# Probabilistic learning for modeling and quantifying model-form uncertainties in nonlinear computational mechanics

Christian Soize, Charbel Farhat

PREPRINT- October 2019
Published online in Int. J. Numer. Meth. Engng, doi: 10.1002/nme.5980

# Probabilistic learning for modeling and quantifying model-form uncertainties in nonlinear computational mechanics

C. Soize[1]*and C. Farhat[2]

[1] *Laboratoire Modélisation et Simulation Multi Echelle, MSME UMR 8208 CNRS, Université Paris-Est, 5 bd Descartes, 77454 Marne-la-Vallee, France*
[2] *Department of Aeronautics and Astronautics, Department of Mechanical Engineering, and Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305-4035, USA*

SUMMARY

Recently, a novel, nonparametric, probabilistic method for modeling and quantifying model-form uncertainties in nonlinear computational mechanics was proposed. Its potential was demonstrated through several uncertainty quantification (UQ) applications in vibration analysis and nonlinear computational structural dynamics. This method, which relies on projection-based model order reduction in order to achieve computational feasibility, exhibits a vector-valued hyperparameter in the probability model of the random reduced-order basis and associated stochastic, projection-based reduced-order model. It identifies this hyperparameter by formulating a statistical inverse problem grounded in target quantities of interest and solving the corresponding nonconvex optimization problem. For many practical applications however, this identification approach is computationally intensive. For this reason, this paper presents a faster, predictor-corrector approach for determining the appropriate value of the vector-valued hyperparameter that is based on a probabilistic learning on manifolds. It also demonstrates the computational advantages of this alternative identification approach through the UQ of two three-dimensional, nonlinear, structural dynamics problems associated with two different configurations of a MEMS device.

Received ...

KEY WORDS: probabilistic learning, model-form uncertainties, nonparametric probabilistic method, model reduction, uncertainty quantification, machine learning

*Notation, nomenclature and acronyms*

Throughout this paper:

A real, deterministic variable is denoted by a lower case letter such as $y$.
A real, deterministic vector is denoted by a boldface, lower case letter such as in $\mathbf{y} = (y_1, \ldots, y_N)$.
A real, random variable is denoted by an upper case letter such as $Y$.
A real, random vector is denoted by a boldface, upper case letter such as in $\mathbf{Y} = (Y_1, \ldots, Y_N)$.
A real, deterministic matrix is denoted by an upper (or lower) case letter between brackets such as $[A]$ (or $[a]$).
A real, random matrix is denoted by a boldface, upper case letter between brackets such as $[\mathbf{A}]$.

$\|\mathbf{y}\|$ denotes the Euclidean norm of vector $\mathbf{y}$.
$E$ denotes the mathematical expectation.
$\mathbb{M}_{N,n}$ denotes the set of $N \times n$ real matrices.
$\mathbb{M}_n$ denotes the set of square $n \times n$ real matrices.

*Correspondence to: Laboratoire Modélisation et Simulation Multi Echelle, MSME UMR 8208 CNRS, Université Paris-Est, 5 bd Descartes, 77454 Marne-la-Vallee, France. E-mail: christian.soize@univ-paris-est.fr

$\mathbb{M}_N^+$ denotes the set of real, symmetric, positive-definite $N \times N$ matrices.
$\mathbb{M}_N^{+0}$ denotes the set of real, symmetric, positive $N \times N$ matrices.
$\mathbb{M}_n^{\mathrm{u}}$ denotes the set of real, upper triangular $n \times n$ matrices with positive or zero diagonal entries.
$\mathbb{R}$ denotes the set of all real numbers.
$\mathbb{C}$ denotes the set of all complex numbers.
$\mathbb{R}^N$ denotes the Euclidean space of vectors $\mathbf{y} = (y_1, \ldots, y_N)$.
$\mathbb{C}^N$ denotes the Hermitian space of vectors $\widehat{\mathbf{y}} = (\widehat{y}_1, \ldots, \widehat{y}_N)$.
$A_{jk}$ denotes the entry $[A]_{jk}$ of matrix $[A]$.
$[A]^T$ denotes the transpose of matrix $[A]$.
$[I_n]$ denotes the identity matrix in $\mathbb{M}_n$.
$[0_{N,n}]$ denotes the zero matrix in $\mathbb{M}_{N,n}$.
$\delta_{jk}$ denotes the Kronecker symbol: $\delta_{jk} = 0$ if $j \neq k$ and $\delta_{jk} = 1$ if $j = k$.
$t$ denotes time $i$ denotes the pure imaginary complex number satisfying $i^2 = -1$.

3D stands for three-dimensional.
dof stands for degree of freedom.
FE stands for finite element.
HDM stands for high-dimensional computational model.
MEMS stands for microelectromechanical systems.
PDF stands for probability density function.
PLM stands for probabilistic learning on manifolds.
POD stands for proper orthogonal decomposition.
PROM stands for projection-based reduced-order model.
QA stands for quality assessment.
QoI stands for quantity of interest.
ROB stands for reduced-order basis.
SPROM stands for stochastic projection-based reduced-order model.
SROB stands for stochastic reduced-order basis.
SVD stands for singular value decomposition.
UQ stands for uncertainty quantification.

## 1. INTRODUCTION

In general, once a computational model must be considered for any purpose, model-form uncertainties become unavoidable. They can result either from the lack of knowledge of the true physics underlying the problem of interest, or the omission or truncation of modeling details. Their raisons d'être range from the lack of available information, to the inability to discern in some circumstances between important and unimportant modeling details. Such uncertainties affect the ability of even an HDM to deliver predictive results – that is, results that match reasonably well their experimental counterparts (which themselves can also be tainted by other types of uncertainties). Examples in computational structural dynamics include constitutive, multiscale, friction, homogenization, and free-play modeling errors. A parametric PROM constructed for accelerating parametric studies or stochastic computations meant to be carried out using an HDM inherits all model-form uncertainties associated with this HDM, because it is obtained by projecting this same HDM onto a carefully computed ROB. Such a reduced-order model is also tainted by additional uncertainties due to, among other factors: the finite sampling used during the offline training of this PROM where the ROB is computed (for example, see [1]); the truncation of the subspace of approximation represented by this ROB using POD or SVD (for example, see [2]); and the approximation errors induced by hyperreduction when this process is used to accelerate the projections incurred by the construction of this PROM (for example, see [3]).

In [4], a nonparametric, probabilistic method was developed to model and quantify model-form uncertainties in deterministic computational models. This method recognizes that data,

whether experimental or numerical and high-dimensional, may contain fundamental information or knowledge that is not captured by deterministic computational models. Therefore, it exploits available data to adapt the *subspace* in which the solution of the problem formulated using the computational model is searched. The method is innovative as it is a significant departure from the dominant parametric approach to UQ where model parameters are adapted instead, in order to discover some form of the information or knowledge encapsulated in data. To this end, the nonparametric, probabilistic method developed in [4] parameterizes the solution subspace by randomizing its representative basis using a vector-valued hyperparameter. This transforms the deterministic computational model into a hyperparametric stochastic one. In order to determine the hyperparameter, the method constructs and solves an inverse statistical problem designed to minimize the discrepancies between the mean values and statistical fluctuations of the QoIs predicted using the hyperparametric stochastic model, and target values obtained from the available data. If the computational model is a deterministic HDM, the method implements all of the above ideas into an SPROM in order to make them computationally feasible. It follows that the method developed in [4] for modeling and quantifying model-form uncertainties essentially extracts the information missing in a given deterministic computational model from data, and infuses this information into a stochastic, lower-dimensional counterpart. For this reason, this method has also been described in [5] as a data-driven, probabilistic, model adaptation method.

As mentioned above, a PROM constructed using a Galerkin [7] or Petrov-Galerkin [8] projection of an HDM inherits all uncertainties present in this HDM. For this reason, and because it is low-dimensional, such a reduced-order model can be used to effectively model and quantify the uncertainties of the HDM of interest. More importantly however – and unlike most if not all other UQ methods proposed in the literature that mention or rely on model reduction for accelerating the stochastic computations they incur – the nonparametric, probabilistic method developed in [4] accounts for the modeling errors introduced by model reduction to construct its underlying SPROM. If experimental data are not available, it leads to the modeling and quantification of the modeling errors due to model reduction only. Otherwise, it leads to the modeling and quantification of the modeling errors of the given HDM, while accounting for those introduced during the process by model reduction.

The potential of this nonparametric, probabilistic method for the UQ of model-form uncertainties in nonlinear structural dynamics problems was demonstrated in [4]. The method was extended to model-form uncertainties arising in generalized eigenvalue computations in [5], where its performance for such applications was demonstrated through the vibration analysis of two different real-world aircaft for which experimental data is available [9]. In [10], the computational aspects of this method were refined by hyperreducing the SPROM to accelerate its execution time. The potential of the resulting enhanced approach for quantifying model-form uncertainties in eigenvalue computations, while accounting for modeling errors due to hyperreduction, was demonstrated through what-if vibration analysis scenarios. These were associated with shape changes of a supersonic jet engine nozzle with inner walls, outer walls, stringers, and baffles [10].

Given an HDM of interest of dimension $N$, the family of nonparametric, probabilistic methods for UQ described in [4], [5], and [10] consists in substituting the ROB used for constructing a nonlinear PROM of dimension $n \ll N$ by an adapted SROB, which leads to the construction of a nonlinear SPROM of dimension $n$. The probability model of the SROB depends on the aforementioned vector-valued hyperparameter, $\boldsymbol{\alpha}$, which belongs to an admissible set $\mathcal{C}_{\boldsymbol{\alpha}} \subset \mathbb{R}^{m_{\boldsymbol{\alpha}}}$, where $m_{\boldsymbol{\alpha}} = 2 + n(n+1)/2$. This family of UQ methods performs the identification of $\boldsymbol{\alpha}$ by solving a nonconvex optimization problem formulated using an objective (cost) function related to a certain distance between some given targets for the QoIs, and the corresponding predictions carried out using the nonlinear SPROM. Again, the target values for the QoIs can be extracted from experimental data when these are available, or corresponding HDM predictions when experimental data are not available. If $n$ is greater than a few units, $m_\alpha = n(n+1)/2$ is sufficiently large to challenge the fast solution of this optimization problem and therefore to compromise the execution in near-real-time of this family of nonparametric, probabilistic UQ methods. To this end, this paper presents an alternative, more economical approach for identifying the vector-valued hyperparameter

$\boldsymbol{\alpha}$ for practical values of the low-dimension $n$ of the SPROM. First, this approach constructs a low-dimensional predictor $\boldsymbol{\alpha}^0$ for $\boldsymbol{\alpha}$ of dimension $m_{\boldsymbol{\alpha}}^0 \ll 2 + n(n+1)/2$ whose computational cost is relatively small. If it judges the predictor to be insufficiently accurate, it constructs next a corrector $\boldsymbol{\alpha}^{\text{opt}}$ of the same dimension using a probabilistic learning on manifolds (PLM) based on ideas that are similar to those recently described in [11, 12] and demonstrated in [13]. By avoiding to call a large number of times the nonlinear SPROM, this alternative approach achieves a lower computational complexity than the random search optimizer originally proposed in [4] for identifying the hyperparameter $\boldsymbol{\alpha}$.

The remainder of this paper is organized as follows. Section 2 sets the context of this paper to that of nonlinear computational structural dynamics problems in the time domain. It also overviews the nonparametric probabilistic method proposed in [4] for modeling and quantifying model-form uncertainties for two purposes: to keep this paper as self-contained as possible; and more importantly, to highlight the presence, role, and significance of the hyperparameter $\boldsymbol{\alpha}$ of this method. Section 3 briefly recalls the nonconvex optimization problem formulated in [4] for determining this vector-valued hyperparameter and outlines the potential difficulties associated with the solution of this problem using a conventional approach. Section 4 introduces the main contribution of this paper. It starts by describing the construction of a low-dimensional predictor $\boldsymbol{\alpha}^0$ for $\boldsymbol{\alpha}$ that is based on the Fourier transform of the stochastic solution of the nonlinear problem of interest. This stochastic solution is computed in near-real-time using the SPROM. Next, Section 4 describes the construction of the proposed corrector based on the aforementioned probabilistic learning on manifolds. Section 5 discusses the performance of the nonparametric probabilistic method for UQ equipped with the proposed predictor-corrector approach for two 3D, nonlinear, structural dynamics problems associated with two different configurations of a MEMS device. Finally, Section 6 concludes this paper.

*REMARK 1.* PROMs are usually designed to accelerate the repeated processing of parametric HDMs known as $\boldsymbol{\mu}$-HDMs. In this context, $\boldsymbol{\mu}$ denotes a vector-valued parameter representing, for example, a parameteric shape, material property, boundary condition, or source term. Consequently, the PROMs and SPROMs mentioned above have been referred to in [4, 5, 10] as $\boldsymbol{\mu}$-PROMs and $\boldsymbol{\mu}$-SPROMs, respectively. Here, it is noted however that the vector-valued parameter $\boldsymbol{\mu}$ is unrelated to the vector-valued hyperparameter $\boldsymbol{\alpha}$ discussed above, and that the contribution of this paper is independent from and agnostic about how the parametric aspect of a $\boldsymbol{\mu}$-PROM is treated. The problem discussed in this paper and the proposed solution to this problem are also independent of the concept of hyperreduction and agnostic about how such an approximation is performed. For these reasons, and in order to avoid any unnecessary distractions, the contribution of this paper is presented without any reference to, or discussion of, the parametric and hyperreduction aspects of a PROM or SPROM. For a demonstration of how all these concepts perform well together, the reader is referred to [4, 5, 10].

## 2. NONPARAMETRIC PROBABILISTIC METHOD FOR MODEL-FORM UNCERTAINTIES IN NONLINEAR COMPUTATIONAL MECHANICS

In principle, the family of nonparametric, probabilistic methods for UQ described in [4], [5], and [10] can be extended to any linear or nonlinear computational problem for which a PROM can be constructed. However, the details associated with any extension can be problem dependent. For this reason, and for the sake of clarity and simplicity, the focus of this entire paper is set here on nonlinear transient dynamics problems.

### *2.1. Nonlinear high-dimensional computational model*

After semi-discretization by the FE method, a large class of nonlinear transient dynamics problems can be written as

$$[M]\,\ddot{\mathbf{y}}(t) + \mathbf{g}(\mathbf{y}(t), \dot{\mathbf{y}}(t)) = \mathbf{f}(t) \quad , \quad t \in ]t_0, T]\,, \tag{2.1}$$

where $\mathbf{y}(t) = (y_1(t), \ldots, y_N(t))$ is the $\mathbb{R}^N$ vector of the $N$ displacement dofs before the application of the boundary conditions, $\dot{\mathbf{y}}(t) = d\mathbf{y}(t)/dt$ and $\ddot{\mathbf{y}}(t) = d^2\mathbf{y}(t)/dt^2$ are the corresponding velocity and acceleration vectors, $[M]$ is the FE mass matrix belonging to $\mathbb{M}_N^+$ and is assumed here to be independent of $t$, $\mathbf{g}(\mathbf{y}(t), \dot{\mathbf{y}}(t))$ is the $\mathbb{R}^N$ vector of internal forces at time $t$ and is assumed here to depend on $\mathbf{y}(t)$ and $\dot{\mathbf{y}}(t)$, $\mathbf{f}(t)$ is the $\mathbb{R}^N$ vector of external forces at time $t$, and finally, $t_0$ and $T$ define the time interval of interest and satisfy $-\infty < t_0 < T < +\infty$.

The above equation defines an HDM whose initial conditions can be written as

$$\mathbf{y}(t_0) = \mathbf{y}_0 \quad , \quad \dot{\mathbf{y}}(t_0) = \mathbf{y}_1 \, , \tag{2.2}$$

and whose boundary conditions and any other governing linear constraints can be written altogether as

$$[B]^T \mathbf{y}(t) = \mathbf{0}_{N_{\text{CD}}} \quad , \quad t \in [t_0, T] \, . \tag{2.3}$$

Here, $[B]$ is a time independent matrix in $\mathbb{M}_{N,N_{\text{CD}}}$ satisfying $[B]^T[B] = [I_{N_{\text{CD}}}]$, $N_{CD} < N$, and $\mathbf{y}_0$ and $\mathbf{y}_1$ are two given vectors in $\mathbb{R}^N$ that satisfy the constraint (2.3).

The QoIs (also called observations) at time $t$ associated with the above HDM are represented by the vector $\mathbf{o}(t) = (o_1(t), \ldots, o_{m_\mathbf{o}}(t))$ with values in $\mathbb{R}^{m_\mathbf{o}}$. This vector is written here as

$$\mathbf{o}(t) = \mathbf{h}_{\text{time}}(\mathbf{y}(t)) \quad , \quad t \in [t_0, T] \, , \tag{2.4}$$

where $\mathbf{h}_{\text{time}}$ is a given mapping with values in $\mathbb{R}^{m_\mathbf{o}}$ (for example, $\mathbf{h}_{\text{time}}(\mathbf{y}(t)) = [\mathbb{O}]\, d^2\mathbf{y}(t)/dt^2$, where $[\mathbb{O}]$ is a given constant matrix in $\mathbb{M}_{m_\mathbf{o},N}$).

### *2.2. Construction of a nonlinear projection-based reduced-order model*

Let $[V] \in \mathbb{M}_{N,n}$ be a ROB of dimension $n \ll N$ constructed for approximating the solution $\{\mathbf{y}(t), t \in [t_0, T]\}$, and satisfying the constraint equation

$$[B]^T[V] = [0_{N_{\text{CD}},n}] \, . \tag{2.5}$$

In [4], $[V]$ was furthermore constructed to satisfy the orthonormality condition $[V]^T[M]\,[V] = [I_n]$. In this work, $[V]$ is constructed to satisfy instead the orthonormality condition

$$[V]^T[V] = [I_n] \, . \tag{2.6}$$

The Galerkin projection of the HDM defined in Equations (2.1) to (2.4) onto the ROB $[V]$ yields the following nonlinear PROM

$$\mathbf{y}^{(n)}(t) = [V]\,\mathbf{q}(t) \quad , \quad t \in [t_0, T] \, , \tag{2.7}$$

$$[V]^T[M]\,[V]\,\ddot{\mathbf{q}}(t) + [V]^T\mathbf{g}([V]\,\mathbf{q}(t), [V]\,\dot{\mathbf{q}}(t)) = [V]^T\mathbf{f}(t) \quad , \quad t \in ]t_0, T] \, , \tag{2.8}$$

whose initial conditions are

$$\mathbf{q}(t_0) = [V]^T\mathbf{y}_0 \quad , \quad \dot{\mathbf{q}}(t_0) = [V]^T\mathbf{y}_1 \, , \tag{2.9}$$

where $\{\mathbf{y}^{(n)}(t), t \in [t_0, T]\}$ is the $n$-order approximation of $\{\mathbf{y}(t), t \in [t_0, T]\}$. Concerning the QoIs, the corresponding approximation $\mathbf{o}^{(n)}$ of $\mathbf{o}$ is given by

$$\mathbf{o}^{(n)}(t) = \mathbf{h}_{\text{time}}(\mathbf{y}^{(n)}(t)) \quad , \quad t \in [t_0, T] \, . \tag{2.10}$$

For a given $n \ll N$, the approximation error associated with the prediction of the QoIs using the nonlinear PROM instead of the nonlinear HDM can be estimated *a priori* as

$$\varepsilon(n) = \int_{t_0}^{T} \|\mathbf{o}(t) - \mathbf{o}^{(n)}(t)\|^2 \, dt \, .$$

### 2.3. Construction of a nonlinear stochastic projection-based reduced-order model

The details of the construction of the SPROM associated with the HDM defined in Equations (2.1) to (2.4) can be found in [4]. Here, a short summary of this construction is adapted to the othonormalization condition (2.6) of the underlying ROB.

The nonparametric probabilistic method for the modeling and quantification of model-form uncertainties described here consists in substituting the deterministic ROB $[V]$ by a stochastic counterpart $[\mathbf{W}]$, referred to as the SROB. In view of the constraint Equation (2.5) and the orthonormality condition (2.6), this SROB must verify the following properties:

- $[\mathbf{W}] : \theta \mapsto [\mathbf{W}(\theta)]$ is a random variable (random matrix), defined on a probability space $(\Theta, \mathcal{T}, \mathcal{P})$, with values in $\mathbb{M}_{N,n}$.
- The support of its probability distribution, $P_{[\mathbf{W}]}$, is in the subset of $\mathbb{M}_{N,n}$ defined by the following constraints which are verified almost surely

$$[B]^T[\mathbf{W}] = [0_{N_{\text{CD}},n}]\,, \tag{2.11}$$

$$[\mathbf{W}]^T[\mathbf{W}] = [I_n]\,. \tag{2.12}$$

- The probability distribution $P_{[\mathbf{W}]}$ of $[\mathbf{W}]$ depends on a vector-valued hyperparameter $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{m_{\boldsymbol{\alpha}}})$ belonging to a subset $\mathcal{C}_{\boldsymbol{\alpha}}$ of $\mathbb{R}^{m_{\boldsymbol{\alpha}}}$.

The construction of such an SROB is summarized in Section 2.4. The nonlinear SPROM associated with the nonlinear PROM described by Equations (2.7) to (2.10) is obtained by substituting $[V]$ with the random matrix $[\mathbf{W}]$. Consequently, $\mathbf{y}^{(n)}$, $\mathbf{q}$, and $\mathbf{o}^{(n)}$ are transformed into the stochastic processes $\mathbf{Y}^{(n)}$, $\mathbf{Q}$, and $\mathbf{O}^{(n)}$, and the nonlinear SPROM is given by

$$\mathbf{Y}^{(n)}(t) = [\mathbf{W}]\,\mathbf{Q}(t) \quad , \quad t \in [t_0, T]\,, \tag{2.13}$$

$$[\mathbf{W}]^T[M]\,[\mathbf{W}]\,\ddot{\mathbf{Q}}(t) + [\mathbf{W}]^T\mathbf{g}([\mathbf{W}]\,\mathbf{Q}(t), [\mathbf{W}]\,\dot{\mathbf{Q}}(t)) = [\mathbf{W}]^T\mathbf{f}(t) \quad , \quad t \in ]t_0, T]\,, \tag{2.14}$$

and the initial conditions

$$\mathbf{Q}(t_0) = [\mathbf{W}]^T\mathbf{y}_0 \quad , \quad \dot{\mathbf{Q}}(t_0) = [\mathbf{W}]^T\mathbf{y}_1\,. \tag{2.15}$$

The $\mathbb{R}^n$-valued stochastic solution $\{\mathbf{Q}(t;\boldsymbol{\alpha}), t \in [t_0, T]\}$ of Equation (2.14) with the initial conditions (2.15) depends on $\boldsymbol{\alpha} \in \mathcal{C}_{\boldsymbol{\alpha}}$. The stochastic process $\{\mathbf{Y}^{(n)}(t;\boldsymbol{\alpha}), t \in [t_0, T]\}$ is the $n$-order approximation of the stochastic process $\{\mathbf{Y}(t), t \in [t_0, T]\}$. As for the QoIs, the corresponding approximation $\{\mathbf{O}^{(n)}(t;\boldsymbol{\alpha}), t \in [t_0, T]\}$ of $\{\mathbf{O}(t), t \in [t_0, T]\}$ is given, for all $t \in [t_0, T]$ and $\boldsymbol{\alpha} \in \mathcal{C}_{\boldsymbol{\alpha}}$, by

$$\mathbf{O}^{(n)}(t;\boldsymbol{\alpha}) = \mathbf{h}_{\text{time}}(\mathbf{Y}^{(n)}(t;\boldsymbol{\alpha}))\,. \tag{2.16}$$

### 2.4. Construction of the stochastic reduced-order basis

As explained in Section 2.4, the model-form uncertainties of the computational model of interest are taken into account by applying the nonparametric probabilistic method proposed in [4]. This method consists in substituting the reduced-order basis $[V]$ by a stochastic reduced-order basis $[\mathbf{W}]$ in the PROM. Due to Equation (2.11), the $N \times n$ random matrix $[\mathbf{W}]$ is constructed on a subset of the compact Stiefel manifold associated with Equation (2.12). A description of the stochastic model underlying $[\mathbf{W}]$ can be found in [4]. The random matrix $[\mathbf{W}]$, which verifies Equations (2.11) and (2.12), is a second-order random variable defined on a probability space $(\Theta, \mathcal{T}, \mathcal{P})$. Specifically, it is built as follows:

$$[\mathbf{W}] = ([V] + s\,[\mathbf{Z}])\,[H_s(\mathbf{Z})]\,, \tag{2.17}$$

$$[H_s(\mathbf{Z})] = ([I_n] + s^2\,[\mathbf{Z}]^T[\mathbf{Z}])^{-1/2}\,, \tag{2.18}$$

$$[\mathbf{Z}] = [\mathbf{A}] - [V]\,[\mathbf{D}]\,, \tag{2.19}$$

$$[\mathbf{D}] = ([V]^T[\mathbf{A}] + [\mathbf{A}]^T[V])/2\,, \tag{2.20}$$

$$[\mathbf{A}] = [\mathbf{U}] - [B]\,\{[B]^T\,[\mathbf{U}]\}\,, \tag{2.21}$$

$$[\mathbf{U}] = [\mathbf{G}(\beta)]\,[\sigma]\,, \tag{2.22}$$

where $[\mathbf{G}(\beta)]$ is a second-order, centered, $\mathbb{M}_{N,n}$-valued random matrix defined on $(\Theta, \mathcal{T}, \mathcal{P})$ and $[\sigma]$ is a given upper triangular matrix in $\mathbb{M}_n^{\mathrm{u}}$ (positive or zeros diagonal entries). The probability distribution of $[\mathbf{W}]$ and a generator of this random matrix are described in Appendix D of [4]. The $m_{\boldsymbol{\alpha}} = 2 + n(n+1)/2$ hyperparameters of the stochastic model of $[\mathbf{W}]$ are as follows:

- A deterministic, real parameter $s$ satisfying $0 \leq s_L \leq s \leq s_U \leq 1$, where the lower bound $s_L$ and upper bound $s_U$ are fixed; note that if $s_L = s = 0$, then $[\mathbf{W}] = [V]$ is deterministic and there are no statistical fluctuations.
- A deterministic, real parameter $\beta$ satisfying $0 < \beta_L \leq \beta \leq \beta_U < +\infty$, where the lower bound $\beta_L$ and upper bound $\beta_U$ are fixed.
- An upper triangular matrix $[\sigma]$ in $\mathbb{M}_n^{\mathrm{u}}$ (positive or zero diagonal entries) parameterized by $m_\sigma = n(n+1)/2$ and characterized by the following property: for all $1 \leq k \leq k' \leq n$, $[\sigma_L]_{kk'} \leq [\sigma]_{kk'} \leq [\sigma_U]_{kk'}$. The lower bounds are the entries of the upper triangular matrix $[\sigma_L]$: they are fixed in $\mathbb{M}_n^{\mathrm{u}}$. The upper bounds are the entries of the upper triangular matrix $[\sigma_U]$: they are fixed in $\mathbb{M}_n^{\mathrm{u}}$.

Hence, the hyperparameter of the nonparametric probabilistic method reviewed here is the vector $\boldsymbol{\alpha} = (s, \beta, \{[\sigma]_{kk'}, 1 \leq k \leq k' \leq n\})$ of dimension $m_{\boldsymbol{\alpha}} = 2 + n(n+1)/2$. It can be simply re-written as $\boldsymbol{\alpha} = (s, \beta, [\sigma])$. It belongs to the admissible set $\mathcal{C}_{\boldsymbol{\alpha}}$ defined by the lower and upper bounds of every one of its components. For any fixed value of $\boldsymbol{\alpha}$ in $\mathcal{C}_{\boldsymbol{\alpha}}$, the generator of the random matrix $[\mathbf{G}(\beta)]$ described in [4] enables the computation of any realization $[\mathbf{G}(\beta;\theta)]$ of $[\mathbf{G}(\beta)]$ for $\theta$ in $\Theta$. It also enables the obtention of the corresponding realization $[\mathbf{W}(\theta)]$ of the random matrix $[\mathbf{W}]$, using Equations (2.17) to (2.22).

Note that even though the SROB $[\mathbf{W}]$ depends on the hyperparameter $\boldsymbol{\alpha}$ (see Equations (2.17) to (2.22)), this dependency is not highlighted in the remainder of this paper by writing $[\mathbf{W}(\boldsymbol{\alpha})]$, for the sake of simplicity.

*REMARK 2.* Regarding the interpretation of the individual hyperparameters constituting the vector-valued hyperparameter $\alpha$, the following is noteworthy:

- From Eqs. (2.17) to (2.22), it follows that the random matrix $[\mathbf{W}]$ depends on the hyperparameters $s$ and $[\sigma]$, and $s$ is a scaling parameter of the matrix-valued hyperparameter $[\sigma]$.
- From the construction of the random matrix $[\mathbf{G}(\beta)]$ described in details in [4], it follows that the hyperparameter $\beta$ enables the control of the statistical correlations between the random components of a same column of the random matrix $[\mathbf{G}(\beta)]$.
- In [4], $[\sigma]$ was introduced as the upper triangular matrix resulting from the Cholesky factorization of an unknown but positive-definite square matrix $[c_n]$ that participates to the construction of the correlation tensor of the random matrix $[\mathbf{U}]$ (specifically, see Eq. (D.25) on page 875 of [4]). Hence, identifying $[c_n] = [\sigma]^T[\sigma]$ can be performed by identifying instead the upper triangular matrix $[\sigma]$. More generally, one can identify an upper triangular matrix $[\sigma]$ with positive *or zero* diagonal entries and define $[c_n] = [\sigma]^T[\sigma]$. In both cases, the matrix-valued hyperparameter $[\sigma]$ enables the control of the statistical correlations between the columns of the random matrix $[\mathbf{G}(\beta)]$ using roughly half the number of individual hyperparameters that would be otherwise needed to identify $[c_n]$ directly.

## 3. IDENTIFICATION OF THE HYPERPARAMETER

Let $\{\mathbf{o}^{\text{target}}(t), t \in [t_0, T]\}$ with values in $\mathbb{R}^{m_{\mathbf{o}}}$ be the vector of observed targets associated with $\{\mathbf{y}^{\text{target}}(t), t \in [t_0, T]\}$ with values in $\mathbb{R}^N$. If the model-form uncertainties and/or modeling errors tainting the SPROM defined by (2.13) and (2.14) are due only to model reduction, then $\mathbf{o}^{\text{target}} = \mathbf{o}$. On the other hand, if they are due to both of the model reduction process and the model-form uncertainties and/or modeling errors existing in the HDM, then $\mathbf{o}^{\text{target}} = \mathbf{o}^{\text{exp}}$, where $\mathbf{o}^{\text{exp}}$ is based on experimental data.

The identification of the hyperparameter $\boldsymbol{\alpha}$ in $\mathcal{C}_{\boldsymbol{\alpha}} \subset \mathbb{R}^{m_\alpha}$ can be performed using various approaches for formulating a statistical inverse problem related to $\mathbf{o}^{\text{target}}$, including the maximum likelihood and nonlinear Least-Squares methods [14, 15, 16]. For example, in the context of nonlinear transient dynamics problems, the vector-valued hyperparameter $\boldsymbol{\alpha}$ was determined in [4] by:

- Defining a total cost function of the form

$$J(\boldsymbol{\alpha}) = w_J\, J_{\text{mean}}\,(\boldsymbol{\alpha}) + (1 - w_J)\, J_{\text{std}}\,(\boldsymbol{\alpha})\,, \tag{3.1}$$

  where:
  - $J_{\text{mean}}\,(\boldsymbol{\alpha})$ and $J_{\text{std}}\,(\boldsymbol{\alpha})$ are partial cost functions formulated using the deterministic, time-dependent functions $\{\mathbf{o}^{\text{target}}(t), t \in [t_0, T]\}$ and $\{\mathbf{o}^{(n)}(t;\boldsymbol{\alpha}), t \in [t_0, T]\}$, and the stochastic, time-dependent functions $\{\mathbf{O}^{(n)}(t;\boldsymbol{\alpha}), t \in [t_0, T]\}$. These functions are designed to control the discrepancies between the mean values and standard deviations of $\mathbf{O}^{(n)}(t)$ and $\mathbf{o}^{\text{target}}(t)$, respectively.
  - $w_J$ is a real scalar statisfying $0 \leq w_J \leq 1$, so that the total cost function $J$ is a convex combination of the partial cost functions $J_{\text{mean}}$ and $J_{\text{std}}$ .
- Solving the associated minimization problem

$$\boldsymbol{\alpha}^{\text{opt}} = \min_{\boldsymbol{\alpha} \in \mathcal{C}_{\boldsymbol{\alpha}}} J(\boldsymbol{\alpha})\,. \tag{3.2}$$

In [5] and [10], this approach was extended to the context of eigenvalue computations where the observed targets are algebraic rather than time-dependent quantities, by tailoring the definitions of $J_{\text{mean}}\,(\boldsymbol{\alpha})$ and $J_{\text{std}}\,(\boldsymbol{\alpha})$ to this context. In any case, the reader is referred to [4] for the detailed development of the expressions of these two partial cost functions in the case of applications in nonlinear structural dynamics, and to [5] and [10] for that of applications in vibration analysis. These expressions are not repeated here because an alternative approach to (3.1) and (3.2) is proposed in this paper for identifying the hyperparameter $\boldsymbol{\alpha}$, for the main reason outlined below.

Indeed, the optimization problem defined by (3.1) and (3.2) can be solved using either a deterministic method such as the interior point algorithm (with potential problems due to the non convexity of the cost function $J(\boldsymbol{\alpha})$), or a probabilistic technique such as a genetic algorithm. For each candidate/iterate solution $\boldsymbol{\alpha} = (s, \beta, [\sigma])$ proposed by the optimizer, the evaluation of $J(\boldsymbol{\alpha})$ is performed in either case using a Monte Carlo simulation method equipped with $\nu_{\text{sim}}$ independent realizations $\{[\mathbf{G}(\beta;\theta_\ell)], \ell = 1, \ldots, \nu_{\text{sim}}\}$ of the random matrix $[\mathbf{G}(\beta)]$ as the stochastic solver for Equations (2.13) to (2.22), and the mean-square convergence with respect to $\nu_{\text{sim}}$ is controlled. If the chosen optimizer incurs $\nu_J$ evaluations of the cost function $J(\boldsymbol{\alpha})$, it incurs $\nu_J \times \nu_{\text{sim}}$ repeated solutions of the governing forward problem using the nonlinear SPROM (2.13, 2.14). During the application of the nonparametric probabilistic method outlined above to the UQ of nonlinear transient dynamics problems [4] and generalized eigenvalue problems [5, 10], it was found that as soon as the dimension of the SPROM $n$ dictated by accuracy requirements becomes greater than a few units, $m_\alpha$ becomes sufficiently large to induce a very large number of optimization iterations $\nu_J$. In this case, the numerical cost of the identification approach of the hyperparameter $\boldsymbol{\alpha}$ defined by (3.1) and (3.2) becomes significant, if not prohibitive, particularly in the context of model reduction. For this reason, a faster alternative approach for identifying $\boldsymbol{\alpha}$ is presented next.

## 4. PREDICTOR-CORRECTOR METHOD FOR THE ESTIMATION OF THE HYPERPARAMETER

The alternative approach presented here for identifying the vector-valued hyperparamter $\boldsymbol{\alpha}$ of the probability model of the SROB operates on Fourier transforms of target values of time-dependent QoIs, and Fourier transforms of counterpart values predicted using the constructed nonlinear SPROM. The choice of working in the frequency domain instead of the time domain is made here

because in general, the phase between a time-dependent target value of a QoI and its counterpart value predicted using a PROM is generally unknown, and that between a time-dependent target value of a QoI and its counterpart value predicted using an SPROM is random and therefore never known. Hence, it is easier to formulate the discrepancy (error) in the frequency domain by focusing on the modulus of the complex-valued Fourier transforms of measured and computed time domain responses.

The method presented here for identifying $\boldsymbol{\alpha}$ is a predictor-corrector approach based on a probabilistic learning on manifolds. Its predictor component can be written as $\boldsymbol{\alpha}^0 = (s^0, \beta^0, [\sigma^0])$ in $\mathcal{C}_{\boldsymbol{\alpha}}$, where $[\sigma^0]$ is a predictor for the matrix-valued hyperparameter $[\sigma] \in \mathbb{M}_n^{\mathrm{u}}$ that is algebraically constructed through adapted approximations, $\beta^0 = (\beta_L + \beta_U)/2$, where the subscripts $L$ and $U$ designate here and throughout the remainder of this paper lower and upper bounds of the quantity they designate, and $s^0$ is identified by constructing an optimization problem and solving it using a fast grid search in algorithm.

If the predictor must be improved, the proposed corrector is based on solving another optimization problem where the cost function is constructed using the extreme values statistics of a positive-valued random variable $\mathbb{Q}$ that quantifies the error between the frequency-dependent target values of some QoIs and their frequency-dependent predicted values. The independent realizations of the extreme values statistics are extracted from a large number of additional realizations that are computed by a probabilistic learning on manifolds for $\mathbb{Q}$ without exercising the nonlinear SPROM. In this case, the optimizer is the interior point algorithm. The PLM requires the knowledge of an initial dataset of a relatively small number of points $\{\boldsymbol{\alpha}_d^{\ell'}\}_{\ell'}$ localized in the neighborhood of the predictor $\boldsymbol{\alpha}^0$, and the corresponding realizations $\{\mathbb{q}_d^{\ell'}\}_{\ell'}$ of $\{\mathbb{Q}(\boldsymbol{\alpha}_d^{\ell'})\}_{\ell'}$ computed using the nonlinear SPROM.

### *4.1. Observation in the frequency domain and stochastic error*

Let $\mathcal{B}_o = [0\,, \omega_o]$ denote a frequency band of observation that is sampled by $m_\omega$ points using a constant frequency increment $\Delta\omega$, which implies that $\omega_0 = m_\omega\,\Delta\omega$. For any vector-valued time function $\{\mathbf{z}(t), t \in [t_0, T]\}$, the associated complex vector-valued frequency function $\{\widehat{\mathbf{z}}(\omega), \omega \in \mathcal{B}_o\}$ is defined as $\widehat{\mathbf{z}}(\omega) = \displaystyle\int_{t_0}^{T} e^{-i\omega\,t}\mathbf{z}(t)\,dt$. This function can be viewed as the restriction to $\mathcal{B}_o$ of the Fourier transform of $\{\mathbf{z}(t)\,t \in [t_0\,, T]\}$, if $\mathbf{z}(t) = \mathbf{0}$ for all $t$ in $\mathbb{R}\backslash[t_0, T]$ (which will be the case, up to a small relative error, for the quantities to which this Fourier transform will be applied to in the remainder of this paper).

Let also $\widehat{\mathbf{o}}(\omega) = (\widehat{o}_1(\omega), \ldots, \widehat{o}_{m_{\mathbf{o}}}(\omega)) \in \mathbb{C}^{m_{\mathbf{o}}}$, with $\widehat{\mathbf{o}}(\omega) = \mathbf{h}_{\text{freq}}(-\omega^2\,\widehat{\mathbf{y}}(\omega))$, denote the complex vector of frequency-dependent QoIs, $\{\widehat{\mathbf{o}}(\omega), \omega \in \mathcal{B}_o\}$, associated with the time-dependent QoIs, $\{\mathbf{o}(t), t \in [t_0, T]\}$. The mapping $\mathbf{h}_{\text{freq}}$ from $\mathbb{C}^N$ into $\mathbb{C}^{m_{\mathbf{o}}}$ is linear and represented by the matrix $[\mathbb{O}]$ in $\mathbb{M}_{m_{\mathbf{o}},N}$. It extracts the observed dofs from the vector $-\omega^2\,\widehat{\mathbf{y}}(\omega) \in \mathbb{C}^N$. Such a definition of frequency-dependent QoIs is applied here to $\{\mathbf{o}^{(n)}(t), t \in [t_0, T]\}$ defined in Equation (2.10), the stochastic observation $\{\mathbf{O}^{(n)}(t, \boldsymbol{\alpha}), t \in [t_0, T]\}$ defined in Equation (2.16), and to the target data $\{\mathbf{o}^{\text{target}}(t), t \in [t_0, T]\}$ defined in Section 3. It follows that for all $\omega$ in $\mathcal{B}_o$,

$$\widehat{\mathbf{o}}^{\,(n)}(\omega) = -\omega^2[\mathbb{O}]\widehat{\mathbf{y}}^{(n)}(\omega))\ ,\ \widehat{\mathbf{O}}^{(n)}(\omega;\boldsymbol{\alpha}) = -\omega^2[\mathbb{O}]\widehat{\mathbf{Y}}^{(n)}(\omega;\boldsymbol{\alpha}))\ ,\ \widehat{\mathbf{o}}^{\,\text{target}}(\omega) = -\omega^2[\mathbb{O}]\widehat{\mathbf{y}}^{\text{target}}(\omega))\,.$$

Now, let $\{\mathbf{dB}^{(n)}(\omega;\boldsymbol{\alpha}) = (\mathrm{dB}_1^{(n)}(\omega;\boldsymbol{\alpha}), \ldots, \mathrm{dB}_{m_o}^{(n)}(\omega;\boldsymbol{\alpha}))\,, \omega \in \mathcal{B}_o\}$ denote the stochastic process defined on $(\Theta, \mathcal{T}, \mathcal{P})$ and indexed by $\mathcal{B}_o$, with values in $\mathbb{R}^{m_o}$ such that, for all $j = 1, \ldots, m_o$,

$$\mathrm{dB}_j^{(n)}(\omega;\boldsymbol{\alpha}) = \log_{10}(|\widehat{O}_j^{(n)}(\omega;\boldsymbol{\alpha})|)\,.$$

The following functions $\omega \mapsto \mathbf{db}^{\text{target}}(\omega) = (\mathrm{db}_1^{\text{target}}(\omega), \ldots, \mathrm{db}_{m_o}^{\text{target}}(\omega))$ and $\omega \mapsto \mathbf{db}^{(n)}(\omega) = (\mathrm{db}_1^{(n)}(\omega), \ldots, \mathrm{db}_{m_o}^{(n)}(\omega))$ with values in $\mathbb{R}^{m_o}$ are similarly defined on $\mathcal{B}_o$ – that is, such that for all $j = 1, \ldots, m_o$,

$$\mathrm{db}_j^{\text{target}}(\omega) = \log_{10}(|\widehat{o}_j^{\,\text{target}}(\omega)|) \quad , \quad \mathrm{db}_j^{(n)}(\omega) = \log_{10}(|\widehat{o}_j^{(n)}(\omega)|)\,.$$

For a fixed value of the dimension $n$ of the SPROM, the stochastic discrepancy (error) between the target values of the QoIs and their counterpart values predicted using the nonlinear SPROM is denoted by the following positive-valued random variable $\mathbb{Q}(\boldsymbol{\alpha}) = \{\mathbb{Q}(\boldsymbol{\alpha};\theta), \theta \in \Theta\}$ defined on $(\Theta, \mathcal{T}, \mathcal{P})$ as follows

$$\mathbb{Q}(\boldsymbol{\alpha};\theta) = \int_{\mathcal{B}_o} \|\mathbf{db}^{\text{target}}(\omega) - \mathbf{dB}^{(n)}(\omega;\boldsymbol{\alpha};\theta)\|^2_{\boldsymbol{\mu}}\, d\omega\,, \tag{4.1}$$

where, for any function $\omega \mapsto \mathbf{r}(\omega) = (r_1(\omega), \ldots, r_{m_o}(\omega))$ defined on $\mathcal{B}_o$ with values in $\mathbb{R}^{m_o}$ and for all $\omega \in \mathcal{B}_o$, $\|\mathbf{r}(\omega)\|_{\boldsymbol{\mu}}$ is defined as

$$\|\mathbf{r}(\omega)\|_{\boldsymbol{\mu}} = \{\sum_{j=1}^{m_o} (r_j(\omega))^2\, \mu_j(\omega)\}^{1/2}\,, \tag{4.2}$$

with

$$\mu_j(\omega) = \frac{e_j(\omega)}{\max_{\omega\in\mathcal{B}_o}\{e_j(\omega)\}} \quad \text{and} \quad e_j(\omega) = 10^{\mathrm{db}_j^{\text{target}}(\omega)}\, |\mathrm{db}_j^{\text{target}}(\omega) - \mathrm{db}_j^{(n)}(\omega)|\,. \tag{4.3}$$

For a fixed hyperparameter $\boldsymbol{\alpha}$, the realization $\mathbf{dB}^{(n)}(\omega;\boldsymbol{\alpha};\theta)$ of positive-valued random variable $\mathbf{dB}^{(n)}(\omega;\boldsymbol{\alpha})$ (for which $\omega$ given is fixed in $\mathcal{B}_o$ ) is computed by solving the realization $\theta \in \Theta$ of the nonlinear SPROM.

Next, two real numbers $q^{\text{target}}$ and $q^{\text{error}}$ associated with the random variable $\mathbb{Q}(\boldsymbol{\alpha})$ are defined as follows

$$q^{\text{target}} = \int_{\mathcal{B}_o} \|\mathbf{db}^{\text{target}}(\omega)\|^2_{\boldsymbol{\mu}}\, d\omega \quad , \quad q^{\text{error}} = \int_{\mathcal{B}_o} \|\mathbf{db}^{\text{target}}(\omega) - \mathbf{db}^{(n)}(\omega)\|^2_{\boldsymbol{\mu}}\, d\omega\,. \tag{4.4}$$

These two positive numbers are used in the remainder of this paper to normalize the cost function of an optimization problem devoted to the identification of the hyperparameter $\boldsymbol{\alpha}$.

*REMARK 3.* For the purpose of the work presented in this paper, the function $\boldsymbol{\mu}(\omega) = (\mu_1(\omega), \ldots, \mu_{m_o}(\omega))$ defined above was designed by numerical experiments to privilege those frequencies in the band $\mathcal{B}_o$ for which the error is the greatest and make it possible to increase the sensitivity of the random variable $\mathbb{Q}(\boldsymbol{\alpha})$ to variations in the hyperparameter $\boldsymbol{\alpha}$.

### *4.2. Construction of the predictor*

The construction of the predictor is performed in two stages as described below.

**Stage 1**. The first stage consists in constructing a predictor $[\sigma^0]$ for $[\sigma]$. Examining the algebraic structure of Equation (2.13) and Equations (2.17) to (2.22), the idea here is to estimate $[\sigma^0]$ by using an approximation of the error between the generalized response obtained using the nonlinear SPROM and the projection of the target on the SROB. Let $\mathbf{a}^{(n)}(\omega) = -\omega^2\widehat{\mathbf{q}}(\omega) \in \mathbb{C}^n$ be the frequency-dependent generalized acceleration associated with the generalized coordinate $\mathbf{q}$ computed using the nonlinear SPROM (see Section 2.2) and let $\mathbf{a}^{\text{target}}(\omega)$ be its counterpart target. From Equation (2.6) and the least-squares inversion of Equation (2.7), it follows that

$$\mathbf{a}^{(n)}(\omega) = [V]^T(-\omega^2\widehat{\mathbf{y}}^{(n)}(\omega)) \quad , \quad \mathbf{a}^{\text{target}}(\omega) = [V]^T(-\omega^2\widehat{\mathbf{y}}^{\text{target}}(\omega))\,. \tag{4.5}$$

In order to focus the construction of the error on the QoIs, the computation of $\mathbf{a}^{(n)}(\omega) = (a_1^{(n)}(\omega), \ldots, a_n^{(n)}(\omega))$ and $\mathbf{a}^{\text{target}}(\omega) = (a_1^{\text{target}}(\omega), \ldots, a_n^{\text{target}}(\omega)$ can be performed only for the QoIs rather than all dofs of the HDM. This is well adapted to the case where the target consists of experimental data for the QoIs. In any case, this error can be written as $\mathbf{a}^{\text{error}}(\omega) = (a_1^{\text{error}}(\omega), \ldots, a_n^{\text{error}}(\omega))$ where, for all $\omega$ in $\mathcal{B}_o$,

$$a_k^{\text{error}}(\omega) = |\,(|a_k^{\text{target}}(\omega)| - |a_k^{(n)}(\omega)|)\,| \quad , \quad k = 1, \ldots, n\,.$$

Define

$$[g] = \int_{\mathcal{B}_o} \mathbf{a}^{\text{error}}(\omega)\, \mathbf{a}^{\text{error}}(\omega)^T \, d\omega \,.$$

This positive matrix is assumed to be definite ($[g] \in \mathbb{M}_n^+$) and is scaled as follows to obtain the matrix $[g_s] \in \mathbb{M}_n^+$

$$[g_s] = \frac{1}{g_{\text{max}}}\,[g] \quad , \quad g_{\text{max}} = \max_{k,k'} \{\, [g]_{kk'} \,\} \,.$$

Let $[\widetilde{\sigma}_f^0]/\sqrt{m_\omega}$ be the upper triangular $(n \times n)$ real matrix resulting from the Cholesky factorization of the matrix $[g_s]$, where the normalization constant, $m_\omega$, is the number of sampling points of the frequency band $\mathcal{B}_o$ (see Section 4.1). For constructing $[\sigma^0]$, the matrix $[\widetilde{\sigma}_f^0]$ is first transformed into a sparse upper matrix belonging to $\mathbb{M}_n^{\text{u}}$ in order to keep only its entries that have a major influence on controlling the error between the target and the generalized response computed using the nonlinear SPROM. This construction allows the reduction of the number of hyperparameters, $m_{\boldsymbol{\alpha}}$, whose maximum value is $n(n+1)/2$. It is carried out in two steps as follows. Let $\tau_{\text{rel}}$ be a relative tolerance (for instance $\tau_{\text{rel}} = 0.15$), and $\tau = \tau_{\text{rel}} \max_k [\widetilde{\sigma}_f^0]_{kk}$ be the corresponding absolute tolerance. Define the upper triangular matrix $[\sigma_f^0]$ in $\mathbb{M}_n^{\text{u}}$ as follows

$$[\sigma_f^0]_{kk'} = 0 \ \text{ if } \ [\widetilde{\sigma}_f^0]_{kk'} \leq \tau \quad \text{and} \quad [\sigma_f^0]_{kk'} = [\widetilde{\sigma}_f^0]_{kk'} \ \text{ if } \ [\widetilde{\sigma}_f^0]_{kk'} > \tau \,.$$

Then, a predictor $[\sigma^0]$ for $[\sigma]$ is obtained by keeping in each column $k'$ of the matrix $[\sigma_f^0]$ only the entry corresponding to the maximum over $k$ of $|[\sigma_f^0]_{kk'}|$. For $k'$ fixed in $\{1, \ldots, n\}$, let $k_0(k') = \arg\max_k \{|[\sigma_f^0]_{kk'}|\}$. Thus, for $1 \leq k \leq k' \leq n$, the upper triangular matrix $[\sigma^0]$ is defined as

$$[\sigma^0]_{kk'} = 0 \ \text{ if } \ k \neq k_0(k') \quad \text{and} \quad [\sigma^0]_{kk'} = [\sigma_f^0]_{kk'} \ \text{ if } \ k = k_0(k') \,. \tag{4.6}$$

Now, let $m_\sigma^0$ denote the number of non-zero entries in the matrix $[\sigma^0]$. Given that $1 \leq m_\sigma^0 \leq n$, it follows that $m_{\boldsymbol{\alpha}}^0 = 2 + m_\sigma^0 \leq 2 + n$, which highlights the substantial reduction of the number of hyperparameters in the matrix $[\sigma]$ achieved by the above construction of a predictor for this matrix.

**Stage 2**. The second stage consists in constructing a predictor $s^0$ for $s$. Let $\beta^0 = (\beta_L + \beta_U)/2$ denote a fixed value of the hyperparameter $\beta$, and recall the predictor $[\sigma^0]$ for $[\sigma]$ defined in Equation (4.6). The predictor $s^0$ for the hyperparameter $s \in [s_L, s_U] \subset \mathbb{R}^+$ is proposed here as the solution of the following optimization problem

$$s^0 = \arg \min_{s \in [s_L, s_U]} J_{\text{freq}}(s) \,, \tag{4.7}$$

where

$$J_{\text{freq}}(s) = \frac{w_J}{q^{\text{target}}}\, \text{mean}_{\mathbb{Q}}(s) + \frac{1 - w_J}{\gamma_J\, q^{\text{target}}}\, |\text{std}_{\mathbb{Q}}(s) - \gamma_J\, q^{\text{error}}| \,, \tag{4.8}$$

$$\begin{aligned} \text{mean}_{\mathbb{Q}}(s) &= E\{\mathbb{Q}(s, \beta^0, [\sigma^0])\} \,, & (4.9)\\ \text{std}_{\mathbb{Q}}(s) &= \{E\{\mathbb{Q}(s, \beta^0, [\sigma^0])^2\} - \text{mean}_{\mathbb{Q}}(s)^2\}^{1/2} \,, & (4.10)\\ \boldsymbol{\alpha} &= (s, \beta^0, [\sigma^0]) \,, & (4.11) \end{aligned}$$

$w_J$ is fixed in $[0, 1]$ (for example, $w_J = 0.5$), $q^{\text{target}}$ and $q^{\text{error}}$ are defined in Equation (4.4), and $\gamma_J$ is a constant fixed in $[0, 1]$. The value of $\gamma_J$ is application-dependent and must be estimated as explained in Section 5. The nonconvex optimization problem defined in (4.7) is solved using a grid search algorithm associated with a grid $s_L = s_1 < s_2 < \ldots < s_{m_s} = s_U$ constituted of $m_s$ sampling points of the interval $[s_L, s_U]$. For a given $s_j$, the mean value and standard deviation of the random variable $\mathbb{Q}(s_j, \beta^0, [\sigma^0])$ are estimated using the classical statistical estimators and $\nu_s$ independent realizations of the random variable $\mathbb{Q}(s_j, \beta^0, [\sigma^0])$ computed by solving $\nu_s$ realizations of the nonlinear SPROM. Therefore, the computational cost of the proposed construction of the predictor $s^0$ is dominated by that of $m_s \times \nu_s$ realizations of the nonlinear SPROM.

This completes the description of the construction of the predictor $\boldsymbol{\alpha}^0$ for the hyperparameter $\boldsymbol{\alpha}$, which can be written as

$$\boldsymbol{\alpha}^0 = (s^0, \beta^0, [\sigma^0]) \in \mathbb{R}^{m^0_{\boldsymbol{\alpha}}} \quad \text{with} \quad m^0_{\boldsymbol{\alpha}} = 2 + m^0_\sigma \,. \tag{4.12}$$

*REMARK 4.* In **Stage 2** of the construction of the predictor $\boldsymbol{\alpha}^0$, the value of $\beta$ is fixed to $\beta^0 = (\beta_L + \beta_U)/2$ and the hyperparameter $s$ is determined by solving the optimization problem (4.7), where $\beta^0$ and $[\sigma^0]$ are given. Alternatively, $[\sigma]$ could have been fixed to some value $[\sigma^0]$ and a different optimization problem could have been formulated to determine $s$ and $\beta$. However, numerical experiments carried out for assessing the performance of the predictor $\boldsymbol{\alpha}^0$ have revealed that the random matrix $[\mathbf{W}]$ is not sensitive with respect to different values of $\beta$. For this reason, the first aforementioned approach consisting in fixing $\beta$ to the midpoint value of the admissible interval $[\beta_L,\ \beta_U]$, where appropriate values of $\beta_L$ and $\beta_U$ are determined experimentally, was chosen in formulating **Stage 2** of the construction of $\boldsymbol{\alpha}^0$ (note however that in the construction of the corrector $\boldsymbol{\alpha}^{\text{opt}}$ described in Section 4.4, $\beta$ is not fixed to some value *a priori*).

### *4.3. Quality assessment of the predictor*

The quality of the predictor $\boldsymbol{\alpha}^0 = (s^0, \beta^0, [\sigma^0])$ of dimension $m^0_{\boldsymbol{\alpha}}$ can be assessed *a posteriori*, using the following empirical indicator that is adapted to the logarithmic scale,

$$\mathcal{I} = \mathcal{I}^+ + \frac{1}{\mathcal{I}^-} \quad , \quad \mathcal{I}^\pm = \sum_{j=1}^{m_{\mathbf{o}}} \mathcal{I}^\pm_j \,, \tag{4.13}$$

where

$$\mathcal{I}^\pm_j = \frac{\int_{\mathcal{B}_o} |\text{db}^{\text{target}}_j(\omega) - \text{db}^\pm_j(\omega)|\, d\omega}{\int_{\mathcal{B}_o} |\text{db}^{\text{target}}_j(\omega) + \text{db}^\pm_j(\omega)|\, d\omega} \,, \tag{4.14}$$

and $\{\text{db}^+_j(\omega), \omega \in \mathcal{B}_o\}$ and $\{\text{db}^-_j(\omega), \omega \in \mathcal{B}_o\}$ are the upper and lower envelopes of the confidence region of the family of random variables $\{\text{dB}^{(n)}_j(\omega, \boldsymbol{\alpha}^0), \omega \in \mathcal{B}_o\}$ for a probability level $p_c$ (for instance, $p_c = 0.98$). This means that, for all $\omega$ in $\mathcal{B}_o$,

$$\text{Proba}\{\text{dB}^{(n)}_j(\omega, \boldsymbol{\alpha}^0) \le \text{db}^+_j(\omega)\} \ge p_c \quad , \quad \text{Proba}\{\text{dB}^{(n)}_j(\omega, \boldsymbol{\alpha}^0) \ge \text{db}^-_j(\omega)\} \ge 1 - p_c \,. \tag{4.15}$$

The dimensionless indicator $\mathcal{I}$ defined in Equation (4.13) is a measure of a membership of the target $\{\text{db}^{\text{target}}_j(\omega, \boldsymbol{\alpha}^0), \omega \in \mathcal{B}_o\}$ to the confidence region $\{[\text{db}^-_j(\omega), \text{db}^+_j(\omega)], \omega \in \mathcal{B}_o\}$. The smaller the $\mathcal{I}$, the better the prediction of the probabilistic model of model-form uncertainties to represent the target.

### *4.4. Construction of the corrector*

The computation of the corrector $\boldsymbol{\alpha}^{\text{opt}}$ associated with the predictor $\boldsymbol{\alpha}^0$ for the hyperparameter $\boldsymbol{\alpha}$ is based on:

- The following hypotheses:
  - The hyperparameter $\boldsymbol{\alpha} = (s, \beta, [\sigma])$ estimated by the corrector is an update of the predictor value $\boldsymbol{\alpha}^0 = (s^0, \beta^0, [\sigma^0])$ defined in Equation (4.12); therefore, the dimension of $\boldsymbol{\alpha}$ remains $m^0_{\boldsymbol{\alpha}} = 2 + m^0_\sigma$.
  - The $m^0_\sigma$ non-zero entries of the upper triangular matrix $[\sigma^0]$ are represented by the vector $\boldsymbol{\sigma}^0 = (\sigma^0_1, \ldots, \sigma^0_{m^0_\sigma})$.
  - The corrector $[\sigma]$ for $[\sigma^0]$ has the same sparsity pattern as $[\sigma^0]$ and therefore $m^0_\sigma$ non-zero entries; these are similarly represented by the vector $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{m^0_\sigma})$.
  - The bounds of the hyperparameters $s$, $\beta$, and $\boldsymbol{\sigma}$, which have to be specified for the corrector algorithm, are defined in terms of the known values $s^0$, $\beta^0$, and $\boldsymbol{\sigma}^0$ of the predictor. These bounds enable the definition of the admissible set $\mathcal{C}^0_{\boldsymbol{\alpha}}$ of values for the hyperparameter $\boldsymbol{\alpha}$, where the superscript 0 emphasizes the dependence of $\mathcal{C}^0_{\boldsymbol{\alpha}}$ on $\boldsymbol{\alpha}^0$.

- The PLM for solving nonconvex optimization problems described in [11], and whose advantages relative to alternatives are discussed in [12] and [13].

*REMARK 5*. The probabilistic model of $[\mathbf{W}]$ being an informative prior probabilistic model, some of its parameters can be chosen arbitrarily. Here, the sparsity pattern of $[\sigma]$ is chosen a *a priori* to be the same as that of $[\sigma^0]$ – which was designed in Section 4.2 for the sole purpose of decreasing the number of individual hyperparameters to identify. This design is relevant to the capacity of this probabilistic model to generate, for given observations, the most narrow confidence regions containing the targets.

*4.4.1. Lower bounds for the corrector algorithm.* As already stated, the lower and upper bounds of the hyperparameters $s$, $\beta$, and $\boldsymbol{\sigma}$, which depend on $\boldsymbol{\alpha}^0$, define the admissible set $\mathcal{C}^0_{\boldsymbol{\alpha}} \subset \mathbb{R}^{m^0_{\boldsymbol{\alpha}}}$ for candidate values of $\boldsymbol{\alpha}$. The values of the constants appearing in the bounds presented below result from recent and previously performed numerical experiments [4, 5, 10]. These bounds are constructed as follows.

- The hyperparameter $s$ belongs to $[s_L, s_U]$, where $s_L = s^0(1-0.5)$ and $s_U = s^0(1+0.5)$. This interval is centered at the predictor $s^0$ and has been identified by numerical experiments. Since the predictor is assumed to be sufficiently good, $s$ should not deviate significantly from $s^0$.
- The hyperparameter $\beta$ belongs to $[\beta_L, \beta_U]$, where $\beta_L = 0.01$ and $\beta_U = 0.03$. These proposed values for the bounds are consistent with the optimal values obtained in all previous applications. However, they are not critical and can be slightly modified.
- The lower bound $\boldsymbol{\sigma}^L = (\sigma^L_1, \ldots, \sigma^L_{m^0_\sigma})$ and the upper bound $\boldsymbol{\sigma}^U = (\sigma^U_1, \ldots, \sigma^U_{m^0_\sigma})$ of the hyperparameter $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{m^0_\sigma})$ are such that, for all $k = 1, \ldots, m^0_\sigma$, $\sigma_k \in [\sigma^L_k, \sigma^U_k]$:
  - If $\sigma_k$ corresponds to a diagonal entry of $[\sigma]$, the lower bound is $\sigma^L_k = \max\{c_{dm} \times \sigma^0_k, \tau\}$ and the upper bound is $\sigma^U_k = c_{dM} \times \sigma^0_k$, where $c_{dM} = 4$ and $c_{dm} = 1/c_{dM}$.
  - If $\sigma_k$ corresponds to an upper extradiagonal entry of $[\sigma]$, there are two cases to consider depending on the sign of $\sigma^0_k$:
    - If $\sigma^0_k > 0$, the lower bound is $\sigma^L_k = c_{em} \times \sigma^0_k$ and the upper bound is $\sigma^U_k = c_{eM} \times \sigma^0_k$, where $c_{eM} = 20$ and $c_{em} = 1/c_{eM}$.
    - If $\sigma^0_k < 0$, the lower bound is $\sigma^L_k = c_{eM} \times \sigma^0_k$ and the upper bound is $\sigma^U_k = c_{em} \times \sigma^0_k$.

Note that the proposed values for $c_{dM}$ and $c_{eM}$ are sufficiently large, given the normalization of the matrix $[\sigma]$ (for further details, see [4]).

Hence, the admissible set for $\boldsymbol{\alpha}$ is defined as

$$\mathcal{C}^0_{\boldsymbol{\alpha}} = \{\boldsymbol{\alpha} = (s, \beta, \boldsymbol{\sigma}),\ s \in [s_L, s_U],\ \eta \in [\beta_L, \beta_U],\ \boldsymbol{\sigma} \in \prod_{k=1}^{m^0_\sigma} [\sigma^L_k, \sigma^U_k]\}. \tag{4.16}$$

*4.4.2. Optimization problem.* Similarly to the construction of the predictor $s^0$ for $s$ (see **Stage 2** in Section 4.2), the construction of the corrector for the hyperparameter $\boldsymbol{\alpha}$ consists in solving an optimization problem of the form

$$\boldsymbol{\alpha}^{\text{opt}} = \arg \min_{\boldsymbol{\alpha} \in \mathcal{C}^0_{\boldsymbol{\alpha}}} J(\boldsymbol{\alpha}), \tag{4.17}$$

where $\mathcal{C}^0_{\boldsymbol{\alpha}}$ is defined in Equation (4.16). A candidate choice for the cost function $J$ could be the same as that in Equation (4.8) – that is,

$$J(\boldsymbol{\alpha}) = \frac{w_J}{q^{\text{target}}} \text{mean}_{\mathbb{Q}}(\boldsymbol{\alpha}) + \frac{1 - w_J}{\gamma_J\, q^{\text{target}}} \left| \text{std}_{\mathbb{Q}}(\boldsymbol{\alpha}) - \gamma_J\, q^{\text{error}} \right|, \tag{4.18}$$

where $\text{mean}_{\mathbb{Q}}(\boldsymbol{\alpha})$ and $\text{std}_{\mathbb{Q}}(\boldsymbol{\alpha})$ are the mean value and the standard deviation of the random variable $\mathbb{Q}(\boldsymbol{\alpha})$ defined in Equation (4.1). However, this cost function is not the most appropriate here for the reasons given below.

The random variable $\mathbb{Q}(\boldsymbol{\alpha})$ is positive almost surely. If its standard deviation were to tend to zero, its mean value would also have to tend to zero by virtue of the Tchebychev inequality. Since in principle the target does not coincide with the response of the nonlinear SPROM, the confidence region of the prediction computed using the nonlinear SPROM does not have a zero width. This means that the standard deviation of $\mathbb{Q}(\boldsymbol{\alpha})$ must not tend to zero and therefore its mean value cannot *a priori* be zero for the optimal value $\boldsymbol{\alpha}^{\text{opt}}$. In other words, in order to reduce the mean value of $\mathbb{Q}(\boldsymbol{\alpha})$, it is necessary to generate statistical fluctuations in the nonlinear SPROM. This is the reason why, thanks to the presence of its second term, the cost function (4.8) (or (4.18)) allows the minimization of the mean value while maintaining a significant standard deviation. Nevertheless, since $\mathbb{Q}(\boldsymbol{\alpha})$ is a positive-valued random variable, solving the optimization (4.17) problem using the cost function defined in (4.18) consists in searching for the minimum of a positive-valued cost function. From a statistical viewpoint, the efficiency of the minimization process can be increased by using the minimum statistics of $\mathbb{Q}(\boldsymbol{\alpha})$ instead of all its possible values. For this reason, the cost function $J(\boldsymbol{\alpha})$ defined in (4.18) is replaced here by

$$J(\boldsymbol{\alpha}) = \frac{w_J}{q^{\text{target}}}\,\text{mean}_{\mathbb{Q}_{\text{min}}}(\boldsymbol{\alpha}) + \frac{1-w_J}{\gamma_J\, q^{\text{target}}}\,|\text{std}_{\mathbb{Q}_{\text{min}}}(\boldsymbol{\alpha}) - \gamma_J\, q^{\text{error}}|\,, \tag{4.19}$$

where $\text{mean}_{\mathbb{Q}_{\text{min}}}(\boldsymbol{\alpha})$ and $\text{std}_{\mathbb{Q}_{\text{min}}}(\boldsymbol{\alpha})$ are the mean value and standard deviation of the random variable $\mathbb{Q}_{\text{min}} = \min\{\mathbb{Q}\}$, which depends on $\boldsymbol{\alpha}$ and is defined as the minimum statistics for the random variable $\mathbb{Q}(\boldsymbol{\alpha})$. The mean value and the standard deviation of $\mathbb{Q}_{\text{min}}(\boldsymbol{\alpha})$ can be written as

$$\text{mean}_{\mathbb{Q}_{\text{min}}}(\boldsymbol{\alpha}) = E\{\mathbb{Q}_{\text{min}}(\boldsymbol{\alpha})\} \quad , \quad \text{std}_{\mathbb{Q}_{\text{min}}}(\boldsymbol{\alpha}) = \{E\{\mathbb{Q}_{\text{min}}(\boldsymbol{\alpha})^2\} - \text{mean}_{\mathbb{Q}_{\text{min}}}(\boldsymbol{\alpha})^2\}^{1/2}\,. \tag{4.20}$$

*4.4.3. Probabilistic learning on manifolds for estimating the cost function.* Recall from the technical arguments presented at the end of Section 3 that whichever algorithm is used to solve the optimization problem (4.17) equipped with a cost function such as that defined in Equations (4.19) and (4.20), this algorithm is bound to experience technical difficulties that can be summarized as follows. For each value of $\boldsymbol{\alpha}$ proposed by the optimizer, the mean value and the standard deviation of $\mathbb{Q}_{\text{min}}(\boldsymbol{\alpha})$ defined in Equation (4.20) should be estimated using a large number, $\nu_{\text{sim}}$, of independent realizations of the random variable $\mathbb{Q}(\boldsymbol{\alpha})$. These are typically obtained using the Monte Carlo technique applied to the nonlinear SPROM. Hence, if the chosen optimization algorithm requires $\nu_{\boldsymbol{\alpha}}$ evaluations of the cost function $J(\boldsymbol{\alpha})$, the nonlinear SPROM must be exercised $\nu_{\boldsymbol{\alpha}} \times \nu_{\text{sim}}$ times. If $n$ is not sufficiently small, $\nu_{\boldsymbol{\alpha}}$ can be expected to be large and the total solution time to be cost prohibitive.

On the other hand, the PLM described in this section will incur only $\nu_d$ calls to the nonlinear SPROM, where $\nu_d \ll \nu_{\boldsymbol{\alpha}} \times \nu_{\text{sim}}$. Therefore, it will drastically reduce the number of times this computational model is exercised and thereby drastically reduce the total computational cost associated with the identification of $\boldsymbol{\alpha}$. This method has two key components, namely:

- A PLM that enables the generation of $\nu_{\text{ar}}$ additional realizations concentrated on a manifold identified using an initial set of $\nu_d$ data points (with $\nu_d \ll \nu_{\text{ar}}$). These additional realizations are determined without performing any additional exploitation of the nonlinear SPROM.
- A smoothing technique for estimating $J(\boldsymbol{\alpha}^g)$ at any trial point $\boldsymbol{\alpha}^g$ in $\mathcal{C}^0_{\boldsymbol{\alpha}}$ generated for the optimization algorithm, using only the $\nu_{\text{ar}}$ additional realizations computed from the initial dataset.

The method consists of $4$ steps:

**Step 1.** The first step consists in constructing an initial dataset by exercising $\nu_d$ times the nonlinear SPROM. To this end, consider a fixed number $\nu_d$ of values of the hyperparameter $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^1_d, \ldots, \boldsymbol{\alpha}^{\nu_d}_d$ in $\mathcal{C}^0_{\boldsymbol{\alpha}}$, obtained using a uniform random sampling of $\nu_d$ points in the compact set $\mathcal{C}^0_{\boldsymbol{\alpha}}$ defined in Equation (4.16). For each $\ell'$ fixed in $\{1, \ldots, \nu_d\}$, a realization $\theta^{\ell'}$ of the nonlinear SPROM is constructed with $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{\ell'}$ and excercised to obtain the realization $\mathfrak{q}^{\ell'}_d = \mathbb{Q}(\boldsymbol{\alpha}^{\ell'}; \theta^{\ell'})$ of the stochastic error. Then, $\nu_d$ data points $\mathbf{x}^1_d, \ldots, \mathbf{x}^{\nu_d}_d$ in $\mathbb{R}^{m^0_{\boldsymbol{\alpha}}+1}$ are

introduced such that, for $\ell' = 1, \ldots, \nu_d$,

$$\mathbf{x}_d^{\ell'} = (\boldsymbol{\alpha}_d^{\ell'}, \mathbb{q}_d^{\ell'}) \in \mathcal{C}_{\boldsymbol{\alpha}}^0 \times \mathbb{R}^+ \subset \mathbb{R}^{m_{\boldsymbol{\alpha}}^0+1}.$$

**Step 2.** The second step consists in constructing a diffusion maps basis and in generating additional independent realizations. For this purpose, the following random variable is introduced

$$\mathbf{X} = (\boldsymbol{\mathcal{A}}, \mathcal{Q}),$$

defined on $(\Theta, \mathcal{T}, \mathcal{P})$, with values in $\mathcal{C}_{\boldsymbol{\alpha}}^0 \times \mathbb{R}^+ \subset \mathbb{R}^{m_{\boldsymbol{\alpha}}^0+1}$ such that $\{\mathbf{x}_d^{\ell'}, \ell' = 1, \ldots, \nu_d\}$ are $\nu_d$ independent realizations of $\mathbf{X}$. The diffusion maps basis is constructed using only the initial dataset $\{\mathbf{x}_d^{\ell'}, \ell' = 1, \ldots, \nu_d\}$. Then, $\nu_{\text{ar}} \gg \nu_d$ additional independent realizations, $\{\mathbf{x}_{\text{ar}}^{\ell}, \ell = 1, \ldots, \nu_{\text{ar}}\}$, of the random vector $\mathbf{X}$ are generated using the PLM without using the nonlinear SPROM (for further details, the reader is referred to [11] as the details of this part of the method are similar to those described in this reference). Once these additional realizations have been computed, the $\nu_{\text{ar}}$ additional independent realizations $\{\boldsymbol{\alpha}_{\text{ar}}^{\ell}, \ell = 1, \ldots, \nu_{\text{ar}}\}$ of the random variable $\boldsymbol{\mathcal{A}}$ and $\nu_{\text{ar}}$ additional independent realizations $\{\mathbb{q}_{\text{ar}}^{\ell}, \ell = 1, \ldots, \nu_{\text{ar}}\}$ of the random variable $\mathcal{Q}$ are obtained by performing the following extraction

$$(\boldsymbol{\alpha}_{\text{ar}}^{\ell}, \mathbb{q}_{\text{ar}}^{\ell}) = \mathbf{x}_{\text{ar}}^{\ell} \quad , \quad \ell = 1, \ldots, \nu_{\text{ar}}.$$

If $\nu_d$ is sufficiently large, the learning can be expected to be successful and the realizations $\{\boldsymbol{\alpha}_{\text{ar}}^{\ell}\}_\ell$ and $\{\mathbb{q}_{\text{ar}}^{\ell}\}_\ell$ will belong to $\mathcal{C}_{\boldsymbol{\alpha}}^0$ and $\mathbb{R}^+$, respectively.

**Step 3.** The third step pertains to the computation of independent realizations of the minimum statistics, represented by the positive-valued random variable $\mathcal{Q}_{\text{min}}$ on $(\Theta, \mathcal{T}, \mathcal{P})$, for the random variable $\mathcal{Q}$. Let $\nu_r$ and $\nu_e$ be two integers such that $\nu_{\text{ar}} = \nu_e \times \nu_r$. Consequently, $\nu_r = \nu_{\text{ar}}/\nu_e$. It is assumed here that $\nu_e$ and $\nu_r$ are sufficiently large – say, $\nu_e = 100$, $\nu_r = 10\,000$, and then $\nu_{\text{ar}} = 1\,000\,000$. Consider the $\nu_{\text{ar}}$ realizations $\{\boldsymbol{\alpha}_{\text{ar}}^{\ell}\}_\ell \in \mathcal{C}_{\boldsymbol{\alpha}}^0$ of $\boldsymbol{\mathcal{A}}$ and the $\nu_{\text{ar}}$ realizations $\{\mathbb{q}_{\text{ar}}^{\ell}\}_\ell \in \mathbb{R}^+$ of $\mathcal{Q}$, all of which have been computed in Step 2. For a sufficiently large value of $\nu_e$, the minimum statistics of the positive-valued random variable $\mathcal{Q}$ is typically defined as the positive-valued random variable $\mathcal{Q}_{\text{min}}$ such that $\mathcal{Q}_{\text{min}} = \min\{\mathcal{Q}^{(1)}, \ldots, \mathcal{Q}^{(\nu_e)}\}$, where $\mathcal{Q}^{(1)}, \ldots, \mathcal{Q}^{(\nu_e)}$ are $\nu_e$ independent copies of the positive-valued random variable $\mathcal{Q}$. The random variable $\mathcal{Q}_{\text{min}}$ depends on $\nu_e$. However, the dependence on $\nu_e$ is not highlighted here in order to keep the notation simple. The $\nu_r$ realizations $\{\mathbb{q}_{\text{min}}^1, \ldots, \mathbb{q}_{\text{min}}^{\nu_r}\}$ of $\mathcal{Q}_{\text{min}}$, which are assumed to be independent, are then computed as

$$\mathbb{q}_{\text{min}}^r = \min_{\ell \in \{\nu_e(r-1)+1, \ldots, r\nu_e\}} \mathbb{q}_{\text{ar}}^{\ell} \quad , \quad r = 1, \ldots, \nu_r, \tag{4.21}$$

where the minimum $\mathbb{q}_{\text{min}}^r = \mathbb{q}_{\text{ar}}^{\ell_r}$ is reached for $\ell = \ell_r \in \{\nu_e(r-1) + 1, \ldots, r\nu_e\}$. Now, let $\boldsymbol{\alpha}_{\text{min}}^r$ designate $\boldsymbol{\alpha}_{\text{ar}}^{\ell_r}$ – that is, $\boldsymbol{\alpha}_{\text{min}}^r = \boldsymbol{\alpha}_{\text{ar}}^{\ell_r}$ – and let $\boldsymbol{\mathcal{A}}_{\text{min}}$ denote the random vector associated with $\mathcal{Q}_{\text{min}}$ and for which $\{\boldsymbol{\alpha}_{\text{min}}^r, r = 1, \ldots, \nu_r\}$ are $\nu_r$ independent realizations. Note that, here, $\boldsymbol{\mathcal{A}}_{\text{min}}$ is simply a notation: in particular, it does not represent the minimum statistics of $\boldsymbol{\mathcal{A}}$. It is a random variable for which the realizations correspond to those of $\mathcal{Q}_{\text{min}}$.

**Step 4.** The last step consists in evaluating the cost function $J(\boldsymbol{\alpha}^g)$ at any point $\boldsymbol{\alpha}^g$ in $\mathcal{C}_{\boldsymbol{\alpha}}^0$ using only the additional independent realizations $\{(\boldsymbol{\alpha}_{\text{min}}^r, \mathbb{q}_{\text{min}}^r), r = 1, \ldots, \nu_r\}$ of the random variable $(\boldsymbol{\mathcal{A}}_{\text{min}}, \mathcal{Q}_{\text{min}})$. Taking into account Equations (4.19) and (4.20), the objective is to compute $E\{\mathbb{Q}_{\text{min}}(\boldsymbol{\alpha}^g)\}$ and $E\{\mathbb{Q}_{\text{min}}(\boldsymbol{\alpha}^g)^2\}$, where for $\boldsymbol{\alpha}^g$ given in $\mathcal{C}_{\boldsymbol{\alpha}}^0$, the random variable $\mathbb{Q}_{\text{min}}(\boldsymbol{\alpha}^g)$ is the minimum statistics for the random variable $\mathbb{Q}(\boldsymbol{\alpha}^g)$ (the stochastic error). For $\kappa = 1, 2$,

$$E\{(\mathbb{Q}_{\text{min}}(\boldsymbol{\alpha}^g))^\kappa\} = E\{(\mathcal{Q}_{\text{min}})^\kappa \,|\, \boldsymbol{\mathcal{A}} = \boldsymbol{\alpha}^g\}, \tag{4.22}$$

where the conditional mathematical expectation appearing in the right-hand side of Equation (4.22) is carried out using the kernel density estimation method and the additional realizations $\{(\boldsymbol{\alpha}^r_{\text{min}}, \mathbb{q}^r_{\text{min}}), r = 1, \ldots, \nu_r\}$ computed in Step 3 (the details of the computation of the conditional expectation can be found in [13]).

*REMARK 6.* There is no theoretical limit for the value of the dimension $m^0_{\boldsymbol{\alpha}}$ of the hyperparameter $\boldsymbol{\alpha}$ in the proposed PLM. However, the larger $m^0_{\boldsymbol{\alpha}}$ the larger the overall computational cost, which is true for any optimization algorithm adapted to the solution of a nonconvex problem. In any case, for the PLM to be effective, the number of data points $\nu_d$ in the initial dataset for which each point $\mathbf{x}^{\ell'}_d$ is computed using the nonlinear SPROM must be adapted to the dimension $m^0_{\boldsymbol{\alpha}}$ of the hyperparameter $\boldsymbol{\alpha}$, so that this method can actually achieve learning. In general, if $m^0_{\boldsymbol{\alpha}}$ is increased, the number of data points $\nu_d$ should also be increased.

*4.4.4. Optimization algorithms. A priori*, a grid search algorithm [17] is a candidate choice for the solution of the nonconvex optimization problem defined in Equations (4.17), (4.19), and (4.20); however, this choice would be strongly limited to the case where $m^0_{\boldsymbol{\alpha}}$ is of the order of a few units and therefore is not necessarily appropriate here. Other candidate choices are any Bayesian optimization algorithm [18], or any random search [17] or genetic algorithm [19]. In general, such algorithms tend to generate a very large number of evaluations of the cost function and therefore are relatively computationally intensive. However, thanks to the PLM introduced for evaluating the cost function, this shortcoming is avoided here. Therefore, such algorithms are appropriate in this context.

In principle, gradient-based algorithms [20] are not suitable for the solution of nonconvex optimization problems. However, they may be considered here for two reasons:

- A good estimation of the solution $\boldsymbol{\alpha}^{\text{opt}}$ is provided by the predictor $\boldsymbol{\alpha}^0$ and can be used as an advantageous initial condition. In this case, $\boldsymbol{\alpha}^{\text{opt}}$ can be searched in $\mathcal{C}^0_{\boldsymbol{\alpha}}$ defined in Equation (4.16), which is localized in a neighborhood of $\boldsymbol{\alpha}^0$.
- Thanks to the PLM, a very large number $\nu_{\text{ar}}$ of independent realizations of the stochastic error is available for constructing a good estimate of the minimum statistics (see Step 3 of the PLM described in Section 4.4.3).

These two specificities justify the use of a deterministic algorithm such as the interior-point algorithm (for example, see [21]) for solving the optimization problem defined in Equations (4.17), (4.19), and (4.20). Furthermore, it is noted that in several numerical experiments associated with applications such as those discussed in Section 5, the deterministic interior-point algorithm outperformed various genetic algorithms at solving the nonconvex optimization problem underlying the construction of the corrector for $\boldsymbol{\alpha}^0$.

*4.4.5. Initialization of the interior-point algorithm.* Because it was computed using the cost function defined by Equation (4.19), the predictor $\boldsymbol{\alpha}^0$ cannot be applied "as is" to initialize the solution of the minimization problem (4.17) formulated using the cost function defined by Equation (4.19). Instead, the initial condition of the latter problem, $\boldsymbol{\alpha}^{\text{init}}$, should be chosen in $\mathcal{C}^0_{\boldsymbol{\alpha}}$ (4.16) which, as explained in Section 4.4, accounts for $\boldsymbol{\alpha}^0$. Furthermore, since the cost function (4.19, 4.20) is formulated in terms of the minimum statistics, $\mathcal{Q}_{\text{min}}$, of the random variable $\mathcal{Q}$, $\boldsymbol{\alpha}^{\text{init}}$ can be selected by considering the corresponding $\mathcal{Q}_{\text{min}}$. Recalling that the minimum statistics are constructed from the $\nu_{\text{ar}}$ independent realizations $\{\mathbb{q}^\ell_{\text{ar}}, \ell = 1, \ldots, \nu_{\text{ar}}\}$ of $\mathcal{Q}$ associated with the $\nu_{\text{ar}}$ realizations $\{\boldsymbol{\alpha}^\ell_{\text{ar}}, \ell = 1, \ldots, \nu_{\text{ar}}\}$ of $\boldsymbol{\mathcal{A}}$, it follows that a candidate choice for $\boldsymbol{\alpha}^{\text{init}}$ could be $\boldsymbol{\alpha}^{\text{init}} = \boldsymbol{\alpha}^{\ell^{\text{init}}}_{\text{ar}}$, where $\ell^{\text{init}} = \arg\min_{\ell \in \{1,\ldots,\nu_{\text{ar}}\}} \mathbb{q}^\ell_{\text{ar}}$. Unfortunately, this minimum value is selected almost surely and is too sensitive to the statistical fluctuations. For this reason, a more appropriate choice for $\boldsymbol{\alpha}^{\text{init}}$ is to first compute $\mathbb{q}^{\text{init}}$ such that

$$\mathbb{q}^{\text{init}} = \arg \max_{\mathbb{q} \in \mathbb{R}^+} p_{\mathcal{Q}_{\text{min}}}(\mathbb{q}) \,,$$

where $\mathbb{q} \mapsto p_{\mathcal{Q}_{\text{min}}}(\mathbb{q})$ on $\mathbb{R}^+$ is the PDF of $\mathcal{Q}_{\text{min}}$, then choose $\boldsymbol{\alpha}^{\text{init}} = \boldsymbol{\alpha}_{\text{ar}}^{\ell^{\text{init}}}$, where $\ell^{\text{init}}$ is such that $\ell^{\text{init}} = \arg\min_{\ell \in \{1,\ldots,\nu_{\text{ar}}\}} |\mathbb{q}^{\text{init}} - \mathbb{q}_{ar}^{\ell}|$. The PDF $p_{\mathcal{Q}_{\text{min}}}(\mathbb{q})$ can be estimated using the Gaussian kernel-density estimation method (for example, see [22, 23]) and the $\nu_r$ independent realizations $\{\mathbb{q}_{\text{min}}^1, \ldots, \mathbb{q}_{\text{min}}^{\nu_r}\}$ computed in Step 3 of the PLM described in Section 4.4.3.

*4.4.6. Quality assessment of the corrector.* Once the corrector $\boldsymbol{\alpha}^{\text{opt}}$ has been determined by solving problem (4.17), its quality can be assessed as explained in Section 4.3 but after substituting $\boldsymbol{\alpha}^0$ by $\boldsymbol{\alpha}^{\text{opt}}$.

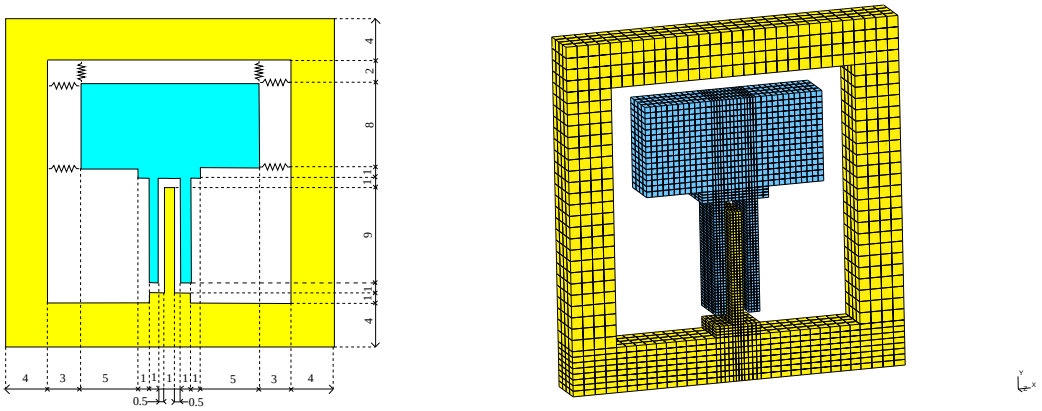
## 5. APPLICATIONS

The performance of the predictor-corrector approach outlined in Section 4 for identifying the vector-valued hyperparameter $\boldsymbol{\alpha}$ of the nonparametric probabilistic approach for modeling and quantifying model form uncertainties outlined in Section 2 is illustrated here for two 3D, nonlinear, structural dynamics problems associated with two different configurations of the same MEMS device. This device is made of two parts: a mobile part, and a suspended part attached to the mobile part. It is excited by a transient acceleration prescribed at the base of the mobile part. The two considered configurations of this device differ only by the type of nonlinearity that characterizes them. In the first considered configuration (Config-1), the structural nonlinearity is induced by nonlinear stiffnesses between the suspended and mobile parts of the device. In the second one (Config-2), the structural nonlinearity is induced by the presence of barriers that induce shocks between the mobile and suspended parts. For Config-1, the level of model-form uncertainties is relatively low; for Config-2, it is high. Hence, these two different configurations are also chosen for the additional purpose of demonstrating the performance of the nonparametric probabilistic approach for modeling and quantifying model-form uncertainties for two very different levels of uncertainty.

For both problems considered here, the solution of the underlying nonlinear PROM (2.8) is time-integrated using the midpoint rule equipped with adaptive time-stepping. This, in order to guarantee the convergence at each time-step of the fixed point method applied to the solution of the nonlinear discrete equations of motion with a relative precision of $10^{-8}$.

### *5.1. Three-dimensional MEMS device*

A schematic of the 3D MEMS device considered here is shown in Figure 1(a). The mobile part of this device is constituted of a square frame with a vertical beam attached to it. Its suspended part is constituted of a parallelepipedic solid with two attached vertical beams. The suspended part is attached to the mobile part by a 3D suspension made of 20 springs. Its geometry is described in a Cartesian coordinate system $Ox_1x_2x_3$ that is attached to its mobile part and adopted as a reference frame. The origin $O$ of this frame is located at the bottom left corner of the device. The axis $Ox_1$ is horizontal and oriented positively from left to right. The axis $Ox_2$ is vertical and oriented positively from bottom to top. The axis $Ox_3$ is perpendicular to the plane $Ox_1x_2$; it is also oriented positively from bottom to top. The MEMS itself is a cylinder of noncircular cross section whose major axis is $Ox_3$; the dimensions of its plane section are given in Figure 1(a). The external width of the square frame is $30\times10^{-6}\,m$, its external height is $31\times10^{-6}\,m$, and its depth is $4\times10^{-6}\,m$. The vertical surface at the bottom of the MEMS is described by ($x_1 \in [0\,,\, 30\times10^{-6}]$, $x_2 \in [0\,,\, 31\times10^{-6}]$, $x_3 = -4\times10^{-6}$). The suspended and mobile parts are made of a homogeneous, orthotropic, linear elastic material whose mechanical properties in the aforementioned reference frame are those of a standard $(100)$ silicon wafer [24]. Specifically, the Young moduli of this material are $E_{11} = E_{22} = 169\times10^9\ N/m^2$ and $E_{33} = 130\times10^9\ N/m^2$; its Poisson ratios are $\nu_{23} = 0.36$, $\nu_{31} = 0.28$, and $\nu_{12} = 0.064$; its shear moduli are $G_{23} = G_{31} = 79.6\times10^9\ N/m^2$ and $G_{12} = 50.9\times10^9\ N/m^2$. The mass density of the silicon material is $2\,330\ Kg/m^3$. The stiffnesses of the springs and suspension depend on the axis along which they act: $k_{s1} = 4\ N/m$ (along $Ox_1$), $k_{s2} = 6\ N/m$ (along $Ox_2$), and $k_{s3} = 1.5\ N/m$ (along $Ox_3$). Structural damping is represented




(a) 2D view of the MEMS device (lengths are shown in microns)

(b) 3D view of the FE mesh underlying the HDM

Figure 1. Schematic and FE-based HDM of a 3D MEMS device with a mobile part (yellow) and a suspended part (blue) attached to the mobile part via a suspension system.

using the global damping model described in Appendix A, which enables the control of the modal damping rate of the structural model in the frequency band of analysis.

As stated above, two different configurations of this device are considered here. They differ only by the type of nonlinearity introduced between the two vertical beams of the suspended part and the vertical beam of the mobile part:

*Config-1*. In this configuration, a nonlinear elastic material is inserted between the aforementioned beams. Its constitutive equation corresponds to a cubic, elastic, restoring force whose elastic constant is $k_b = 2\times10^{12}\ N/m$.

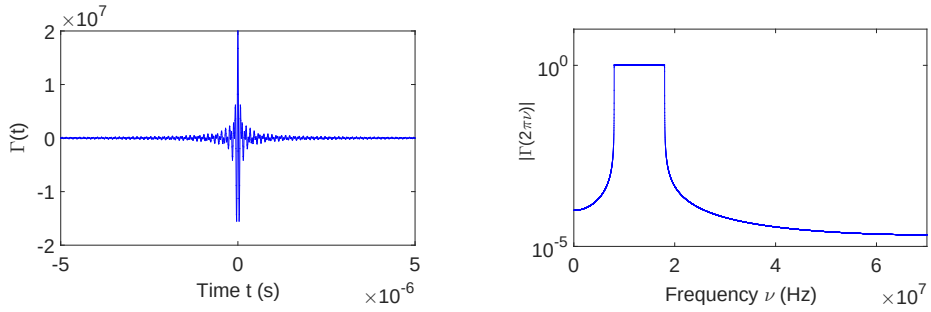
*Config-2*. In this alternative configuration, two continuous elastic barriers are introduced at the left and right sides of the vertical beam attached to the mobile part. The constitutive equation of each elastic barrier is linear and characterized by the elastic constant $k_b = 2\times10^{12}\ N/m$. The horizontal gap between the left (right) beam of the suspended part and the beam of the mobile part is denoted by $\varepsilon_L$ ($\varepsilon_R$). Initially, $\varepsilon_L = \varepsilon_R = 0.5\times10^{-6}\,m$.

For both configurations, zero $x_1$-, $x_2$-, and $x_3$-displacement boundary conditions with respect to the reference frame $Ox_1x_2x_3$ are prescribed at the base of the mobile part of the device – specifically, at the points $\{x_1 \in [0\,,\,30\times10^{-6}]\,,\,x_2 = 0\,,\,x_3 \in [0\,,\,-4\times10^{-6}]\}$. The following time-dependent, square integrable, and real-valued $x_1$-acceleration is also prescribed at these same points

$$\Gamma(t) = \frac{\Gamma_0}{\pi\,t}\,\{\sin(t(\omega_c + \Delta\omega_c/2)) - \sin(t(\omega_c - \Delta\omega_c/2)\} \quad , \quad t \in [t_0, T]\,,$$

where $\Gamma_0 = 120\ m/s^2$, $\omega_c = 2\,\pi\times13\times10^6\ rad/s$ is the central angular frequency, and $\Delta\omega_c = 2\,\pi\times10\times10^6\ rad/s$ is the angular frequency bandwidth. This acceleration is graphically depicted in Figure 2(a). The graph of $\omega \mapsto |\widehat{\Gamma}(\omega)|$ defined on $2\,\pi\times[0\,,\,70\times10^6]\ rad/s$, where $\widehat{\Gamma}(\omega) = \displaystyle\int_{t_0}^{T} e^{-i\omega t}\Gamma(t)\,dt$, is plotted in Figure 2(b). The energy of the excitation signal is mainly concentrated in the frequency band $[-\omega_e, -\omega_{\min}] \cup [\omega_{\min}, \omega_e]$, where $\omega_{\min} = \omega_c - \Delta\omega_c/2 = 2\pi\times8\times10^6\ rad/s$ and $\omega_e = \omega_c + \Delta\omega_c/2 = 2\pi\times18\times10^6\,rad/s$.

At time $t_0$, the device is at rest (its displacement and velocity fields are zero). In all analyses, $t_0 < 0$ is written as $t_0 = -m_0\,\pi/\omega_e = -2.7778\times10^{-5}\ s$, where $m_0 = 1\,000$, and the time-interval of analysis is set to $[t_0, 1.8559\times10^{-4}\ s]$. The latter setting is chosen because at $T = 1.8559\times10^{-4}\ s$, the device is returned to its zero equilibrium with a small relative error. The frequency band of observation is chosen as $\mathcal{B}_o = [0, \omega_o]$, where $\omega_o = 2\,\pi\times70\times10^6\ rad/s$.




(a) Time-dependent acceleration $t \mapsto \Gamma(t)$ (b) Frequency-dependent acceleration $\nu \mapsto \widehat{\Gamma}(2\pi\,\nu)$

Figure 2. Prescribed acceleration at the base of the mobile part of the MEMS device. (a): graph of $\Gamma(t)$ defined in $[-2.78\times10^{-5}\,,\,1.86\times10^{-4}]\ s$ and represented in $[-5\times10^{-6}\,,\,5\times10^{-6}]\ s$. (b): graph of $|\widehat{\Gamma}(2\pi\nu)|$ (in the $\log_{10}$-scale) defined in $[-72\times10^{6}\,,\,72\times10^{6}]\ Hz$ and represented in the frequency band of observation $[0\,,\,70\times10^{6}]\ Hz$.

For both configurations described above, the $x_1$-acceleration of the response of the system is observed at $744$ spatial points distributed across various locations of the MEMS device, in particular on the boundaries of the vertical beams of the suspended and mobile parts. Hence, the dimension of each of the observation vector $\mathbf{o}(t)$ and its frequency domain representation $\widehat{\mathbf{o}}(\omega)$ is $m_{\mathbf{o}} = 744$. All of these observations are used to identify the hyperparameter $\boldsymbol{\alpha}$ using the predictor-corrector approach presented in Section 4. However, in order to keep the number of figures within a reasonable limit, two observations are selected among all $744$ possible ones and denoted by $\text{Obs}_1^1$ and $\text{Obs}_6^1$ for Config-1, and $\text{Obs}_1^2$ and $\text{Obs}_4^2$ for Config-2.

If the dynamical system associated with either considered configuration was linear, the energy of the response signal would be concentrated in the same frequency band $[\omega_{\text{min}}, \omega_e]$ as that of the energy of the excitation signal (see Figure 2(b)). Due to the nonlinearity however, part of the energy of the excitation signal is transferred outside its frequency band and consequently, the frequency band of the response is not $[\omega_{\text{min}}, \omega_e]$ but $[0, \omega_{\text{max}}]$, where $\omega_{\text{max}} > \omega_e$ is to be determined. To this end: the sampling time-step is chosen as $\Delta t = \pi/\omega_{\text{max}} = 6.9444\times10^{-9}\ s$, where $\omega_{\text{max}} = 4\,\omega_e = 2\pi\times72\times10^{6}\ rad/s$; and the frequency band of analysis is set to $[-\omega_{\text{max}}, \omega_{\text{max}}]$, where $\omega_e < \omega_{\text{max}}$. This yields $n_{\text{time}} = 30\,725$ time points in the time-interval $[t_0, T]$. The sampling frequency step is set to $\Delta\omega = 2\pi\times4686.7\ rad/s$, yielding also $n_{\text{freq}} = 30\,725$ frequency points in the frequency band of analysis $[-\omega_{\text{max}}, \omega_{\text{max}}]$.

### *5.2. Finite element based nonlinear high-dimensional models*

First, a linear, FE-based HDM is constructed for the neutral configuration of the 3D MEMS device defined here as the configuration where no material or barriers are inserted between the two vertical beams of the suspended part of this device, and the vertical beam of its mobile part. This HDM has $7\,328$ 8-noded solid elements, $10\,675$ nodes, and $N = 32\,025$ dofs (see Figure 1(b)). There are $205$ of these nodes, which belong to the base of the mobile part of the device: at each of these nodes, all displacement dofs are constrained to zero in the moving reference frame $Ox_1x_2x_3$, due to the boundary conditions described in Section 5.2. Hence, $N_{\text{CD}} = 615$. Next, two nonlinear variants of this HDM are generated to account for the nonlinear specificities of Config-1 and Config-2. Both variants have the same dimension. Furthermore, this dimension is the same as that of the linear HDM as both nonlinear variants can be constructed without generating additional dofs. For this reason, for the sake of simplicity, and unless otherwise specified, HDM is used to refer to the linear HDM associated with the linear, neutral configuration of the device, or to the nonlinear HDM of the same dimension associated with Config-1 or Config-2. In all cases, the displacement vector $\mathbf{y}(t)$ associated with the HDM is measured in the aforementioned reference frame: therefore, it represents

a relative displacement vector. The HDM is used to analyze the dynamic response of the neutral configuration, configuration Config-1, or configuration Config-2 of the device to the prescribed excitation $\Gamma(t)$. In all cases, the associated governing equations for the relative displacement vector, $\mathbf{y}(t)$, are those given in Equations (2.1) to (2.3), where:

- The vector of external forces $\mathbf{f}(t)$ is a function of the mass matrix $[M]$, the $x_1$-dofs where zero relative displacement boundary conditions are applied, and the relative acceleration $\Gamma(t)$.
- The vector of internal forces $\mathbf{g}(\mathbf{y}(t), \dot{\mathbf{y}}(t))$ can be written as

$$\mathbf{g}(\mathbf{y}(t), \dot{\mathbf{y}}(t)) = [\mathbb{D}]\, \dot{\mathbf{y}}(t) + [K]\, \mathbf{y}(t) + \mathbf{f}_{\text{NL}}(\mathbf{y}(t))\,, \tag{5.1}$$

  where:

  - As stated in Section5.1, the damping matrix $[\mathbb{D}]$ is constructed using the global damping model described in Appendix A, which enables the control of the modal damping rate of the modeled device in the frequency band of analysis.
  - The stiffness matrix $[K]$ belongs to $\mathbb{M}_N^{+0}$ and has a null space of dimension $6$ before applying the zero boundary conditions.
  - The nonlinear vector function $\mathbf{f}_{\text{NL}}(\mathbf{y}(t))$ is constructed to account for the structural nonlinearity introduced in Config-1 or Config-2, as applies.
- The initial conditions are $\mathbf{y}(t_0) = \dot{\mathbf{y}}(t_0) = \mathbf{0}$.
- The matrix $[B] \in \mathbb{M}_{N,N_{\text{CD}}}$ represents the $N_{\text{CD}}$ zero boundary conditions and is constructed such that $[B]^T[B] = [I_{N_{\text{CD}}}]$.

### *5.3. Projection-based reduced-order models*

First, a deterministic ROB $[V] \in \mathbb{M}_{N,n}$ of dimension $n \ll N$ is constructed using the classical method of snapshot collection and compression. For each of Config-1 and Config-2, the nonlinear HDM is exercised to compute the time-sampled response $\{\mathbf{y}(j\,\Delta t), j = 1, \ldots, n_{\text{time}}\}$. The solution snapshots $\{\mathbf{y}^k_{\text{snp}}, k = 1, \ldots, n_{\text{snp}}\}$, where $\mathbf{y}^k_{\text{snp}} = \mathbf{y}(k\,\mu_{\text{snp}}\,\Delta t)$, $\mu_{\text{snp}}$ denotes the snapshot-step, and $n_{\text{snp}} = n_{\text{time}}/\mu_{\text{snp}}$, are extracted from this response. Next, the thin singular value decomposition of the matrix $[\mathbf{y}^1_{\text{snp}} \ldots \mathbf{y}^{n_{\text{snp}}}_{\text{snp}}]$ is computed, and the singular values $\mathbb{s}_1 \geq \ldots \geq \mathbb{s}_{n_{\text{snp}}}$ are ordered in descending order. Then, $[V]$ is defined as the matrix whose columns are the $n$ left orthonormal vectors associated with the first $n$ ordered singular values $\mathbb{s}_1 \geq \ldots \geq \mathbb{s}_n$, where $n$ is determined from the convergence rate of the compression measured by $\sum_{k=1}^{n} \mathbb{s}_k^2 / \sum_{k=1}^{n_{\text{snp}}} \mathbb{s}_k^2$.

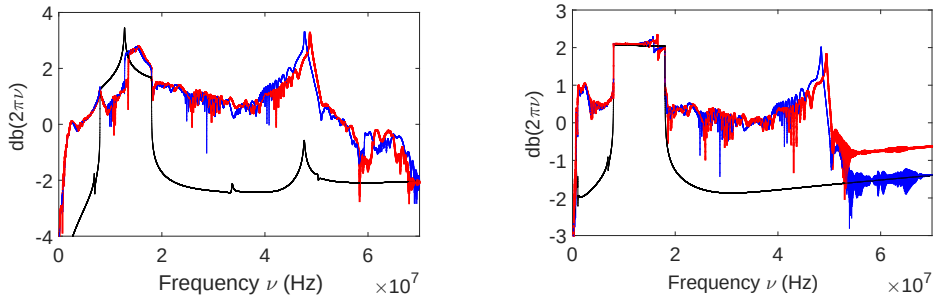
For $n_{\text{time}} = 30\,725$ and $\mu_{\text{snp}} = 10$ ($n_{\text{snp}} = 3\,072$), the aforementioned convergence rate analysis yields $n = 8$ for Config-1 and $n = 20$ for Config-2 (note that in both cases, a ROB based on the natural modes of the linear, neutral configuration of the device requires $n_{\text{mode}} = 40$ modes for convergence and therefore is less efficient).

Next, each constructed ROB $[V] \in \mathbb{M}_{N,n}$ is randomized following the procedure given by Equations (2.17–2.22) to obtain the counterpart SROB $[\mathbf{W}]$. Then, the corresponding nonlinear PROM and SPROM are constructed by Galerkin projection of the nonlinear HDM as in Equations (2.7–2.9) and (2.13–2.15), respectively.

### *5.4. Model-form uncertainty quantification*

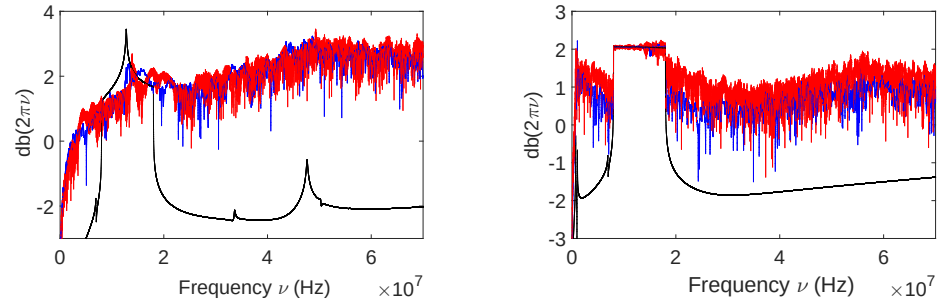
For each nonlinear PROM constructed in Section 5.3, a modified version is designed to generate targets for both selected observations in lieu of experimental data. Specifically, each modified nonlinear PROM constructed for Config-1 or Config-2 is designed to generate model-form uncertainties.

Figure 3 (Config-1) and Figure 4 (Config-2) display the graphs of the function $\nu \mapsto \text{db}^{\text{target}}(2\pi\nu)$ of the observed targets, the graphs of the function $\nu \mapsto \text{db}^{(n)}(2\pi\nu)$ of the counterpart values predicted using the nonlinear PROM, and the graphs of the function $\nu \mapsto \text{db}^{(n)}_{\text{lin}}(2\pi\nu)$ of the counterpart values using the deterministic PROM. The reader can observe that for both configurations:




(a) Observation $\text{Obs}_1^1$ (b) Observation $\text{Obs}_6^1$

Figure 3. Config-1: graphs of the function $\nu \mapsto \text{db}^{\text{target}}(2\pi\nu)$ of the observed targets (red thick lines), graphs of the function $\nu \mapsto \text{db}^{(n)}(2\pi\nu)$ of the counterpart results predicted using the nonlinear PROM (blue mid solid lines), and graphs of the function $\nu \mapsto \text{db}_{\text{lin}}^{(n)}(2\pi\nu)$ of the counterpart results obtained using the linear HDM associated with the linear, neutral configuration of the device (black thin line).

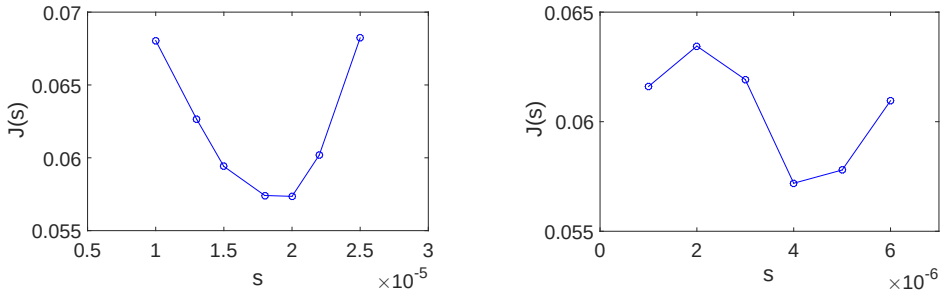



(a) Observation $\text{Obs}_1^2$ (b) Observation $\text{Obs}_4^2$

Figure 4. Config-2: graphs of the function $\nu \mapsto \text{db}^{\text{target}}(2\pi\nu)$ of the observed targets (red thick lines), graphs of the function $\nu \mapsto \text{db}^{(n)}(2\pi\nu)$ of the counterpart results predicted using the nonlinear PROM (blue mid solid lines), and graphs of the function $\nu \mapsto \text{db}_{\text{lin}}^{(n)}(2\pi\nu)$ of the counterpart results obtained using the linear HDM associated with the linear, neutral configuration of the device (black thin line).

- The effect of the structural nonlinearity is very important outside the frequency band $[\omega_{\min}, \omega_e]$ of the excitation, as there are large differences between the numerical predictions obtained using the nonlinear PROM and those obtained using the linear HDM associated with the neutral, linear configuration of the device.
- The model-form uncertainties – which can be analyzed by estimating the difference between the target and numerically predicted values – are not homogeneous neither as a function of the frequency, nor as a function of the selected observation.
- The effects of the model-form uncertainties are small in the frequency band $[\omega_{\min}, \omega_e]$ of the excitation, but can be very large outside this band.
- For observations $\text{Obs}_1^1$ and $\text{Obs}_2^1$, the effects of the model-form uncertainties are smaller than for the observations $\text{Obs}_6^1$ and $\text{Obs}_4^2$.

*5.4.1. Identification of the hyperparameter via the predictor.* For each of Config-1 and Config-2, the predictor $\boldsymbol{\alpha}^0$ for the hyperparameter $\boldsymbol{\alpha}$ is computed using the method presented in Section 4.2. For Config-1, $n = 8$ and therefore the number of hyperparameters associated with the matrix $[\sigma]$ is




(a) Config-1 (b) Config-2

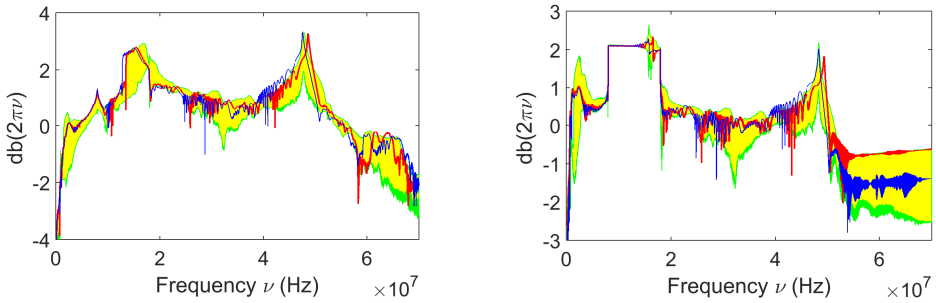
Figure 5. Graph of the function $s \mapsto J_{\text{freq}}(s)$ for the grid used by the grid search algorithm.

$m_\sigma = 36$. For Config-2, $n = 20$ and therefore $m_\sigma = 210$. Specifically, **Stage 1** is used to compute the predictor $[\sigma^0]$: it leads to $m_\sigma^0 = 4$ for Config-1 and to $m_\sigma^0 = 13$ for Config-2. Hence, it achieves a significant reduction in the number of hyperparameters associated with the matrix $[\sigma^0]$. In **Stage 2**, the optimization problem defined in Equations (4.7) and (4.8) is solved using the grid search algorithm and the grid $\{1.0, 1.3, 1.5, 1.8, 2.0, 2.2, 2.5\} \times 10^{-5}$ with $m_s = 7$ points for Config-1, and the grid $\{1.0, 2.0, 3.0, 4.0, 5.0, 6.0\} \times 10^{-6}$ with $m_s = 6$ points for Config-2. For each value of $\boldsymbol{\alpha} = (s, \beta^0, [\sigma^0])$, the nonlinear SPROM is solved for $\nu_s = 100$ independent realizations.

Figure 5(a) and Figure 5(b) show the graph of the function $s \mapsto J_{\text{freq}}(s)$ for Config-1 and Config-2, respectively. The predictor $s^0$ for $s$ is equal to $2.0 \times 10^{-5}$ for Config-1 and to $4.0 \times 10^{-6}$ for Config-2.

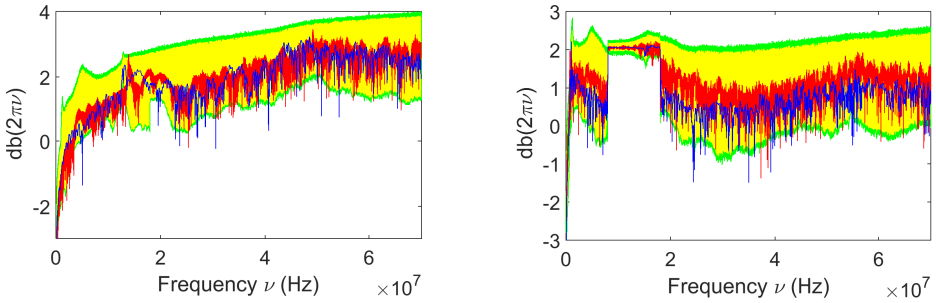
The quality of the computed predictor $\boldsymbol{\alpha}^0$ is assessed by plotting each target and the corresponding confidence region constructed using the nonlinear SPROM for $\boldsymbol{\alpha} = \boldsymbol{\alpha}^0$ and computed using Equation (4.15) with $p_c = 0.98$. For each selected observation, Figure 6 (Config-1) and Figure 7 (Config-2) display the graph of the function $\nu \mapsto \text{db}^{\text{target}}(2\pi\nu)$ of the observed target, the graph of the function $\nu \mapsto \text{db}^{(n)}(2\pi\nu)$ of the observation predicted using the nonlinear PROM, and the graph of the confidence region of the random function $\nu \mapsto \text{dB}^{(n)}(2\pi\nu)$ of the observation computed using the nonlinear SPROM. A quality criterion is that the target belongs to the confidence region with a probability of $0.98$. The reader can observer that for this criterion, the predictor delivers a reasonable performance. Nevertheless, Figure 6 shows that the peak of the target at $48.76 \times 10^6$ $Hz$ does not belong to the confidence region and Figure 7 shows that the upper envelope of the confidence region is somehow high. For these reasons, the corrector for $\boldsymbol{\alpha}^0$ is considered next.

*5.4.2. Identification of the hyperparameter via the corrector.* For both Config-1 and Config-2, the parameters $w_J$ and $\gamma_J$ of the cost function $J(\boldsymbol{\alpha})$ defined by Equation (4.19) are set here to $w_J = 0.5$ and $\gamma_J = 1.0$. A convergence analysis of the PLM is performed with respect to the dimension $\nu_d$ of the initial dataset computed in its Step 1, using the following values of $\nu_d$: 100, 300, 500, 1 000, 1 500, and 2 000. For each of these, the convergence of the quality indicator $\mathcal{I}$ is analyzed with respect to the number of additional realizations $\nu_{\text{ar}}$ using the following set of values for this variable: $\{200\,000, 1\,000\,000, 2\,100\,000\}$. For each value of $\nu_d$ in this set, the value of $\mathcal{I}$ using $\nu_{\text{ar}} = 1\,000\,000$ is found to be close to its counterpart computed using $\nu_{\text{ar}} = 2\,100\,000$, which suggests that convergence is reached for $\nu_{\text{ar}} = 1\,000\,000$. Still, the convergence analysis with respect to $\nu_d$ and the QA of the corrector are performed using the larger value of $\nu_{\text{ar}} = 2\,100\,000$. To this end, the parameters introduced in Equation (4.21) are chosen such that $\nu_e = 100$ and $\nu_r = 21\,000$. The results of the convergence analysis of the PLM with respect to $\nu_d$ are reported in Table I. They show that for Config-1, the corrector improves the value of the quality indicator obtained for the predictor as soon as $\nu_d \geq 1\,000$; the best value is reached for $\nu_d = 2\,000$. For Config-2, the corrector improves the value of the quality indicator obtained for the predictor as soon as $\nu_d \geq 1\,500$, and the best value of $\mathcal{I}$ is obtained for $\nu_d = 1\,500$.




(a) Observation $\text{Obs}_1^1$ (b) Observation $\text{Obs}_6^1$

Figure 6. QA of the predictor for Config-1: graphs of the function $\nu \mapsto \text{db}^{\text{target}}(2\pi\nu)$ of the observed target (red thick line), graphs of the function $\nu \mapsto \text{db}^{(n)}(2\pi\nu)$ of the observation computed using the nonlinear PROM (blue mid solid line), and graphs of the confidence region of the random function $\nu \mapsto \text{dB}^{(n)}(2\pi\nu)$ of the observation predicted using the nonlinear SPROM (yellow region with green lines for the upper and lower envelopes).



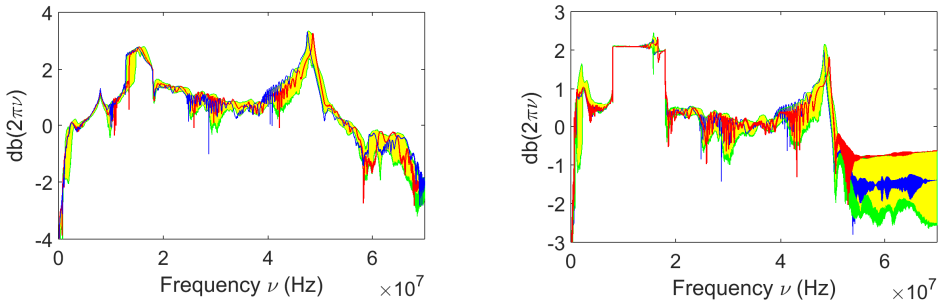

(a) Observation $\text{Obs}_1^2$ (b) Observation $\text{Obs}_4^2$

Figure 7. QA of the predictor for Config-2: graphs of the function $\nu \mapsto \text{db}^{\text{target}}(2\pi\nu)$ of the observed target (red thick line), graphs of the function $\nu \mapsto \text{db}^{(n)}(2\pi\nu)$ of the observation computed using the nonlinear PROM (blue mid solid line), and graphs of the confidence region of the random function $\nu \mapsto \text{dB}^{(n)}(2\pi\nu)$ of the observation predicted using the nonlinear SPROM (yellow region with green lines for the upper and lower envelopes).

Table I. Convergence analysis of the PLM with respect to $\nu_d$.

| | **Predictor** | **Corrector** | | | | | |
|---|---|---|---|---|---|---|---|
| $\nu_d$ | – | 100 | 300 | 500 | 1 000 | 1 500 | 2 000 |
| Config-1: $\mathcal{I}$ | 3.18 | – | – | 3.96 | 3.02 | 3.07 | 2.89 |
| Config-2: $\mathcal{I}$ | 2.01 | 4.30 | 4.60 | 2.80 | 4.21 | 1.90 | 1.83 |

The quality of the corrector for $\boldsymbol{\alpha}$ is further assessed by plotting each target and the corresponding confidence region constructed using the nonlinear SPROM for $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{\text{opt}}$ and computed using Equation (4.15) with $p_c = 0.98$. Attention is focused on the results obtained when the PLM is converged with respect to $\nu_d$ – that is, for $\nu_d = 2\,000$ for Config-1 and $\nu_d = 1\,500$ for Config-2



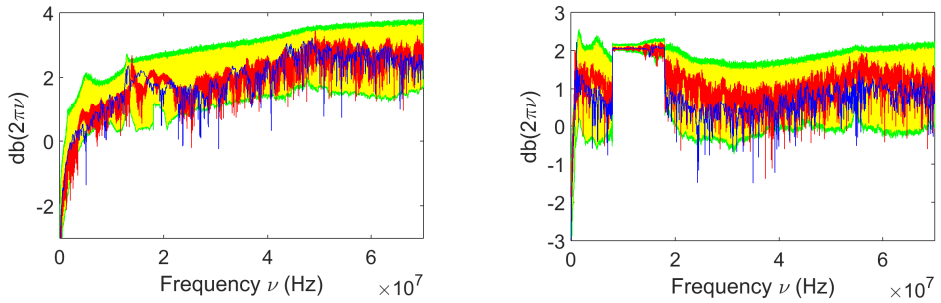

(a) Observation $\text{Obs}_1^1$ (b) : Observation $\text{Obs}_6^1$

Figure 8. QA of the corrector for Config-1: graphs of the function $\nu \mapsto \text{db}^{\text{target}}(2\pi\nu)$ of the observed target (red thick line), graphs of the function $\nu \mapsto \text{db}^{(n)}(2\pi\nu)$ of the observation computed using the nonlinear PROM (blue mid solid line), and graphs of the confidence region of the random function $\nu \mapsto \text{dB}^{(n)}(2\pi\nu)$ of the observation predicted using the nonlinear SPROM (yellow region with green lines for the upper and lower envelopes).

– and with respect to the number of additional realizations – that is, for $\nu_{\text{ar}} = 2\,100\,000$ for both configurations. The optimal values of the hyperparameters $s$ and $\beta$ of $\boldsymbol{\alpha}^{\text{opt}}$ are found to be $s^{\text{opt}} = 1.01 \times 10^{-5}$ and $\beta^{\text{opt}} = 0.026$ for Config-1, and $s^{\text{opt}} = 3.56 \times 10^{-6}$ and $\beta^{\text{opt}} = 0.016$ for Config-2. Figure 8 (Config-1) and Figure 9 (Config-2) display for each selected observation the graph of the function $\nu \mapsto \text{db}^{\text{target}}(2\pi\nu)$ of the observed target, the graph of the function $\nu \mapsto \text{db}^{(n)}(2\pi\nu)$ of the observation computed using the nonlinear PROM, and the graph of the confidence region of the random function $\nu \mapsto \text{dB}^{(n)}(2\pi\nu)$ of the observation predicted using the nonlinear SPROM. Again, a quality criterion is that the target belongs to the confidence region with a probability of $0.98$. Comparing these figures with Figures 6 and 7 shows that the corrector for $\boldsymbol{\alpha}$ improves the confidence region obtained using the predictor for this vector-valued hyperparameter. In particular, Figure 8 shows that the peak of the target at $48.76 \times 10^6\ Hz$ is now inside the computed confidence region and Figure 9 shows that the upper envelope of the computed confidence region is now lower. Figure 8 also shows that for some frequencies, the response of the nonlinear PROM is outside the computed confidence region, but the target is always well within the computed confidence region. These results highlight the ability of the overall nonparametric probabilistic method for modeling and quantifying model-form uncertainties to reach a given target.

*5.4.3. Wall-clock time performance analysis.* All computations described above were performed on a Linux cluster in double precision arithmetic. The main steps of the model-form UQ analysis were performed in parallel execution mode on 70 cores of this cluster as described below:

- Predictor (**Stage 2**): this stage was performed with $m_s = 6$ for Config-1 and $m_s = 7$ for Config-2; for both configurations, the nonlinear SPROM was exercised for $\nu_s = 100$ independent realizations.
- Predictor (QA): for this purpose, the nonlinear SPROM was exercised for $\nu_{\text{sim}} = 1\,000$ independent realizations.
- Corrector (Step 1 PLM): Step 1 of the PLM for constructing the initial dataset was performed with $\nu_d = 2\,000$ for Config-1 and $\nu_d = 1\,500$ for Config-2.
- Corrector (Step 2 PLM): Step 1 of the PLM for computing additional realizations was performed with $\nu_{\text{ar}} = 1\,050 \times \nu_d$ for Config-1 and $\nu_{\text{ar}} = 1\,400 \times \nu_d$ for Config-2.
- Corrector (Step 3 & Step 4 PLM): the interior point algorithm converged in $19$ iterations during which it performed $141$ evaluations of the cost function (4.19) in the case of Config-1; it converged in $58$ iterations during which it performed $950$ evaluations of the cost function in the case of Config-2.




(a) Observation $\text{Obs}_1^2$ (b) Observation $\text{Obs}_4^2$

Figure 9. QA of the corrector for Config-2: graphs of the function $\nu \mapsto \text{db}^{\text{target}}(2\pi\nu)$ of the observed target (red thick line), graphs of the function $\nu \mapsto \text{db}^{(n)}(2\pi\nu)$ of the observation computed using the nonlinear PROM (blue mid solid line), and graphs of the confidence region of the random function $\nu \mapsto \text{dB}^{(n)}(2\pi\nu)$ of the observation predicted using the nonlinear SPROM (yellow region with green lines for the upper and lower envelopes).

– Corrector (QA): for this purpose, the nonlinear SPROM was exercised for $\nu_{\text{sim}} = 1\,000$ independent realizations.

The wall-clock time elapsed in the execution of each of these main steps and the total wall-clock time are reported in Table II for each of Config-1 and Config-2. Two observations are noteworthy:

- In each case, the total wall-clock time of the predictor-corrector approach is dominated by the computational cost of the corrector component: specifically, 94% of the total wall-clock time in the case of Config-1 and 80% in the case of Config-2 is due to the corrector component.
- The model-form UQ analysis of Config-2 is more computationally intensive than that of Config-1, due to the presence of shocks in the second configuration. However, most of the difference between the computational costs of both UQ analyses is due to the predictor component for which the wall-clock time for Config-2 is roughly 4.6 times higher than for Config-1.

Table II. Wall-clock time in seconds (and in hours between parentheses) on 70 cores of a Linux cluster for the main steps of the model-form UQ analysis.

| **Main step** | **Config-1** | **Config-2** |
|---|---|---|
| Predictor (**Stage 2**) | 934 | 3 991 |
| Predictor (QA) | 185 | 1 134 |
| Corrector (Step 1 PLM) | 1 850 | 1 833 |
| Corrector (Step 2 PLM) | 16 018 | 17 243 |
| Corrector (Step 3 & Step 4 PLM) | 11 | 13 |
| Corrector (QA) | 178 | 1 448 |
| **Total (Predictor + Corrector)** | 19 176 **(5.33 hrs)** | 25 662 **(7.13 hrs)** |

As shown above, the solution of the optimization problem (4.17) – that is, the computation of the corrector $\boldsymbol{\alpha}^{\text{opt}}$ – dominates the total cost of the identification of the vector-valued hyperparameter $\boldsymbol{\alpha}$. The total cost of this identification itself dominates the overall computational cost of the

nonparameteric probabilistic method for model-form UQ analysis at the heart of this paper. The computer implementation of the standard solution of problem (4.17) can be described as an overall algorithm with three main loops: an outer-loop where each instance is associated with one iteration of the optimizer and incurs a few evaluations of the cost function to be minimized; an inner-loop for performing the $\nu_{\text{sim}}$ independent realizations (here $\nu_{\text{sim}} = 1\,000$); and an innermost-loop for performing the SPROM-based simulation in the time domain. On the other hand, once training has been performed, the computer implementation of the solution of problem (4.17) using the PLM described in this paper is organized around two main loops only: the same aforementioned outer-loop, and the same aforementioned innermost-loop. Hence, the gain in computational speed enabled by the proposed predictor-corrector approach for identifying $\boldsymbol{\alpha}$ can be easily estimated using the information given above and in Table II. To this end, it is first noted that $\nu_{\text{sim}} = 1\,000$ executions of the nonlinear SPROM in the time domain using 70 cores and the algorithmic parameters described in the introduction of Section 5 and in Section 5.1 require on average 181 seconds wall-clock time for Config-1, and 1 448 seconds for Config-2. It follows that a good estimate of the wall-clock time on 70 cores for the standard approach for determining $\boldsymbol{\alpha}^{\text{opt}}$ is $141 \times 181 = 25\,521$ seconds for Config-1, and $950 \times 1\,448 = 1\,375\,600$ seconds for Config-2. Therefore, the speedup factors enabled by the PLM proposed in this paper are equal to $1.33$ in the case of Config-1, and $53.61$ in the case of Config-2 where the effect of the nonlinearity on the solution of the optimization problem (4.17) is much stronger. In any case, it is emphasized here that these speedup factors are only estimates as the analysis given here assumes that the number of outer-loop instances associated with the interior point algorithm is the same in the presence and absence of the PLM.

## 6. SUMMARY AND CONCLUSIONS

A novel, predictor-corrector approach for identifying the vector-valued hyperparameter of a nonparametric probabilistic method for modeling and quantifying model-form uncertainties is presented in this paper. Its key component is a probabilistic learning on manifolds for constructing a low-cost, surrogate model for the Monte Carlo simulations of interest. Specifically, this model enables the economical generation, within the solution of an inverse statistical problem, of stochastic realizations that are concentrated on a manifold identified using an initial set of data points. The computational advantages of this proposed approach are demonstrated in this paper for the model-form uncertainty quantification analysis, using the aforementioned nonparametric probabilistic method, of two three-dimensional, nonlinear, structural dynamics problems associated with two different configurations of a MEMS device. For this application, a speedup factor – with respect to a standard approach for solving an inverse statistical problem – as high as 56 is achieved.

## ACKNOWLEDGEMENTS

The second author acknowledges partial support by the Army Research Laboratory through the Army High Performance Computing Research Center under Cooperative Agreement W911NF-07-2-0027, partial support by The Boeing Company under Contract Sponsor Ref. 45047, and partial support by DARPA under the Enabling Quantification of Uncertainty in Physical Systems (EQUiPS) program. This document does not necessarily reflect the position of these institutions, and no official endorsement should be inferred.

## APPENDIX A: CONSTRUCTION OF THE DAMPING MATRICES $[\mathbb{D}]$

The structural damping model chosen for constructing the matrix $[\mathbb{D}]$ of the FE-based HDMs of the 3D MEMS device described in Section 5 is based on the neutral configuration of this device defined in Section 5.2 – that is, the configuration where no material or barriers are inserted between the two vertical beams of the suspended part of this device and the vertical beam of its mobile part.

Hence, this model leads to the same matrix $[\mathbb{D}]$ for both Config-1 and Config-2. It corresponds to the following adaptation of the structural damping representation presented in [25] (Chapter VI, Equation (45)).

Let $[\Phi] = [\boldsymbol{\varphi}^1 \dots \boldsymbol{\varphi}^{n_{\text{mode}}}]$ denote the matrix of the first $n_{\text{mode}}$ natural vibration modes associated with the first $n_{\text{mode}}$ lowest eigenvalues $0 < \lambda_1 \leq \dots \leq \lambda_{n_{\text{mode}}}$ of the undamped neutral configuration of the 3D MEMS device. In the time domain, the dynamics of this undamped configuration are governed by the linear equations of motion $[M]\,\ddot{\mathbf{y}}(t) + [K]\,\mathbf{y}(t) = \mathbf{0}$. Then, the deterministic matrix $[\mathbb{D}] \in \mathbb{M}_N^+$ is constructed here as follows:

$$[\mathbb{D}] = \sum_{k=1}^{n_{\text{mode}}} 2\,\xi_d\,(\sqrt{\lambda_k} - \sqrt{\lambda_{n_{\text{mode}}}})\,[M]\,\boldsymbol{\varphi}^k\,([M]\,\boldsymbol{\varphi}^k)^T + 2\,\xi_d\,\sqrt{\lambda_{n_{\text{mode}}}}\,[M]\,. \tag{A.1}$$

The reader can verify that the above definition of $[\mathbb{D}]$ satisfies

$$[\Phi]^T[\mathbb{D}]\,[\Phi] = [\mathcal{D}]\,,$$

where the generalized damping matrix $[\mathcal{D}] \in \mathbb{M}_{n_{\text{mode}}}^+$ is diagonal and its diagonal entries are $[\mathcal{D}]_{kk} = 2\,\xi_d\,\sqrt{\lambda_k}$. For $\{k > n_{\text{mode}}\,, k' > n_{\text{mode}}\}$, this matrix also satisfies $(\boldsymbol{\varphi}^{k'})^T[\mathbb{D}]\boldsymbol{\varphi}^k = 2\,\xi_d\,\sqrt{\lambda_{n_{\text{mode}}}}\delta_{kk'}$. For $\{k \leq n_{\text{mode}}\,, k' > n_{\text{mode}}\}$ and for $\{k > n_{\text{mode}}\,, k' \leq n_{\text{mode}}\}$, it satisfies $(\boldsymbol{\varphi}^{k'})^T[\mathbb{D}]\boldsymbol{\varphi}^k = 0$. It follows that:

- The structural damping model defined by Equation (A.1) yields a constant damping rate $\xi_d$ for the first $n_{\text{mode}}$ natural modes of the system, and yields the damping rates $\{\xi_d\,\sqrt{\lambda_{n_{\text{mode}}}}/\sqrt{\lambda_{n_{\text{mode}}+1}}, \dots, \xi_d\,\sqrt{\lambda_{n_{\text{mode}}}}/\sqrt{\lambda_N}\}$ for the $N - n_{\text{mode}}$ higher natural modes $\{\boldsymbol{\varphi}^{n_{\text{mode}}+1}, \dots, \boldsymbol{\varphi}^N\}$.
- This model is a global damping model that enables the control of the modal damping rate in the frequency band of analysis.

By construction, $n_{\text{mode}}$ is chosen such that $n_{\text{mode}} \geq n$. Then, the deterministic and stochastic reduced-order damping matrices

$$[\mathbb{D}^{(n)}] = [V]^T[\mathbb{D}]\,[V] \quad \text{and} \quad [\mathbf{D}^{(n)}] = [\mathbf{W}]^T[\mathbb{D}]\,[\mathbf{W}]$$

are nonsingular. They can also be written as

$$[\mathbb{D}^{(n)}] = \sum_{k=1}^{n_{\text{mode}}} 2\,\xi_d\,(\sqrt{\lambda_k} - \sqrt{\lambda_{n_{\text{mode}}}})\,\chi^k(\chi^k)^T + 2\,\xi_d\,\sqrt{\lambda_{n_{\text{mode}}}}\,[V]^T[M]\,[V]\,,$$

where $\chi^k = [V]^T[M]\,\boldsymbol{\varphi}^k$ is a deterministic vector with values in $\mathbb{R}^n$, and

$$[\mathbf{D}^{(n)}] = \sum_{k=1}^{n_{\text{mode}}} 2\,\xi_d\,(\sqrt{\lambda_k} - \sqrt{\lambda_{n_{\text{mode}}}})\,\boldsymbol{\Psi}^k(\boldsymbol{\Psi}^k)^T + 2\,\xi_d\,\sqrt{\lambda_{n_{\text{mode}}}}\,[\mathbf{W}]^T[M]\,[\mathbf{W}]\,,$$

where $\boldsymbol{\Psi}^k = [\mathbf{W}]^T[M]\,\boldsymbol{\varphi}^k$ is a random vector with values in $\mathbb{R}^n$.

## REFERENCES


1. Paul-Dubois-Taine A, Amsallem D. An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models. *International Journal for Numerical Methods in Engineering* 2015; **102**(5):1262-1292.
2. Xu, P. Truncated SVD methods for discrete linear ill-posed problems. *Geophysical Journal International* 1998; **135**:505-514.
3. Farhat C, Chapman T, Avery P. Structure-preserving, stability, and accuracy properties of the Energy-Conserving Sampling and Weighting (ECSW) method for the hyper reduction of nonlinear finite element dynamic models. *International Journal for Numerical Methods in Engineering* 2015; **102**(5):1077-1110.
4. Soize C, Farhat C. A nonparametric probabilistic approach for quantifying uncertainties in low- and high-dimensional nonlinear models. *International Journal for Numerical Methods in Engineering* 2017; **109**(6):837-888.

5. Farhat C, Bos A, Avery P, Soize C. Modeling and quantification of model-form uncertainties in eigenvalue computations using a stochastic reduced model. *AIAA Journal* 2018; **56**:1198-1210.
6. Farhat C, Avery P, Chapman T, Cortial J. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering* 2014; **98**(9):625-662.
7. Carlberg K, Bou-Mosleh C, Farhat C. Efficient nonlinear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering* 2011; **86**:155-181.
8. Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics* 2013; **242**:623-647.
9. Gupta A, Seiler P, Danowsky, B. Ground vibration tests on a flexible flying wing aircraft. *AIAA-2016-1753*, AIAA SciTech 2016, San Diego, CA, January 4-8, 2016.
10. Farhat C, Bos A, Tezaur R, Chapman T, Avery P, Soize C. A stochastic projection-based hyperreduced order model for model-form uncertainties in vibration analysis. *AIAA-2018-1410*, AIAA SciTech 2018, Kissimmee, Florida, January 8-12, 2018.
11. Soize C, Ghanem R. Data-driven probability concentration and sampling on manifold. *Journal of Computational Physics* 2016; **3**21:242-258.
12. Soize C, Ghanem R. Polynomial chaos representation of databases on manifolds. *Journal of Computational Physics*, 2017; **3**35:201-221.
13. Ghanem R, Soize C. Probabilistic nonconvex constrained optimization with fixed number of function evaluations. *International Journal for Numerical Methods in Engineering* 2018; **1**13(4):719-741.
14. Spall JC. *Introduction to Stochastic Search and Optimization*. John Wiley and Sons: Hoboken, New Jersey, 2003.
15. Serfling RJ. *Approximation Theorems of Mathematical Statistics*. John Wiley and Sons: Hoboken, New Jersey, 2002.
16. Soize C. *Uncertainty Quantification. An Accelerated Course with Advanced Applications in Computational Engineering*. Interdisciplinary Applied Mathematics. Springer: New York, 2017.
17. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 2012; **13**:281-305.
18. Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems* 2012; **25**:2960-2968.
19. Goldberg DE. *Genetic Algorithms in Search. Optimization & Machine Learning*. Addison-Wesley, Boston, 1989.
20. Bengio Y. Gradient-based optimization of hyperparameters. *Neural Computation* 2000; **12**(8):1889-1900.
21. Byrd RH, Hribar ME, Nocedal J. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization* 1999, **9**(4):877-900.
22. Bowman AW, Azzalini A. *Applied Smoothing Techniques for Data Analysis*. Oxford University Press: Oxford, UK, 1997.
23. Scott DW. *Multivariate Density Estimation: Theory, Practice, and Visualization, Second Edition*. John Wiley and Sons: Hoboken, New Jersey, 2015.
24. Hopcroft MA. What is the Young's modulus of silicon. *Journal of Microelectromechanical systems* 2010, **19(**2):229-238.
25. Ohayon R, Soize C. *Structural Acoustics and Vibration*. Academic Press: San Diego, London, 1998.