_{i<j<l}^{20} \beta_{ijl} w_i w_j w_l + 5 w_1 w_2 w_3 w_4, \tag{61}$$

where (Blatman, 2009)

$$w_i = \begin{cases} 2\left(\dfrac{1.1x_i}{x_i + 0.1} - 0.5\right) & i = 3, 5, 7 \\ 2(x_i - 0.5) & \text{otherwise} \end{cases}, \tag{62}$$

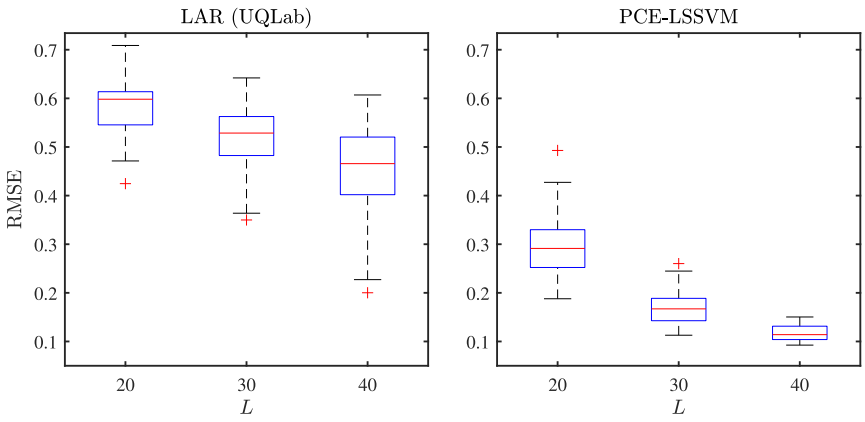

Fig. 14. RMSE obtained with the LAR (left) and PCE-LSSVM (right) methods over 50 independent runs and different training set sizes for the LNA test case.

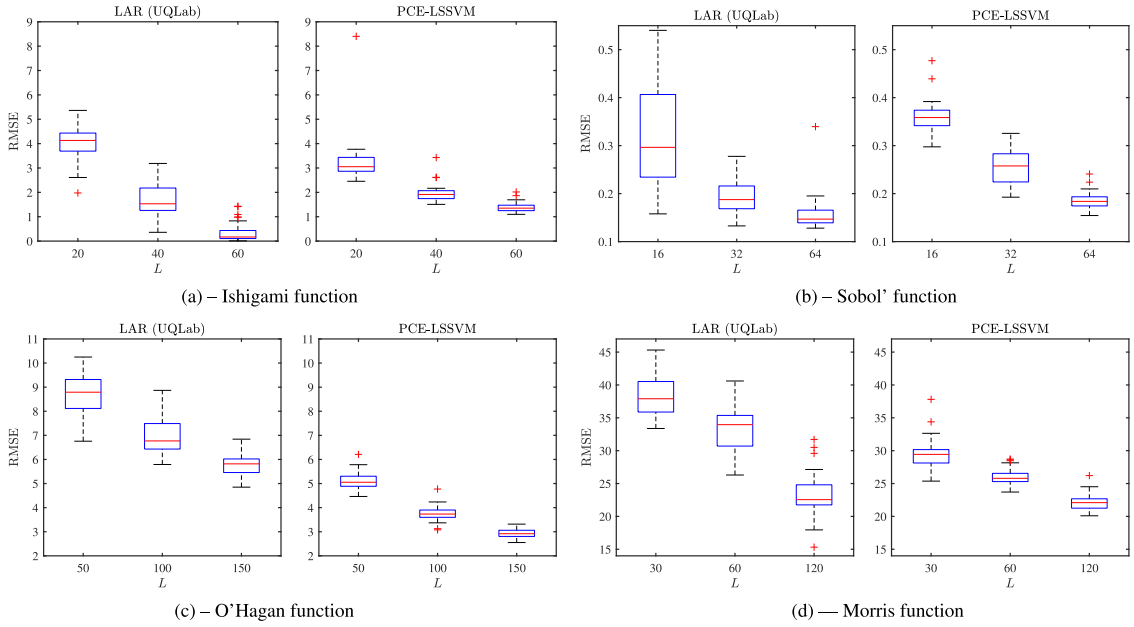

Fig. 15. RMSE obtained with the LAR (left) and PCE-LSSVM (right) methods over 50 independent runs and different training set sizes for the Ishigami (a), Sobol' (b), O'Hagan (c), and Morris (d) benchmark functions.

with $x_i \sim \mathcal{U}(0,1)$, for $i = 1,\dots,20$. The coefficients are defined as follows:

$$\beta_i = \begin{cases} 20 & i \le 10 \\ (-1)^i & i > 10 \end{cases}$$
$$\beta_{ij} = \begin{cases} -15 & i,j \le 6 \\ (-1)^{i+j} & i,j > 6 \end{cases} \tag{63}$$
$$\beta_{ijl} = \begin{cases} -10 & i,j \le 5 \\ 0 & i,j > 5 \end{cases}.$$

We build a third-order Legendre PCE for this function.

8.5. Results

Fig. 15 collects the boxplots describing the RMSE achieved by the LAR and PCE-LSSVM methods over 50 independent runs. For each benchmark, an independent test dataset is considered with $N = 10000$ samples, drawn according a Latin hypercube sampling scheme. The size of the training datasets varies with the test function instead, and the choice is inspired by literature results (Schobi et al., 2015). The comparisons show that for the low-dimensional Ishigami and Sobol' functions, the proposed PCE-LSSVM achieves a lower, or at least less dispersed error with the smaller training datasets. However, the convergence appears to be slower, with the LAR obtaining more accurate predictions for the largest training dataset. For the high-dimensional O'Hagan and Morris functions instead, the PCE-LSSVM method attains a better performance for all training set sizes. For these benchmarks, a similar performance is observed as for the application test cases of Section 7.

Table 8 compares the performance, in terms of training set size and RMSE, of the LAR and PCE-LSSVM methods against alternative state-of-the-art approaches, namely a full Gauss quadrature, a Smolyak' sparse quadrature, and an OLS regression. All the additional methods are implemented via the UQLab toolbox. It should be noted that the two quadrature methods provide a deterministic result, as the sampling points are predefined by the quadrature scheme. Conversely, the OLS, LAR, and PCE-LSSVM techniques use randomly drawn samples and their result is therefore stochastic. Another source of variability in the PCE-LSSVM is the Bayesian optimizer used to tune the hyperparameters. Hence, their RMSE is assessed in terms of both average and standard deviation over 50 runs, the latter indicating the dispersion of the performance across the various training datasets. For the OLS, the minimum required dataset size is considered, corresponding to $L = |\mathcal{K}| + 1$.

The results show that the full Gauss quadrature provides, as expected, the most accurate results. For $d > 4$, the Smolyak' quadrature

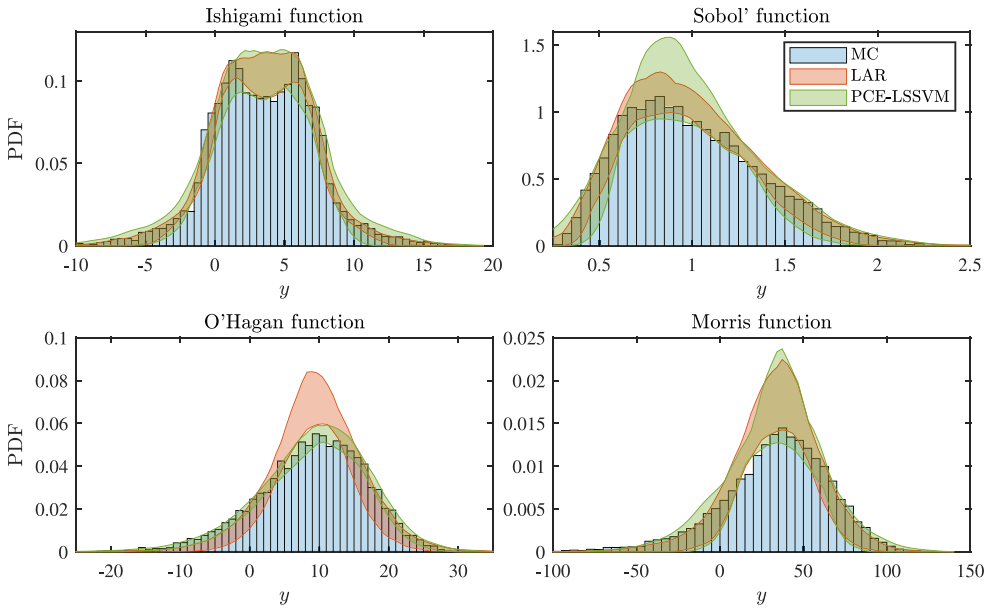

Fig. 16. PDF of the Ishigami (top left), Sobol' (top right), O'Hagan (bottom left), and Morris (bottom right) functions. The blue histograms are the reference distributions. The shaded areas indicate the dispersion of the distributions predicted with the LAR (red) and PCE-LSSVM (green) methods over 50 independent runs.

Table 8
Performance comparison of various PCE methods over the benchmark functions.

Function	Dimensionality	Metric	Full Gauss quadrature	Smolyak' quadrature	OLS	LAR	PCE-LSSVM
Ishigami	3	# training samples	1331	6292	287	20	20
		RMSE (mean)	**0.0065**	0.0287	1.9262	4.0959	3.0964
		RMSE (std)	n/a	n/a	2.4507	0.6719	**0.4323**
Sobol'	8	# training samples	65 536	969	166	16	16
		RMSE (mean)	0.1279	**0.1249**	1.8760	0.2813	0.3669
		RMSE (std)	n/a	n/a	0.9498	0.0900	**0.0552**
O'Hagan	15	# training samples	$\approx 10^9$	OOM	817	50	50
		RMSE (mean)	OOM	OOM	22.1561	8.6353	**5.1110**
		RMSE (std)	n/a	n/a	13.3636	0.8370	**0.3236**
Morris	20	# training samples	$\approx 10^{12}$	OOM	1772	30	30
		RMSE (mean)	OOM	OOM	251.7106	37.4001	**29.2775**
		RMSE (std)	n/a	n/a	229.3047	3.0402	**1.8952**

requires far fewer samples while providing comparable accuracy (cfr. the Sobol' function). Nevertheless, both quadrature methods run out of memory (OOM) for the O'Hagan and Morris functions due to the extremely large number of samples required. It is also noted that the OLS provides larger average error and dispersion for the higher-dimensional benchmarks compared to LAR and PCE-LSSVM, despite the much larger dataset used. For most test cases, the proposed PCE-LSSVM method achieves lower average RMSE and lower dispersion compared to LAR.

For a better visualization of the prediction performance, Fig. 16 compares the distributions predicted with the two methods. The blue histograms are the PDFs of the reference samples. The red and green shaded areas show the dispersion, over the 50 independent runs, of the distributions predicted by the LAR and PCE-LSSVM models, respectively. The results refer to the models trained with the largest dataset, i.e., with $L = 60$ for the Ishigami function, $L = 64$ for the Sobol' function, $L = 150$ for the O'Hagan function, and $L = 120$ for the Morris function. Hence, the area indicates the upper and lower bounds of the predicted distributions. The results are consistent with the performance highlighted in Fig. 15. Indeed, a good accuracy, but also a larger dispersion of the PCE-LSSVM model is observed for the Ishigami and Sobol' functions. The proposed method attains instead a higher accuracy for the O'Hagan and Morris functions. The analysis of this section suggests that the advocated PCE-LSSVM method is competitive with the state-of-the-art LAR method also for these challenging benchmarks.

9. Discussion on accuracy, computational cost, and future research

This section provides an assessment of the proposed PCE-LSSVM method in terms of accuracy and computational cost, and it outlines directions for future research.

The PCE-LSSVM technique was applied on a number of test cases, including an illustrative one-dimensional function, real-life electronic designs with up to 26 uncertain parameters, and standard benchmark functions with up to 20 uncertain inputs. The advocated technique was combined with PCA to efficiently tackle multi-output problems.

For most of the examples, the PCE-LSSVM achieved a very high accuracy even with a very low number of training samples. For the illustrative function (51), the PCE coefficients of a tenth-order expansion were estimated with a good accuracy using five training samples only, and with an excellent accuracy with 20 and 10 training samples for a Gaussian and a uniform input distribution, respectively. Very good results were shown also in terms of RMSE and coefficient of determination R^2 over five independent runs with different sets of randomly drawn training samples. For the electronic designs, the proposed method achieved, for a given training set size, an average RMSE that is up to 7 times smaller and a dispersion that is up to 10 times lower across different training datasets compared to the state-of-the-art LAR method. Specifically, the PCE-LSSVM methods attained a very low error with as few as $L = 10$, $L = 15$, and $L = 40$ training samples for the

Table 9
Summary of the main distinctive features of the proposed PCE-LSSVM method and alternative state-of-the-art approaches.

Feature	Stochastic Galerkin	Full Gauss quadrature	Smolyak' quadrature	OLS	LAR	PCE-LSSVM
Intrusive	yes	no	no	no	no	no
Deterministic	yes	yes	yes	no	no	no
Dispersion	n/a	n/a	n/a	high	moderate	low
# training samples	n/a	very high	high	moderate	small	small

crosstalk scenarios with $d = 6$, $d = 11$, and $d = 26$ uncertain parameters, respectively. This is at least 3 to 45 times smaller than the number of training samples required by alternative approaches based on OLS regression or Gaussian quadratures. Excellent results were reported with $L = 50$ training samples for the slotted waveguide example. This test case was particularly challenging since the response was modeled in a broadband frequency range, in which the scattering parameters exhibited a very large variability. Excellent results were achieved also for the low noise amplifier test case with as few as $L = 40$ training samples. Finally, the proposed method was further benchmarked against standard functions. Compared to LAR, a superior performance was observed especially for the higher-dimensional O'Hagan and Morris functions.

Based on the results reported in this work, Table 9 summarizes the key features of the main state-of-the-art PCE techniques and the proposed PCE-LSSVM method. All methods but the stochastic Galerkin one (not considered in this paper) are data-driven and hence non-intrusive. The OLS, LAR, and PCE-LSSVM are all stochastic, meaning that the achieved result varies depending on the specific choice of the (randomly generated) training dataset. However, as shown by the multiple test cases, the PCE-LSSVM method achieves the lowest dispersion across different training datasets. It was also observed that the quadrature methods require a massive amount of sampling points and quickly run out of memory. The number of training samples required by the OLS regression method is lower-bounded by the total number of PCE coefficients, thereby being still moderately high for high expansion orders and/or large dimensionalities. Both LAR and PCE-LSSVM methods work well with a substantially smaller amount of training data instead, with the latter overall achieving a superior performance in most of the considered test cases.

The good performance of the PCE-LSSVM method can be explained by the fact that it leverages kernels of infinite dimensionality (equivalent to expansions of infinite order), which makes the accuracy virtually infinite for any L^2 function. This contrasts with standard PCEs, whose the accuracy is limited a priori by the predefined expansion order. Moreover, the training is conveniently performed using the solid LSSVM framework, which provides a rigorous calculation of the model coefficients in the dual space using a limited amount of training data. Thanks to the special implicit kernels introduced in Section 5, the PCE coefficients are analytically and inexpensively derived from the dual space coefficients. We conclude that the proposed PCE-LSSVM exploits the most advantageous features of both the PCE and LSSVM methods, i.e., their accuracy as well as the model interpretability in UQ settings of the former and the efficient training of the latter.

Concerning the computational time, the training cost is comparable with standard LSSVM formulations, as in fact we only make use of different kernels. Table 10 summarizes the figures for the proposed test cases. For each example, the configuration that is the most costly for the training, given the PCE order and the number of training samples, is reported. All simulations are performed on a Lenovo Thinkpad X13 Yoga laptop with an Intel(R) Core(TM) i7-10510U, CPU running at 1.8 GHz, and 16 GB of RAM. In all the proposed test cases, the hyperparameters are tuned using a leave-one-out cross-validation strategy, i.e., using as many folds as the number of available training samples L. This becomes less efficient for the largest datasets, since at every iteration L models must be trained (with $L - 1$ samples each). In this case, adopting a strategy with a lower number of folds may be a reasonable tradeoff to improve the training efficiency. The training time is always within a few seconds, except for the test cases of the transient crosstalk and of the microstrip with ground discontinuity. Indeed, these examples require to train multiple models for the principal components in order to capture the entire time- or frequency-domain behavior. The conversion from the dual space to the primal space to retrieve the PCE coefficients is analytical and takes a very limited time even for the largest problems in terms of PCE size (see the last column of Table 10). It should be noted that, for the targeted applications, the training time is often negligible compared to the acquisition of training data. As an example, for the transient crosstalk test case, it took 326 s (5 min 26 s) to simulate the 110 training configurations. On the other hand, the corresponding MC simulation with 5000 samples required 19013 s (5 h 17 min).

At present, implicit kernels were introduced for Hermite and Legendre polynomials only, i.e., for Gaussian and uniform distributions of the inputs. Finding specific kernels for other distribution types is certainly one direction for further research. Moreover, a common limitation of both kernel-based and sparse regression methods such as LAR is that they inherently work with scalar outputs. In order to mitigate this issue, the method was applied in conjunction with PCA to reduce the dimensionality of the output space, rather than training an independent model for each of the original outputs. The combination with an inherent multi-output formulation of the LSSVM (e.g., Xu et al., 2013) is worth to be investigated. Finally, the application of the proposed approach to other kernel-based methods and the improvement of the training strategy are other goals for future works.

10. Conclusions

This paper proposed a novel formulation for computing a PCE model based on the LSSVM method. The approach is based on the introduction of special implicit kernels of infinite dimension, whose feature space basis functions are the Hermite or Legendre polynomials, i.e., the same basis functions as leveraged by the PCE framework for Gaussian and uniform distributions. The training is conveniently performed with a limited number of samples by exploiting the dual space LSSVM formulation. The PCE coefficients up to an arbitrary order are then obtained in post-processing by reverting from the dual space formulation to the primal space one, a step that is made possible by the fact that the feature space functions are known explicitly. Therefore, the training is performed in a nonparametric way, while avoiding forming the complete set of basis functions at once. Moreover, owing to the infinite dimensionality of the kernel, high accuracy is obtained even with a small training set size compared to state-of-the-art techniques, whose maximum accuracy is inherently limited by the expansion order that must be defined a priori.

The advocated approach was first demonstrated based on a simple one-dimensional function. It was then applied to the UQ of electronic designs with up to 26 uncertain parameters and characterized by means of circuit or electromagnetic simulations, as well as to standard benchmark functions with up to 20 inputs. The proposed PCE-LSSVM method achieved an excellent accuracy with a very low amount of training data. A low dispersion of the error across different training datasets was also observed. For most test cases, the advocated approach outperformed a state-of-the-art method based on LAR, whereas alternative PCE-based approaches were not directly comparable owing to the larger number of samples they required.

Plans for future research envisage the extension of the approach to other distribution types, to multi-output formulations, as well as to other kernel-based methods.

Table 10
Summary of the training cost for all the considered test cases.

Test case	PCE order (p)	Dimension (d)	PCE size ($K+1$)	# training samples (L)	Training time	PCE time
Maximum crosstalk	3	6	84	30	7.9 s	< 0.1 s
Transient crosstalk[a]	3	11	364	110	115.8 s	1.7 s
Maximum crosstalk	3	26	3654	80	27.7 s	4.7 s
Microstrip with discontinuity[b]	10	2	66	50	242.5 s	0.4 s
Low noise amplifier	3	25	3276	40	10.9 s	2.5 s
Ishigami function	10	3	286	60	9.3 s	0.1 s
Sobol' function	3	8	165	64	11.1 s	0.1 s
O'Hagan function	3	15	816	150	15.4 s	0.5 s
Morris function	3	20	1771	120	12.1 s	1.6 s

[a] Model for 18 principal components.

[b] Model for 15 complex-valued principal components (a total of 30 models is trained for the real and imaginary parts).

CRediT authorship contribution statement

Paolo Manfredi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Riccardo Trinchero:** Supervision, Resources, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Ahadi, M., Roy, S., 2016. Sparse linear regression (SPLINER) approach for efficient multidimensional uncertainty quantification of high-speed circuits. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 35 (10), 1640–1652. http://dx.doi.org/10.1109/TCAD.2016.2527711.

Anli, F., Gungor, S., 2007. Some useful properties of Legendre polynomials and its applications to neutron transport equation in slab geometry. Appl. Math. Model. 31 (4), 727–733. http://dx.doi.org/10.1016/j.apm.2005.12.005.

Antil, H., Elman, H.C., Onwunta, A., Verma, D., 2023. A deep neural network approach for parameterized PDEs and Bayesian inverse problems. Mach. Learn.: Sci. Technol. http://dx.doi.org/10.1088/2632-2153/ace67c.

Avramova, M.N., Ivanov, K.N., 2010. Verification, validation and uncertainty quantification in multi-physics modeling for nuclear reactor design and safety analysis. Prog. Nucl. Energy 52 (7), 601–614. http://dx.doi.org/10.1016/j.pnucene.2010.03.009.

Bajaj, A., 2023. Performance metrics in machine learning [complete guide]. URL: https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide.

Baptista, R., Stolbunov, V., Nair, P.B., 2019. Some greedy algorithms for sparse polynomial chaos expansions. J. Comput. Phys. 387, 303–325. http://dx.doi.org/10.1016/j.jcp.2019.01.035, URL: https://www.sciencedirect.com/science/article/pii/S0021999119300865.

Bhattacharyya, B., 2022. Uncertainty quantification of dynamical systems by a POD–Kriging surrogate model. J. Comput. Sci. 60, 101602. http://dx.doi.org/10.1016/j.jocs.2022.101602.

Bilionis, I., Zabaras, N., 2012. Multi-output local Gaussian process regression: Applications to uncertainty quantification. J. Comput. Phys. 231 (17), 5718–5746. http://dx.doi.org/10.1016/j.jcp.2012.04.047.

Blatman, G., 2009. Chaos Polynomial Creux Et Adaptatif Pour La Propagation D'incertitudes Et L'analyse De Sensibilité (Ph.D. thesis). Université Blaise Pascal-Clermont-Ferrand II.

Blatman, G., Sudret, B., 2011. Adaptive sparse polynomial chaos expansion based on least angle regression. J. Computational Physics 230 (6), 2345–2367. http://dx.doi.org/10.1016/j.jcp.2010.12.021.

Bürkner, P.-C., Kröker, I., Oladyshkin, S., Nowak, W., 2023. A fully Bayesian sparse polynomial chaos expansion approach with joint priors on the coefficients and global selection of terms. J. Comput. Phys. 488, 112210. http://dx.doi.org/10.1016/j.jcp.2023.112210.

Buss, T., 1996. 2 GHz Low Noise Amplifier With the BFG425W. Technical Report, Philips Semiconductors B.V., Nijmegen, The Netherlands, Appl. Note RNR-T45-96-B-773.

Chahar, R., Mukhopadhyay, T., 2023. Multi-fidelity machine learning based uncertainty quantification of progressive damage in composite laminates through optimal data fusion. Eng. Appl. Artif. Intell. 125, 106647. http://dx.doi.org/10.1016/j.engappai.2023.106647.

Chang, C., Zeng, T., 2023. A hybrid data-driven-physics-constrained Gaussian process regression framework with deep kernel for uncertainty quantification. J. Comput. Phys. 486, 112129. http://dx.doi.org/10.1016/j.jcp.2023.112129.

Cheng, K., Lu, Z., 2018. Adaptive sparse polynomial chaos expansions for global sensitivity analysis based on support vector regression. Comput. Struct. 194, 86–96. http://dx.doi.org/10.1016/j.compstruc.2017.09.002.

Cheng, K., Lu, Z., Zhou, Y., Shi, Y., Wei, Y., 2017. Global sensitivity analysis using support vector regression. Appl. Math. Model. 49, 587–598. http://dx.doi.org/10.1016/j.apm.2017.05.026.

Cui, W., Yan, X., 2009. Adaptive weighted least square support vector machine regression integrated with outlier detection and its application in QSAR. Chemometr. Intell. Lab. Syst. 98 (2), 130–135. http://dx.doi.org/10.1016/j.chemolab.2009.05.008.

Cui, C., Zhang, Z., 2018a. Stochastic collocation with non-Gaussian correlated process variations: Theory, algorithms, and applications. IEEE Trans. Components, Packag. Manuf. Technol. 9 (7), 1362–1375. http://dx.doi.org/10.1109/TCPMT.2018.2889266.

Cui, C., Zhang, Z., 2018b. Uncertainty quantification of electronic and photonic ICs with non-Gaussian correlated process variations. In: Proceedings of the International Conference on Computer-Aided Design. pp. 1–8. http://dx.doi.org/10.1145/3240765.3240860.

Dassault Systemès, 2022. CST studio suite. Release Version 2022.00. URL: https://www.3ds.com/products-services/simulia/products/cst-studio-suite/.

Dietrich, M., Haase, J., 2011. Process Variations and Probabilistic Integrated Circuit Design. Springer Science & Business Media.

Doman, B.G.S., 2015. The Classical Orthogonal Polynomials. World Scientific.

dos Santos, G.S., Luvizotto, L.G.J., Mariani, V.C., dos Santos Coelho, L., 2012. Least squares support vector machines with tuning based on chaotic differential evolution approach applied to the identification of a thermal process. Expert Syst. Appl. 39 (5), 4805–4812. http://dx.doi.org/10.1016/j.eswa.2011.09.137.

Freni, G., Mannina, G., 2010. Bayesian approach for uncertainty quantification in water quality modelling: The influence of prior distribution. J. Hydrol. 392 (1–2), 31–39. http://dx.doi.org/10.1016/j.jhydrol.2010.07.043.

García-Merino, J., Calvo-Jurado, C., Martínez-Pañeda, E., García-Macías, E., 2023. Multielement polynomial chaos Kriging-based metamodelling for Bayesian inference of non-smooth systems. Appl. Math. Model. 116, 510–531. http://dx.doi.org/10.1016/j.apm.2022.11.039.

Garg, S., Chakraborty, S., 2023. VB-DeepONet: A Bayesian operator learning framework for uncertainty quantification. Eng. Appl. Artif. Intell. 118, 105685. http://dx.doi.org/10.1016/j.engappai.2022.105685.

Garnett, R., 2023. Bayesian Optimization. Cambridge University Press.

Ghanem, R.G., Spanos, P.D., 1991. Stochastic Finite Elements: A Spectral Approach. Springer-Verlag, Berlin, Heidelberg.

Gradshteyn, I.S., Ryzhik, I.M., 2014. Table of Integrals, Series, and Products. Academic Press.

Hadigol, M., Doostan, A., 2018. Least squares polynomial chaos expansion: A review of sampling strategies. Comput. Methods Appl. Mech. Engrg. 332, 382–407. http://dx.doi.org/10.1016/j.cma.2017.12.019.

He, J., Mattis, S.A., Butler, T.D., Dawson, C.N., 2019. Data-driven uncertainty quantification for predictive flow and transport modeling using support vector machines. Comput. Geosci. 23, 631–645. http://dx.doi.org/10.1007/s10596-018-9762-4.

Ishigami, T., Homma, T., 1990. An importance quantification technique in uncertainty analysis for computer models. In: [1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis. IEEE, pp. 398–403. http://dx.doi.org/10.1109/ISUMA.1990.151285.

Kaintura, A., Dhaene, T., Spina, D., 2018. Review of polynomial chaos-based methods for uncertainty quantification in modern integrated circuits. Electronics 7 (3), 30. http://dx.doi.org/10.3390/electronics7030030.

Kantarakias, K.D., Papadakis, G., 2023. Sensitivity-enhanced generalized polynomial chaos for efficient uncertainty quantification. J. Comput. Phys. 112377. http://dx.doi.org/10.1016/j.jcp.2023.112377.

Kaytez, F., Taplamacioglu, M.C., Cam, E., Hardalac, F., 2015. Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. Int. J. Electr. Power Energy Syst. 67, 431–438. http://dx.doi.org/10.1016/j.ijepes.2014.12.036.

Kibble, W., 1945. An extension of a theorem of Mehler's on Hermite polynomials. In: Mathematical Proceedings of the Cambridge Philosophical Society. Vol. 41, (1), Cambridge University Press, pp. 12–15. http://dx.doi.org/10.1017/S0305004100022313.

Kim, H., Cheon, H., Ahn, Y.-H., Choi, D.G., 2019. Uncertainty quantification and scenario generation of future solar photovoltaic price for use in energy system models. Energy 168, 370–379. http://dx.doi.org/10.1016/j.energy.2018.11.075.

Lee, D., Rahman, S., 2023. High-dimensional stochastic design optimization under dependent random variables by a dimensionally decomposed generalized polynomial chaos expansion. Int. J. Uncertain. Quantif. 13 (4), http://dx.doi.org/10.1615/Int.J.UncertaintyQuantification.2023043457.

Li, M., Jia, G., Mahmoud, H., Yu, Y.-H., Tom, N., 2023. Physics-constrained Gaussian process model for prediction of hydrodynamic interactions between wave energy converters in an array. Appl. Math. Model. 119, 465–485. http://dx.doi.org/10.1016/j.apm.2023.03.003.

Liu, J., Jiang, C., 2023. Surrogate modeling for high dimensional uncertainty propagation via deep kernel polynomial chaos expansion. Appl. Math. Model. http://dx.doi.org/10.1016/j.apm.2023.05.036.

Lorenzoni, F., Casarin, F., Caldon, M., Islami, K., Modena, C., 2016. Uncertainty quantification in structural health monitoring: Applications on cultural heritage buildings. Mech. Syst. Signal Process. 66, 268–281. http://dx.doi.org/10.1016/j.ymssp.2015.04.032.

Lüthen, N., Marelli, S., Sudret, B., 2021. Sparse polynomial chaos expansions: Literature survey and benchmark. SIAM/ASA J. Uncertain. Quant. 9 (2), 593–649. http://dx.doi.org/10.1137/20M1315774.

Manfredi, P., Grivet-Talocia, S., 2021. Fast stochastic surrogate modeling via rational polynomial chaos expansions and principal component analysis. IEEE Access 9, 102732–102745. http://dx.doi.org/10.1109/ACCESS.2021.3097543.

Manfredi, P., Trinchero, R., 2020. A data compression strategy for the efficient uncertainty quantification of time-domain circuit responses. IEEE Access 8, 92019–92027. http://dx.doi.org/10.1109/ACCESS.2020.2993338.

Marelli, S., Lüthen, N., Sudret, B., 2022. UQLab User Manual – Polynomial Chaos Expansions. Technical Report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, Report UQLab-V2.0-104.

Mehler, F.G., 1866. Ueber Die Entwicklung Einer Function Von Beliebig Vielen Variablen Nach Laplaceschen Functionen Höherer Ordnung. Walter de Gruyter, Berlin, NY, http://dx.doi.org/10.1515/crll.1866.66.161.

Močkus, J., 1989. Bayesian Approach to Global Optimization. Kluwer Academic, Dordrecht.

Moghaddam, V.H., Hamidzadeh, J., 2016. New Hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier. Pattern Recognit. 60, 921–935. http://dx.doi.org/10.1016/j.patcog.2016.07.004.

Mohri, M., Rostamizadeh, A., Talwalkar, A., 2018. Foundations of Machine Learning. MIT Press.

Moosavi, A., Rao, V., Sandu, A., 2021. Machine learning based algorithms for uncertainty quantification in numerical weather prediction models. J. Comput. Sci. 50, 101295. http://dx.doi.org/10.1016/j.jocs.2020.101295.

Morris, M.D., 1991. Factorial sampling plans for preliminary computational experiments. Technometrics 33 (2), 161–174. http://dx.doi.org/10.1080/00401706.1991.10484804.

Najm, H.N., 2009. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. Annu. Rev. Fluid Mech. 41, 35–52. http://dx.doi.org/10.1146/annurev.fluid.010908.165248.

N.V., N.S., 2012. 36BFG425W SPICE model. URL: https://www.nxp.com/downloads/en/spice-model/BFG425W_SPICE.prm.

Oakley, J.E., O'Hagan, A., 2004. Probabilistic sensitivity analysis of complex models: a Bayesian approach. J. R. Stat. Soc. Ser. B Stat. Methodol. 66 (3), 751–769. http://dx.doi.org/10.1111/j.1467-9868.2004.05304.x.

Pang, Z., O'Neill, Z., 2018. Uncertainty quantification and sensitivity analysis of the domestic hot water usage in hotels. Appl. Energy 232, 424–442. http://dx.doi.org/10.1016/j.apenergy.2018.09.221.

Peng, X., Kou, J., Zhang, W., 2023. Multi-fidelity nonlinear unsteady aerodynamic modeling and uncertainty estimation based on Hierarchical Kriging. Appl. Math. Model. 122, 1–21. http://dx.doi.org/10.1016/j.apm.2023.05.031.

Prudnikov, A.P., Brychkov, Y.A., Marichev, O.I., 1990. Integrals and Series, Vol. 2: Special Functions. Gordon and Breach, New York.

Psaros, A.F., Meng, X., Zou, Z., Guo, L., Karniadakis, G.E., 2023. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. J. Comput. Phys. 477, 111902. http://dx.doi.org/10.1016/j.jcp.2022.111902.

Pulch, R., Youssef, M., 2020. Machine learning for trajectories of parametric nonlinear dynamical systems. J. Mach. Learn. Model. Comput. 1 (1), 75–95. http://dx.doi.org/10.1615/JMachLearnModelComput.2020034093.

Qi, D., Harlim, J., 2023. A data-driven statistical-stochastic surrogate modeling strategy for complex nonlinear non-stationary dynamics. J. Comput. Phys. 485, 112085. http://dx.doi.org/10.1016/j.jcp.2023.112085.

Schobi, R., Sudret, B., Wiart, J., 2015. Polynomial-chaos-based Kriging. Int. J. Uncertain. Quantif. 5 (2), http://dx.doi.org/10.1615/Int.J.UncertaintyQuantification.2015012467.

Shin, H., Choi, M., 2023. Physics-informed variational inference for uncertainty quantification of stochastic differential equations. J. Comput. Phys. 487, 112183. http://dx.doi.org/10.1016/j.jcp.2023.112183.

Soboĺ, I., 1993. Sensitivity estimates for nonlinear mathematical models. Math. Model. Comput. Exp. 1, 407.

Srivastav, A., Tewari, A., Dong, B., 2013. Baseline building energy modeling and localized uncertainty quantification using Gaussian mixture models. Energy Build. 65, 438–447. http://dx.doi.org/10.1016/j.enbuild.2013.05.037.

Sudret, B., 2008. Global sensitivity analysis using polynomial chaos expansions. Reliab. Eng. Syst. Saf. 93 (7), 964–979. http://dx.doi.org/10.1016/j.ress.2007.04.002.

Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J., 2002. Least Squares Support Vector Machines. World Scientific Pub. Co., Singapore, http://dx.doi.org/10.1142/5089.

Synopsis Inc., 2009. HSPICE. Release Version B-2008.09. URL: https://www.synopsys.com/implementation-and-signoff/ams-simulation/primesim-hspice.html.

Trinchero, R., Larbi, M., Torun, H.M., Canavero, F.G., Swaminathan, M., 2018. Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters. IEEE Access 7, 4056–4066. http://dx.doi.org/10.1109/ACCESS.2018.2888903.

Vapnik, V.N., 1998. Statistical Learning Theory. Wiley-Interscience, New York, NY.

Vauchel, N., Garnier, É., Gomez, T., 2023. A multi-element non-intrusive Polynomial Chaos method using agglomerative clustering based on the derivatives to study irregular and discontinuous Quantities of Interest. J. Comput. Phys. 473, 111763. http://dx.doi.org/10.1016/j.jcp.2022.111763.

Vladimirescu, A., 1994. The SPICE Book. Wiley, New York.

Weng, T.-W., Zhang, Z., Su, Z., Marzouk, Y., Melloni, A., Daniel, L., 2015. Uncertainty quantification of silicon photonic devices with correlated and non-Gaussian random parameters. Opt. Express 23 (4), 4242–4254. http://dx.doi.org/10.1364/OE.23.004242.

Xiu, D., 2007. Efficient collocational approach for parametric uncertainty analysis. Commun. Comput. Phys. 2 (2), 293–309.

Xiu, D., 2009. Fast numerical methods for stochastic computations: A review. Commun. Comput. Phys. 5 (2–4), 242–272.

Xiu, D., Hesthaven, J.S., 2005. High-order collocation methods for differential equations with random inputs. SIAM J. Sci. Comput. 27 (3), 1118–1139. http://dx.doi.org/10.1137/040615201.

Xiu, D., Karniadakis, G.E., 2002. The Wiener–Askey polynomial chaos for stochastic differential equations. SIAM J. Sci. Comput. 24 (2), 619–644. http://dx.doi.org/10.1137/S1064827501387826.

Xu, S., An, X., Qiao, X., Zhu, L., Li, L., 2013. Multi-output least-squares support vector regression machines. Pattern Recognit. Lett. 34 (9), 1078–1084. http://dx.doi.org/10.1016/j.patrec.2013.01.015.

Yang, H., Fujii, Y., Wang, K.-W., Gorodetsky, A.A., 2023. Control variate polynomial chaos: Optimal fusion of sampling and surrogates for multifidelity uncertainty quantification. Int. J. Uncertain. Quantif. 13 (3), http://dx.doi.org/10.1615/Int.J.UncertaintyQuantification.2022043638.

Zhang, B.-Y., Ni, Y.-Q., 2023. A novel sparse polynomial chaos expansion technique with high adaptiveness for surrogate modelling. Appl. Math. Model. 121, 562–585. http://dx.doi.org/10.1016/j.apm.2023.05.005.

Zhang, C., Shafieezadeh, A., 2023. Nested physics-informed neural network for analysis of transient flows in natural gas pipelines. Eng. Appl. Artif. Intell. 122, 106073. http://dx.doi.org/10.1016/j.engappai.2023.106073.

Zhu, Y., Tang, Y.-H., Kim, C., 2023. Learning stochastic dynamics with statistics-informed neural network. J. Comput. Phys. 474, 111819. http://dx.doi.org/10.1016/j.jcp.2022.111819.