frontiers | Frontiers in Materials

ORIGINAL RESEARCH
published: 24 June 2022
doi: 10.3389/fmats.2022.904707

Parametric Curves Metamodelling Based on Data Clustering, Data Alignment, POD-Based Modes Extraction and PGD-Based Nonlinear Regressions

Victor Champaney[1]*, *Angelo Pasquale*[1,2], *Amine Ammar*[2,3] *and Francisco Chinesta*[1,3,4]

[1]ESI Group Chair @ PIMM Lab, ENSAM Institute of Technology, Paris, France, [2]ESI Group Chair @ LAMPA Lab, ENSAM Institute of Technology, Paris, France, [3]CNRS@CREATE Ltd, Singapore, Singapore, [4]ESI Group, Paris, France

In the context of parametric surrogates, several nontrivial issues arise when a whole curve shall be predicted from given input features. For instance, different sampling or ending points lead to non-aligned curves. This also happens when the curves exhibit a common pattern characterized by critical points at shifted locations (e.g., in mechanics, the elastic-plastic transition or the rupture point for a material). In such cases, classical interpolation methods fail in giving physics-consistent results and appropriate pre-processing steps are required. Moreover, when bifurcations occur into the parametric space, to enhance the accuracy of the surrogate, a coupling with clustering and classification algorithms is needed. In this work we present several methodologies to overcome these issues. We also exploit such surrogates to quantify and propagate uncertainty, furnishing parametric stastistical bounds for the predicted curves. The procedures are exemplified over two problems in Computational Mechanics.

Keywords: parametric curves, data-driven modeling, uncertainty quantification and propagation, POD, PGD

OPEN ACCESS

Edited by:
Chady Ghnatios,
Notre Dame University, Lebanon

Reviewed by:
Attilio Frangi,
Politecnico di Milano, Italy
Yongxing Shen,
Shanghai Jiao Tong University, China
Nicolas Montes,
Universidad CEU Cardenal Herrera, Spain

Correspondence:
Victor Champaney
victor.champaney@ensam.eu

Specialty section:
This article was submitted to Computational Materials Science, a section of the journal Frontiers in Materials

Received: *25 March 2022*
Accepted: *11 May 2022*
Published: *24 June 2022*

Citation:
Champaney V, Pasquale A, Ammar A and Chinesta F (2022) Parametric Curves Metamodelling Based on Data Clustering, Data Alignment, POD-Based Modes Extraction and PGD-Based Nonlinear Regressions. Front. Mater. 9:904707. doi: 10.3389/fmats.2022.904707

1 INTRODUCTION

In a large variety of engineering applications, parametric surrogates are thoroughly powerful tools (Simpson et al., 2001; Prud'homme et al., 2002; Audouze et al., 2013; Mainini and Willcox, 2015; Hesthaven et al., 2016; Benner et al., 2020a). They allow a real-time monitoring and control of the most relevant physical quantities describing a given phenomenon. Moreover, they empower smart decision making, optimizing time and manufacturing costs. The uncertainty propagation in such models is also fundamental to operate efficiently in diagnosis and prognosis. In a non-intrusive framework, given an engineering problem, a Design of Experiments–DoE–based on the problem parameters is established and the corresponding responses of the system are collected into databases, which are used as training data to build the surrogate model via Machine Learning–ML–and Model Order Reduction–MOR–algorithms (Wang and Shan, 2007; Benner et al., 2015; Hesthaven and Ubbiali, 2018; Rajaram et al., 2020; Franchini et al., 2022; Khatouri et al., 2022). Such responses are usually the ensemble of several Quantities of Interest–QoI–observed, for instance, over time (i.e., time series) and can come both from experiments and numerical simulations. Therefore, each QoI is usually a curve, discretized according to the number of sampling points. This is the case when, for example, a material is tested and the force-displacement curve is extracted for different parameters $\mathbf{p}$ defining the material itself. It is also the case when a sensor placed on a mould records the pressure evolution during the mould filling from a resin injected into a mould. In this paper, we propose several strategies to build parametric curves, illustrating the procedure over two applications in computational solid mechanics.

The target quantities representing the system response are univariate functions, depending on d features (parameters), that is $g(x;\mathbf{p}) : X \to \mathbb{R}$, where $\mathbf{p} = (p^1, \ldots, p^d) \in \Omega \subset \mathbb{R}^d$, while $X \subset \mathbb{R}$. The parametric surrogate f^X takes as input a new combination of parameters $\mathbf{p} \in \Omega$ and returns an approximation $\tilde{g}(x;\mathbf{p})$ of $g(x;\mathbf{p})$, that is:

$$\begin{aligned} f^X :&\Omega \to \mathcal{G} \\ &\mathbf{p} \mapsto \tilde{g}(x;\mathbf{p}) : X \to \mathbb{R}, \end{aligned}$$

where $\mathcal{G}$ is a given functional space (in most engineering applications, $\mathcal{G} \subseteq L^2(X)$).

Our procedure mainly consists in the application of non-intrusive nonlinear regressions based on the sparse Proper Generalized Decomposition–sPGD– (Chinesta et al., 2011; Borzacchiello et al., 2017; Ibáñez et al., 2018; Sancarlos et al., 2021), these being efficient under the scarce data availability constraint. Indeed, in real engineering applications, when dealing with simulation-based metamodels, data availability is largely limited by the complexity of the Finite Element–FE–computations. From the High-Fidelity–HiFi–offline simulations, it is often possible to define a Reduced Order Model–ROM–, for instance, by extracting the most relevant Proper Orthogonal Decomposition–POD–modes from the training data (Raghavan et al., 2013; Fareed and Singler, 2019). Consequently, since the curve can be expressed into the extracted POD reduced basis through a set of weighting coefficients, the nonlinear regressions can be applied to predict such coefficients. A similar workflow is applied by Gooijer et al. (2021), where the POD-based surrogate models employ Radial Basis Function–RBF–interpolations. For the sake of completeness, it shall be noticed that the use of POD-based interpolations–PODI–has several limitations and drawbacks, particularly when dealing with non-linear solution manifolds. To alleviate such issues, several works have been conducted in the framework of interpolations on Grassmann manifolds and its tangent space, improving the model robustness over the parametric space (Amsallem and Farhat, 2008; Mosquera et al., 2018, 2021; Friderikos et al., 2020, 2022).

However, ad-hoc physics-based data pre-processing is a fundamental step to be embedded in the procedure. Indeed, when different choices of the parameters carry radically different physical behaviours, the interpolation in the parametric space can lead to nonphysical solutions. In such cases, separate regression sub-models are built, requiring the coupling with some clustering and classification algorithms, leading to the so-called multi-regression strategy.

Another non-trivial issue comes when the curves exhibit a common pattern characterized by some critical points resulting from a change in the physical behaviour. Indeed, a shift among the locations of such critical points in the different curves would cause nonphysical results when employing a classical interpolation. To overcome this matter, we propose a parametrization of the curves accounting for the locations of such critical points and allowing a curve alignment prior to the interpolation.

The main points addressed in this work are:

1. the parametric modeling of a quantity of interest using advanced sparse nonlinear regressions;
2. the parametric modeling of a curve where a data alignement is needed;
3. the statistical parametric modeling based on a parametrized physical model;
4. the statistical parametric model learned from scarce data (measurements);
5. and, finally, the concept of data clustering to overcome bifurcations in the parametric space.

The paper is structured as follows. **Section 2** is mainly a review of some well-known techniques, excepting **SubSection 2.3** which illustrates the implementation of the sPGD algorithm for the prediction of functions defined over an interval (i.e., curves). Elements of novelty are introduced in **Section 3** and 4, where 1) we propose a curve alignment prior to regression; 2) we define a statistical model for uncertainty propagation, furnishing confidence bounds for a parametric curve; 3) we employ a multi-regression, based on clustering and classification, to tackle bifurcations in the parametric space, enhancing the model accuracy. We exemplify the methodologies over two engineering applications in computational solid mechanics. The first application concerns a reduced order model for virtual materials characterized by a parametric Krupkowski hardening law; the second application is related to crack propagation analysis in parametric notched specimens under tensile loading. **Section 5** is a short conclusion, in which possible further developments and approaches are discussed.

2 METHODS

In this section we briefly summarize the main tools in MOR employed in this work. For a complete description of the most recent advances in the MOR community, we refer to the handbooks by Benner et al. (2020c,b,a) and the plentiful bibliography therein.

2.1 POD

We assume that a numerical approximation of the unknown field of interest $u(\mathbf{x}, t)$ is known at the nodes $\mathbf{x}_i$ of a spatial mesh for discrete times $t_j = (j-1)\Delta t$, with $i \in [1, \ldots, n_x]$ and $j \in [1, \ldots, n_t]$. We use the notation $u(\mathbf{x}_i, t_j) \equiv u^j(\mathbf{x}_i) \equiv u_i^j$ and define $\mathbf{u}^j$ as the vector of nodal values u_i^j at time t_j. The main objective of the POD is to obtain the most typical or characteristic structure $\phi(\mathbf{x})$ among these $u^j(\mathbf{x})$, $\forall j$. For this purpose, we maximize the scalar quantity

$$\lambda = \frac{\sum_{j=1}^{n_t} \left[\sum_{i=1}^{n_x} \phi(\mathbf{x}_i) u^j(\mathbf{x}_i) \right]^2}{\sum_{i=1}^{n_x} (\phi(\mathbf{x}_i))^2},$$

which leads to the following eigenvalue problem $\mathbf{C}\boldsymbol{\phi} = \lambda\boldsymbol{\phi}$, where

$$C_{kl} = \sum_{j=1}^{n_t} u^j(\mathbf{x}_k) u^j(\mathbf{x}_l), \ \mathbf{C} = \sum_{j=1}^{n_t} \mathbf{u}^j (\mathbf{u}^j)^T$$

is the two-point correlation matrix (symmetric and positive definite). Defining the matrix

$$\mathbf{Q} = \begin{bmatrix} \mathbf{u}^1 & \mathbf{u}^2 & \cdots & \mathbf{u}^{n_t} \end{bmatrix}$$

we have $\mathbf{C} = \mathbf{Q}\mathbf{Q}^T$.

In order to obtain a reduced-order model, we first solve the eigenvalue problem and select the r eigenvectors $\boldsymbol{\phi}_i$ associated with the highest eigenvalues (truncated SVD at rank r), with in practice $r \ll n_x$. Thus r eigenvectors are placed in the columns of a matrix $\mathbf{B}$ that allows reducing $\mathbf{U}$ into its reduced counterpart $\boldsymbol{\gamma}$, according to $\mathbf{U} = \mathbf{B}\boldsymbol{\gamma}$. Then, considering the full-size system $\mathbf{KU} = \mathbf{F}$, we have $\mathbf{KB}\boldsymbol{\gamma} = \mathbf{F}$. Premultiplying by $\mathbf{B}^T$ one gets $\mathbf{B}^T\mathbf{KB}\boldsymbol{\gamma} = \mathbf{B}^T\mathbf{F}$ and, with new definitions, the reduced counterpart becomes $\mathbf{k}\boldsymbol{\gamma} = \mathbf{f}$.

The main drawback related to such a procedure is the size of the eigenproblem to be solved, the size of the correlation matrix $\mathbf{C}$, $n_x \times n_x$, with n_x scaling with the number of nodes considered in the problem discretization that can reach in some applications millions and much more. The so-called Snapshot-POD allows alleviating the just referred issue (Hilberg et al., 1994). The basic concept is that, when $n_t \ll n_x$, it is much more convenient solving the eigenvalue problem for $\tilde{\mathbf{C}} = \mathbf{Q}^T\mathbf{Q}$, whose size scales with n_t, then retrieve the modes related to the highest eigenvalues.

2.2 PODI

The origin of the non-intrusive POD, comes from the so-called POD with interpolation. PODI consider different snapshots related with different values of the model parameter p, $\mathbf{U}(p_i)$, $i = 1, \ldots, n_s$, without loss of generality assumed scalar and ordered, i.e. $p_1 < \ldots < p_{n_s}$.

Then, as usual in POD-based MOR, the reduced basis is extracted, $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_r$. Now, for a given parameter p, with $p_1 < p < p_{n_s}$ and $p \neq \{p_1, p_2, \ldots, p_{n_s}\}$, instead of expressing the searched solution into the reduced basis $\mathbf{U}(p) = \sum_{i=1}^{r} \gamma_i(p)\boldsymbol{\phi}_i$, and then looking for the coefficient $\gamma_i(p)$ by Galerkin projection, i.e., by solving $(\mathbf{B}^T\mathbf{KB})\boldsymbol{\gamma} = \mathbf{B}^T\mathbf{F}$ (that requires assembling the matrix and performing the matrix products before finally solving the reduced linear system of equations), PODI proceeds as follows.

- *Sampling*: $\mathbf{U}(p_i) \equiv \mathbf{U}_i$, $i = 1, \ldots, n_s$;
- *Reduced basis extraction*: POD is applied to extract the reduced basis $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_r$;
- *Reproduction*: calculation of $\boldsymbol{\gamma}_i$. For that, we look to express $\mathbf{U}_i = \sum_{j=1}^{r} \gamma_j^i \boldsymbol{\phi}_j$. Premultiplying by $\boldsymbol{\phi}_k$ and taking into account the orthonormality of the reduced basis, it results

$$\boldsymbol{\phi}_k^T \mathbf{U}_i = \gamma_k^i.$$

Repeating for all $i \in \{1, \ldots, n_s\}$ and $k \in \{1, \ldots, r\}$, we obtain $\boldsymbol{\gamma}_i$ (the reduced counterpart of $\mathbf{U}_i$).

- *Interpolation*: with the reduced solution representations $\boldsymbol{\gamma}_i \equiv \boldsymbol{\gamma}(p = p_i)$, one is tempted for any other p to proceed by interpolation, i.e.

$$\boldsymbol{\gamma}(p) = \sum_{i=1}^{r} \boldsymbol{\gamma}_i \mathcal{F}_i(p),$$

with $\mathcal{F}_i(p)$ the approximation functions, that define an interpolation as soon as $\mathcal{F}_i(p_j) = \delta_{ij}$, with δ_{ij} the Kronecker delta.

- *Reconstruction*: with $\boldsymbol{\gamma}(p)$ obtained, the solution can be reconstructed everywhere from the nodal values $\mathbf{U}(p) = \mathbf{B}\boldsymbol{\gamma}(p)$.

2.2.1 Extension to Multi-Parametric Settings

The just discussed procedure seems very appealing, however, its extension to highly-multidimensional settings remains difficult because of usual approximation bases suffer from the so-called curse of dimensionality.

In the case of moderate dimensionality, the PODI algorithm is easily generalizable. For that purpose we first reformulate the PODI described above as follows: the reconstruction $\mathbf{U}(p) = \mathbf{B}\boldsymbol{\gamma}(p)$ can be expressed in the equivalent form:

$$\mathbf{U}(p) = \sum_{k=1}^{r} \gamma_k(p) \boldsymbol{\phi}_k;$$

with $\gamma_k^i \equiv \gamma_k(p_i)$ known, the interpolation can be expressed as:

$$\mathbf{U}(p) = \sum_{k=1}^{r} \left(\sum_{i=1}^{n_s} \gamma_k^i \mathcal{F}_i(p) \right) \boldsymbol{\phi}_k,$$

that is directly generalizable to the multi-parametric setting where the scalar p is replaced by the parameters vector $\mathbf{p}$, with the interpolation expressed now as

$$\mathbf{U}(\mathbf{p}) = \sum_{k=1}^{r} \left(\sum_{i=1}^{n_s} \gamma_k^i \mathcal{F}_i(\mathbf{p}) \right) \boldsymbol{\phi}_k.$$

As previously mentioned the main difficulty associated with the technique just described is the difficulty of interpolating when the number of parameters (the size of vector $\mathbf{p}$) increases too much. Separated representations in sparse settings, addressed in **Subsection 2.3**, succeed in circumventing the just referred difficulty.

2.3 Advanced Sparse PGD-Based Nonlinear Regressions

Here we discuss the PGD-based regression methods to build metamodels depending on d features. In particular, we focus on the case where, for a given choice of the parameters.

1. a single-valued output is measured;
2. a vector-valued output is measured;
3. a single-valued output is measured over a certain interval.

2.3.1 Single-Valued Output

In the case of a scalar output, the general problems consists of constructing the function

$$f\left(p^1,\ldots,p^d\right): \Omega \subset \mathbb{R}^d \to \mathbb{R},$$

that depends on d features (parameters) p^k, $k = 1,\ldots,d$, taking values in the parametric space Ω, where a sparse sample of n_s points and the corresponding outputs are known.

The so-called sparse PGD (sPGD) expresses the function f from a low-rank separated representation

$$f\left(p^1,\ldots,p^d\right) \approx \tilde{f}^M\left(p^1,\ldots,p^d\right) = \sum_{m=1}^{M}\prod_{k=1}^{d}\psi_m^k\left(p^k\right), \tag{1}$$

constructed from rank-one updates within a greedy constructor. In the previous expression $\tilde{f}^M$ refers to the approximation, M the number of employed modes (sums) and ψ_m^k are the one-dimensional functions concerning the mode m and the dimension k.

Functions ψ_m^k, $m = 1,\ldots,M$ and $k = 1,\ldots,d$ are expressed from a standard approximation basis $\mathbf{N}_m^k$, via coefficients $\mathbf{a}_m^k$:

$$\psi_m^k\left(p^k\right) = \sum_{j=1}^{D} N_{j,m}^k\left(p^k\right) a_{j,m}^k = \left(\mathbf{N}_m^k\right)^T \mathbf{a}_m^k, \tag{2}$$

where D represents the number of degrees of freedom (nodes) of the chosen approximation and $\mathbf{N}_m^k$ is the vector collecting the shape functions.

In the context of usual regression the approximation $\tilde{f}^M$ results from

$$\tilde{f}^M = \arg\min_{f^*}\|f - f^*\|_2^2 = \arg\min_{f^*}\sum_{i=1}^{n_s}\left|f\left(\mathbf{p}_i\right) - f^*\left(\mathbf{p}_i\right)\right|^2, \tag{3}$$

where $\tilde{f}^M$ takes the separated form of **Eq. 1**, n_s is the number of sampling points to train the model and $\mathbf{p}_i$ the vectors that contain the input data points of the training set. Notice that, to avoid overfitting, the number of basis functions D must be $D < n_s$.

The approximation coefficients of each one-dimensional function are computed by employing a greedy algorithm, such that, once the approximation up to order $M-1$ is known, the Mth order term reads

$$\tilde{f}^M = \sum_{m=1}^{M-1}\prod_{k=1}^{d}\psi_m^k\left(p^k\right) + \prod_{k=1}^{d}\psi_M^k\left(p^k\right).$$

The computed function is expected to approximate f not only in the training set but in any point $\mathbf{p} \in \Omega$.

The main issue is how to ally rich approximations and scarce available data, while avoiding overfitting. For that purpose a modal adaptivity strategy–MAS–was associated to the sPGD, however, it has been observed that the desired accuracy is not achieved before reaching overfitting or the algorithm stops too early when using MAS in some cases. This last issue implies a PGD solution composed of low order approximation functions, thus not getting an as rich as desired function. Some papers describing the just referred techniques are (Borzacchiello et al., 2017; Ibáñez et al., 2018).

In addition, in problems where just a few terms of the interpolation basis are present (that is, there are just some sparse non-zero elements in the interpolation basis to be determined), the strategy fails in recognizing the true model and therefore lacks accuracy.

To solve these difficulties, different regularizations were proposed (Sancarlos et al., 2021), combining L^2 and L^1 norms affecting the coefficients $\mathbf{a}_m^k$, in order to increase the predictive performances beyond the sPGD capabilities, or to construct parsimonious models while improving predictive performances.

2.3.2 Vector-Valued Output

In the case of a multidimensional output, we seek the function

$$\mathbf{f}\left(p^1,\ldots,p^d\right) = \begin{bmatrix} f_1\left(p^1,\ldots,p^d\right) \\ f_2\left(p^1,\ldots,p^d\right) \\ \vdots \\ f_n\left(p^1,\ldots,p^d\right) \end{bmatrix} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^n.$$

This is a trivial extension of the single-valued function, where each component $f_i(p^1,\ldots,p^d)$, for $i = 1,\ldots,n$, is fitted independently using the procedures explained in **Subsection 2.3.1**.

2.3.3 Single-Valued Output Over an Interval

Let us now consider the case when, d features (parameters), the system output is a univariate function of the variable x, that is $g(x;\mathbf{p}) : X \to \mathbb{R}$, where $\mathbf{p} = (p^1,\ldots,p^d) \in \Omega \subset \mathbb{R}^d$, while $X \subset \mathbb{R}$. The parametric surrogate f^X takes as input a new combination of parameters $\mathbf{p} \in \Omega$ and returns an approximation $\tilde{g}(x;\mathbf{p})$ of $g(x;\mathbf{p})$, that is:

$$\begin{aligned} f^X &: \Omega \to \mathcal{G} \\ &\mathbf{p} \mapsto \tilde{g}\left(x;\mathbf{p}\right) : X \to \mathbb{R}, \end{aligned}$$

where $\mathcal{G}$ is a given functional space.

Usually, the target function $g(x)$ is evaluated (known) in a finite number n_x of sampling points, that is the discrete ensemble $X = \{x_j\}_{j=1}^{n_x}$.

In this case, the coordinate x can be considered as an additional parameter, and the approximation problem can be reformulated as seeking the function

$$f\left(\mathbf{p},p^{d+1}\right) : \Xi \subset \mathbb{R}^{d+1} \to \mathbb{R}.$$

We have dropped the subscript X related to the variable x since the approximation problem has been recast into a new parametric framework defined by Ξ. The newly defined parametric coordinate p^{d+1} accounts for the location in which $g(x)$ shall be approximated, that is:

$$f\left(\mathbf{p},p^{d+1}\right) = \tilde{g}\left(p^{d+1}\right) \approx g\left(p^{d+1};\mathbf{p}\right).$$

Such coordinate is thus much richer than the others, given the very fine discretization in n_x points available along this direction, compared to the sparse knowledge concerning the first d parametric coordinates belonging to Ω.

Equation 1 now reads:

$$f\left(p^1,\ldots,p^d,p^{d+1}\right) \approx \tilde{f}^M\left(p^1,\ldots,p^d,p^{d+1}\right) = \sum_{m=1}^{M}\prod_{k=1}^{d+1}\psi_m^k\left(p^k\right),$$

where the univariate functions of the first d parameters $\{\psi_m^k\}_{k=1}^d$, for $m=1,\ldots,M$, are still expressed by the same polynomial basis, as defined in **Eq. 2**. However, functions ψ_m^{d+1} can be expressed through standard piecewise linear basis functions (i.e., Lagrangian hat functions), defined over the n_x discretization points of the coordinate x:

$$\psi_m^{d+1}\left(p^{d+1}\right) = \sum_{j=1}^{n_x} N_{j,m}^{d+1}\left(p^{d+1}\right)a_{j,m}^{d+1} = \left(\mathbf{N}_m^{d+1}\right)^T\mathbf{a}_m^{d+1}$$

where n_x is the number of discretization and

$$N_{j,m}^{d+1}(x) = \begin{cases} 0, & x < x_{j-1} \\ \left(x - x_{j-1}\right)/h, & x_{j-1} \le x < x_j \\ 1 - \left(x - x_j\right)/h, & x_j \le x < x_{j+1} \\ 0, & x \ge x_{j+1}, \end{cases}$$

with h denoting an uniform discretization step. In particular,

$$N_{j,m}^{d+1}\left(x_i\right) = \begin{cases} 1, & i = j \\ 0, & i \ne j. \end{cases}$$

The minimization problem **(Eq. 3)** can also be recast as

$$\tilde{f}^M = \arg\min_{f^*}\sum_{j=1}^{n_x}\sum_{i=1}^{n_s}\left|f\left(\mathbf{p}_i,p_j^{d+1}\right) - f^*\left(\mathbf{p}_i,p_j^{d+1}\right)\right|^2.$$

With these definitions made, the algorithm runs as previously explained.

2.3.3.1 POD Modes Extraction

Here we reformulate the approximation problem of curves within a POD-based MOR builder, which can be seen as a data pre-compression and dimensionality reduction approach. Indeed, considering the training data $\{g_i(x)\}_{i=1}^{n_s}$, for $x \in X = \{x_j\}_{j=1}^{n_x}$, the following snapshots matrix can be built:

$$\mathbf{S} = \begin{bmatrix}\mathbf{g}_1 & \mathbf{g}_2 & \cdots & \mathbf{g}_{n_s}\end{bmatrix} \in \mathbb{R}^{n_x \times n_s},$$

where $\mathbf{g} \in \mathbb{R}^{n_x \times 1}$ contains the evaluations of $g(x)$ over the discrete ensemble X.

A reduced factorization of the snapshots matrix is then obtained via a standard truncated POD of rank r:

$$\mathbf{S} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

where $\mathbf{U} \in \mathbb{R}^{n_x \times r}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{r \times r}$, $\mathbf{V} \in \mathbb{R}^{n_s \times r}$. From these, we can define the matrices of POD modes and coefficients, respectively:

$$\boldsymbol{\Phi} := \mathbf{U} = \begin{bmatrix}\boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \cdots & \boldsymbol{\phi}_r\end{bmatrix},\ \boldsymbol{\Lambda} := \mathbf{V}\boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_2 & \cdots & \boldsymbol{\lambda}_r\end{bmatrix}$$

In particular, the matrix $\boldsymbol{\Phi}$ contains, by columns, the functions of the reduced POD basis $\{\phi_i(x)\}_{i=1}^r$ evaluated at points in X, while $\boldsymbol{\Lambda}$ collects the projection coefficients into the reduced basis. A generic curve $g_k(x)$ belonging to the training dataset, for $k = 1,\ldots,n_s$ and with $x \in X$, has the reduced counterpart

$$g_k^{(r)}(x) = \sum_{i=1}^{r}\lambda_{k,i}\phi_i(x), \tag{4}$$

and, in particular, its discrete form reads

$$\mathbf{g}_k^{(r)} = \boldsymbol{\Lambda}_{k,\bullet}\boldsymbol{\Phi}^T,$$

where $\boldsymbol{\Lambda}_{k,}\bullet$ denotes the kth row of the matrix $\boldsymbol{\Lambda}$.

Let us consider now a parametric curve depending on d features $\overline{\mathbf{p}} \in \Omega$, that is $g(x;\overline{\mathbf{p}})$, for $x \in X$. From **Eq. 4** it is clear that, once the reduced basis matrix $\boldsymbol{\Phi}$ available, such function is projected over this basis only through the POD (parametric) coefficients $\{\lambda_i(\mathbf{p})\}_{i=1}^r$:

$$g^{(r)}\left(x;\overline{\mathbf{p}}\right) = \sum_{i=1}^{r}\lambda_i\left(\overline{\mathbf{p}}\right)\phi_i(x).$$

The above equation suggests that a reduced-order parametric metamodel for the curves can be built considering only the set of coefficients $\{\lambda_i(\mathbf{p})\}_{i=1}^r$. In particular, the following parametric function shall be constructed:

$$\mathbf{f}(\mathbf{p}) = \begin{bmatrix}\lambda_1(\mathbf{p}) \\ \lambda_2(\mathbf{p}) \\ \vdots \\ \lambda_r(\mathbf{p})\end{bmatrix} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^r,$$

from the available training dataset $\{\mathbf{p}_k, \boldsymbol{\Lambda}_{k,\bullet} = (\lambda_{k,1},\lambda_{k,2},\ldots,\lambda_{k,r})\}_{k=1}^{n_s}$ obtained after the POD. This problem can be solved by the algorithm exposed in **Subsection 2.3.2**.

2.4 Multi-Regression

Creating a unique regression in large physical and parametric domains is a tricky issue. From one side, constructing a regression of a quantity of interest is much more accurate than creating the parametric curve (e.g., the parametric time evolution of the solution at a certain point), that in turn, becomes much more accurate than creating a regression of a field. The reason is that in general regressions are constructed by using the L^2-norm, and consequently, if a given field exhibits strong localizations, these local behaviors are sacrificed in benefit of a quite good solution everywhere (on average).

Thus, a valuable route for enhancing accuracy consists in partitioning the physical space, in order to perform a regression in each of the resulting patches. Local quasi-linear regressions perform in general better than rich nonlinear regressions in the whole space domain.

The main issue in using multiple regressions, one per patch, is that the continuity can be lost on the patch boundaries. One could try to enforce the continuity, for example within a Partition of Unity–PU–framework, however, continuity is not compulsory, and then, on the patch borders (or in its neighborhood) one could compute the regressions from both sides and average them. Another possibility is taking profit of those discontinuities for refinement purposes, as usually considered within the finite element method framework.

In the case of parametric models the issue that we just discussed not only affects the spatial domain, but also the parametric one. In that case, making a partition of the multi-parametric space is not simple. One possibility consists in clustering the solutions related to the considered sampling, for example by invoking the *k-means*. Then, a nonlinear regression is created from the solutions in each cluster. Finally, the trickiest issue becomes the way of associating a cluster to any parameters choice, that is, performing an accurate classification. The procedure can be summarized in the following steps:

1. clustering high-fidelity solutions related to a design of experiments;
2. creating a regression model in each cluster (for instance, via the algorithms presented in **Subsection 2.3**);
3. constructing a classifier able to associate a cluster to any parameters choice and to select the most suitable regression model.

2.5 *k*-Means

k-means is one of the earliest methods for non-supervised vector quantization in artificial intelligence (MacQueen, 1967). In essence, as the Support Vector Machines–SVMs– (Cristianini and Shawe-Taylor, 2000) would do in the context of supervised learning models, *k*-means performs *cluster analysis*. In other words, this technique groups a set of objects such that every member of the group or *cluster* is more similar (closer) to the other members of the cluster than to any member of the rest of clusters.

In the case of *k*-means, this partition is made on the basis that each experimental data pertains to the cluster with the nearest mean. As can be readily noticed, this is equivalent to computing Voronoi cells in the data. Formally, if we have a set of observations in the form of high-dimensional vectors $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)$, we aim at partitioning these M observations into k sets $(k \leq M)$, $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$, such that

$$\mathcal{S} = \arg\min_{\mathcal{S}^*} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i^*} \| \mathbf{x} - \mu_i \|^2,$$

where μ_i is the mean of each cluster.

3 DATA ALIGNMENT AND UNCERTAINTY PROPAGATION

In this Section we will present the curve parameterization based on data alignment to obtain an accurate physics-informed interpolation. We will exemplify the procedure to study the mechanical response of parametric materials loaded in tension.

In this Section we consider a parametric study over dog bone tensile test samples, as sketched in **Figure 1**. We are interested in the influence of the 3 parameters (n, K, ε_0) characterizing the Krupkowski hardening law (also known as *Swift hardening law*), widely used in FEM software

$$\sigma = K(\varepsilon + \varepsilon_0)^n,$$

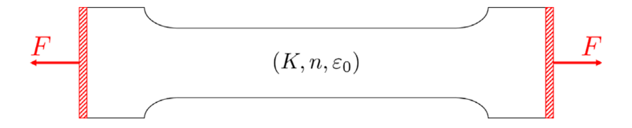

FIGURE 1 | Parametric dog bone specimen loaded in tension.

linking the True Strength and the True Strain. ε denotes the effective plastic strain, ε_0 the offset strain, n the strain hardening exponent and K the material constant.

The image in **Figure 2** top shows two patterns of the Force-Displacement curve, obtained for two different choices of the Krupkowski parameters (blue and orange lines). A classical interpolation of these two patterns would result in the non-physical black dashed pattern.

In what follows, we propose a procedure to overcome such spurious effects, based on a curve alignment prior to interpolate. The method is illustrated over the Force-Displacement curves. However, for the sake of generality, we refer to such curves as generic functions $g(x)$, presenting two characteristic behaviors in the so-called primary and secondary zones. In the specific case of Force-Displacement, the primary zone is the elastic response of the material, up to the yield point x_E. The secondary zone is the post yield behaviour up to the failure point x_F, as illustrated in **Figure 3**. We will also refer to x_E as the "transition point" and to x_F as the "end point", related to the specimen fracture.

We assume that the behaviors in the primary and secondary zone, $g^1(x)$ and $g^2(x)$ respectively, and the transition and end points, x_E and x_F respectively, depend on a series of parameters grouped in vector $\mathbf{p}$, i.e. $g^1(x;\mathbf{p}) \equiv g(x \in [0, x_E]; \mathbf{p})$, $g^2(x;\mathbf{p}) \equiv g(x \in [x_E, x_F]; \mathbf{p})$, $x_E(\mathbf{p})$ and $x_F(\mathbf{p})$. Indeed, when considering different choices of the model parameter $\mathbf{p}_i = (K_i, n_i, \varepsilon_{0,i}), i = 1, \ldots, n_s$, one obtains a set of curves, as the

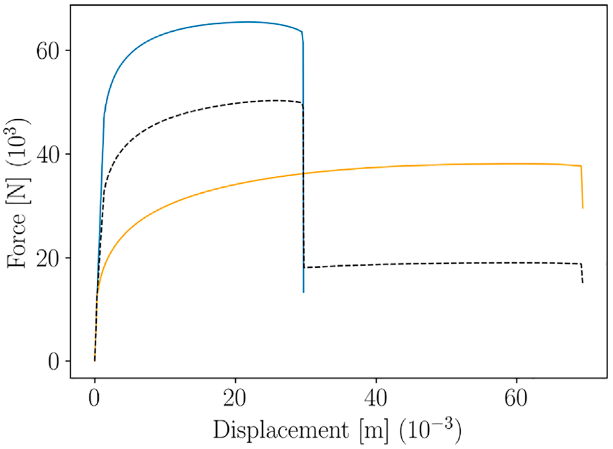

FIGURE 2 | Main issue encountered when using standard interpolations on non-aligned curves (the black dashed line represents the interpolation between the two colored lines).

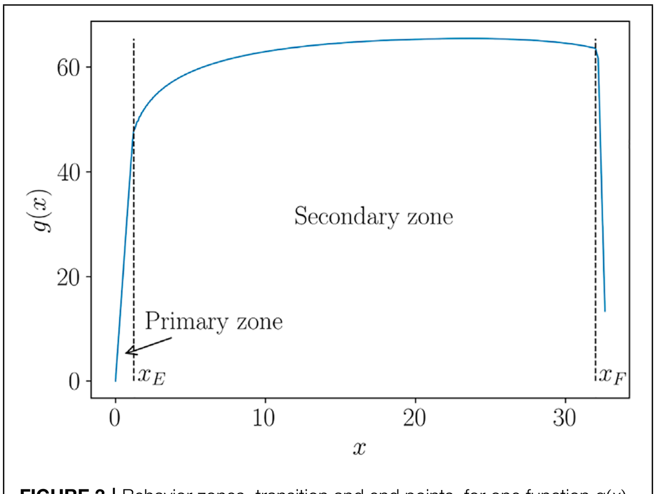

FIGURE 3 | Behavior zones, transition and end points, for one function $g(x)$.

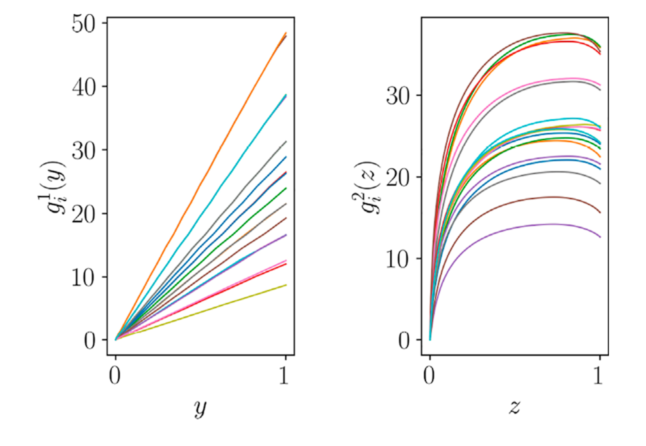

FIGURE 5 | Functions $g_i^1(y) \equiv g^1(y;\mathbf{p}_i)$ (left) and $g_i^2(z) \equiv g^2(z;\mathbf{p}_i)$ (right), for $i = 1,\ldots,n_s$.

ones shown in **Figure 4**, for instance. Such curves correspond to a sparse DoE (Latin Hypercube) of 20 points in the 3-dimensional parametric space $\Omega = I_K \times I_n \times I_{\varepsilon_0}$, considering the parameters bounds specified in **Table 1**. Numerical simulations have been carried out with VPS simulation software from ESI Group. The variable x corresponds to the displacement in mm, while the function $g(x)$ to the force in kN.

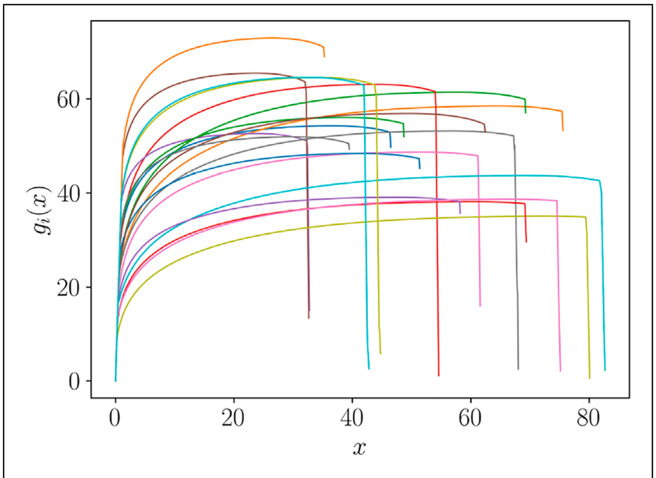

FIGURE 4 | Curves $g(x;\mathrm{p}_i)$ related to different choices of the model features $\mathbf{p}_i = (K_i,n_i,\varepsilon_{0,i}), i = 1,\ldots,n_s$.

TABLE 1 | Parametric ranges.

***K* (MPa)**	n	ε_0
(400, 700)	(0.1, 0.3)	$(0.5, 3) \cdot 10^{-3}$

Once the transition and end points of each curve have been determined, the curves can be rediscretized over the same number of points (through a standard piecewise linear interpolation, for instance). To align them, we define a dimensionless coordinate in each zone, y in the primary zone, $x \in [0, x_E]$, and z in the secondary zone, $x \in [x_E, x_F]$, both defined through the change of variable

$$y = \frac{x}{x_E}, \; y \in [0,1] \text{ and } x \in [0, x_E],$$

and

$$z = \frac{x - x_E}{x_F - x_E}, \; z \in [0,1] \text{ and } x \in [x_E, x_F],$$

expressions that hold for each curve $g(x;\mathbf{p}_i)$, $i = 1,\ldots,n_s$, with

$$y = \frac{x}{x_E^i}, \; y \in [0,1] \text{ and } x \in [0, x_E^i],$$

and

$$z = \frac{x - x_E^i}{x_F^i - x_E^i}, \; z \in [0,1] \text{ and } x \in [x_E^i, x_F^i].$$

Figure 5 depicts functions $g_i^1(y) \equiv g^1(y;\mathbf{p}_i)$ and $g_i^2(z) \equiv g^2(z;\mathbf{p}_i)$.

Actually, this procedure amounts at performing an alignment based on a dilatation of the curves in the first and secondary zone, as shown in **Figure 6**. In such case, we can express the aligned curves as functions of $\tilde{x} \in [0,2]$.

Once the curves have been aligned, the nonlinear regressor presented in **Subsection 2.3.3** can be invoked to build the parametric metamodel of the curve. This can be done separately in each zone or over the whole newly defined coordinate $\tilde{x}$. However, before proceeding with the regression, we address an ulterior parametrization via the Proper Orthogonal Decomposition to achieve a further Model Reduction as discussed in Paragraph 2.3.3.

3.1 POD Modes Extraction

In order to extract the most significant modes able to describe these functions, the POD can be applied in each group of curves in **Figure 5**. This amounts to build the snapshot matrix within

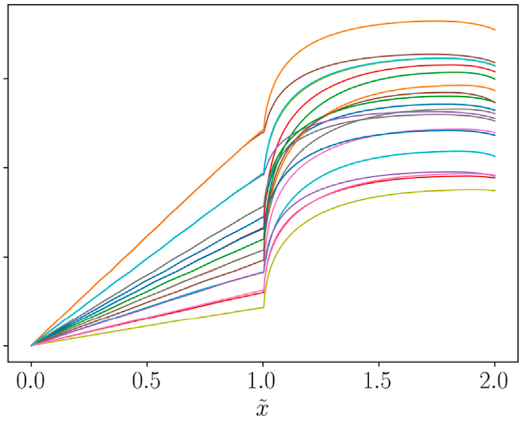

FIGURE 6 | Functions $\tilde{g}_i(\tilde{x})$, for $i = 1, \ldots, n_s$, obtained after dilatation.

each group and perform a truncated SVD. In the case that serves here to illustrate the procedure, a single mode suffices for describing the almost linear functions in the primary zone, that will be noted by $\xi_1(y)$, whereas in the secondary zone two functions are needed, $\phi_1(z)$ and $\phi_2(z)$.

Thus, any function $g_i^1(y)$ can be expressed $\forall i$ as

$$g_i^1(y) = \alpha_1^i \xi_1(y),$$

whereas functions $g_i^2(z)$, $\forall i$, read

$$g_i^2(z) = \beta_1^i \phi_1(z) + \beta_2^i \phi_2(z).$$

The α and β coefficients can be easily computed by simple projection, i.e.

$$\int_0^1 g_i^1(y)\xi_1(y)\ \mathrm{d}y = \alpha_1^i,$$

where the normality of $\xi_1(y)$ was used. In the same way, and taking into account the orthonormality of functions $\phi_1(z)$ and $\phi_2(z)$,

$$\int_0^1 g_i^2(z)\phi_1(z)\ \mathrm{d}z\ = \beta_1^i,$$

and

$$\int_0^1 g_i^2(z)\phi_2(z)\ \mathrm{d}z\ = \beta_2^i.$$

Thus, for each curve $g_i(x)$ we succeeded to extract its five main descriptors: x_E^i, x_F^i, α_1^i, β_1^i and β_2^i, all of them related to the features grouped in vector $\mathbf{p}_i$.

Now, each of these descriptors can be expressed parametrically, $x_E(\mathbf{p}), x_F(\mathbf{p}), \alpha_1(\mathbf{p}), \beta_1(\mathbf{p})$ and $\beta_2(\mathbf{p})$, by using the regression techniques described in **Subsection 2.3.1** for scalar quantities.

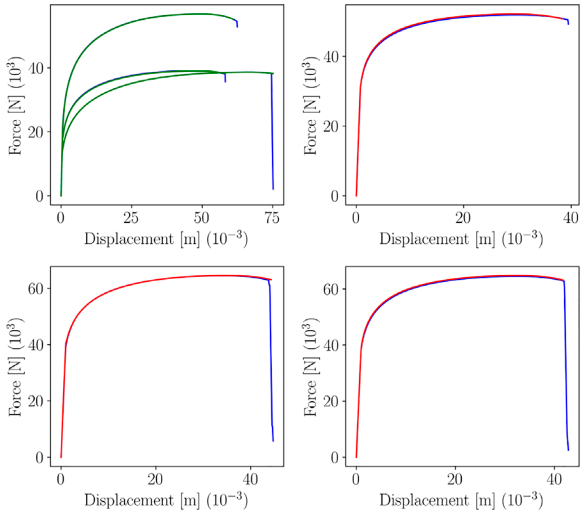

FIGURE 7 | sPGD predictions (green line for training, red for testing) versus true curve (blue line).

3.2 Curves Reconstruction

When considering a choice of the parameters $\mathbf{p}$, the curves descriptors are extracted from the regressions $x_E(\mathbf{p}), x_F(\mathbf{p}), \alpha_1(\mathbf{p}), \beta_1(\mathbf{p})$ and $\beta_2(\mathbf{p})$, the dimensionless coordinates defining both zones calculated from

$$y = \frac{x}{x_E(\mathbf{p})} \rightarrow x = y\ x_E(\mathbf{p}),$$

and

$$z = \frac{x - x_E(\mathbf{p})}{x_F(\mathbf{p}) - x_E(\mathbf{p})} \rightarrow x = x_E(\mathbf{p}) + z\ \ (x_F(\mathbf{p}) - x_E(\mathbf{p})),$$

and, finally, the curve in each zone reconstructed according to

$$g^1(y;\mathbf{p}) = \alpha_1(\mathbf{p})\xi_1(y),$$

and

$$g^2(z;\mathbf{p}) = \beta_1(\mathbf{p})\phi_1(z) + \beta_2(\mathbf{p})\phi_2(z),$$

from which the curve $g(x;\mathbf{p})$ can be straightforward obtained via

$$g(x;\mathbf{p}) = \begin{cases} \alpha_1(\mathbf{p})\xi_1\left(\dfrac{x}{x_E(\mathbf{p})}\right), & x \in [0, x_E(\mathbf{p})] \\ \beta_1(\mathbf{p})\phi_1\left(\dfrac{x - x_E(\mathbf{p})}{x_F(\mathbf{p}) - x_E(\mathbf{p})}\right) + \beta_2(\mathbf{p})\phi_2\left(\dfrac{x - x_E(\mathbf{p})}{x_F(\mathbf{p}) - x_E(\mathbf{p})}\right), & x \in [x_E(\mathbf{p}), x_F(\mathbf{p})]. \end{cases}$$

To build the parametric metamodel, 17 curves have been used to train the sPGD regressor, while the remaining 3 for testing. **Figure 7** shows the resulting predictions over 3 training points and test points.

3.3 Real-Time Calibration

Now, given an experimental curve $g(x)$, its parameters are extracted according to.

- x_E from the point at which the change of behavior occurs (for instance, computing the function derivatives by means of finite differences);
- x_F is the terminal point;
- α_1 follows from $y = \frac{x}{x_E}$ and $\int_0^1 g(y)\xi_1(y)\ \mathrm{d}y = \alpha_1$;

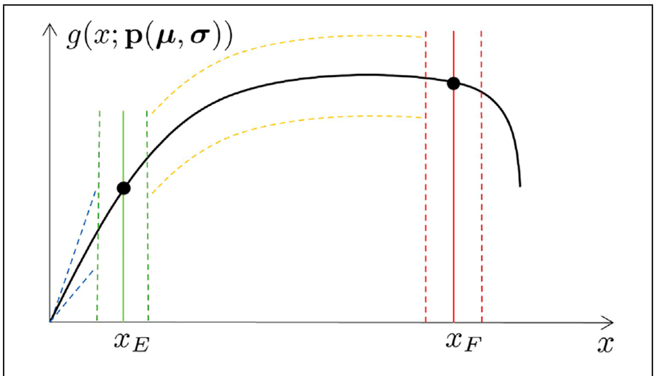

FIGURE 8 | Sketch of curve envelopes.

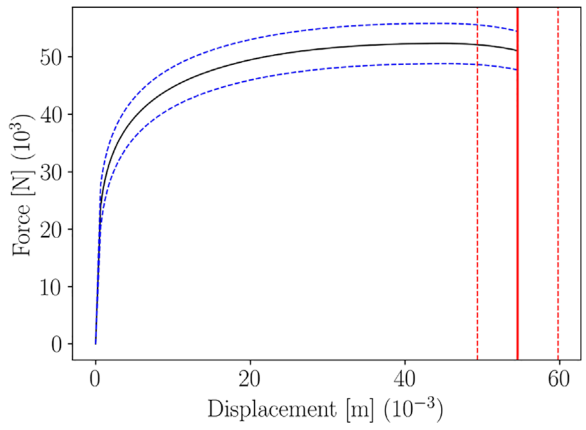

FIGURE 9 | Confidence Interval of level 0.95 for the parametric Force-Displacement curve and for the rupture point, for a given choice of μ and σ.

TABLE 2 | Parametric ranges.

R (mm)	*S* (mm)	*h* (mm)
(3, 8)	(0, 25)	(0.8, 1.6)

- β_1 follows from $z = \frac{x - x_E}{x_F - x_E}$ and $\int_0^1 g(z)\phi_1(z)\ \mathrm{d}z = \beta_1$;
- β_2 follows from $z = \frac{x - x_E}{x_F - x_E}$ and $\int_0^1 g(z)\phi_2(z)\ \mathrm{d}z = \beta_2$.

Then, from the regression models $x_E(\mathbf{p}), x_F(\mathbf{p}), x_1(\mathbf{p}), \beta_1(\mathbf{p})$ and $\beta_2(\mathbf{p})$, the inverse problem is solved for extracting the associated parameters, $\mathbf{p}$.

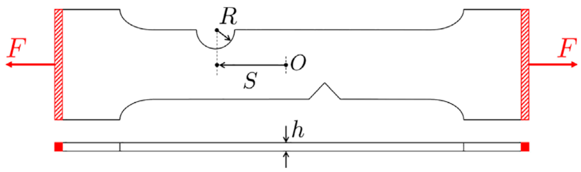

FIGURE 10 | Parametric notched dog bone specimen loaded in tension (top and side views).

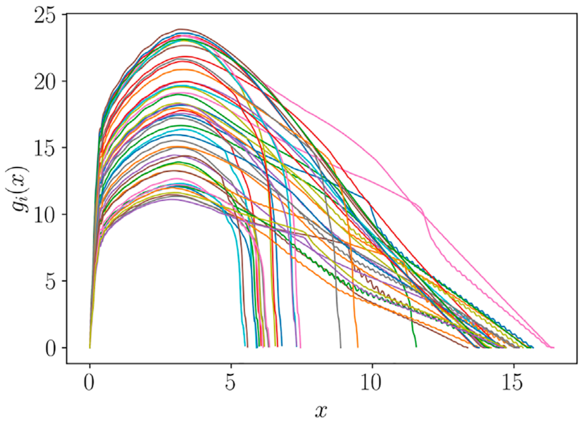

FIGURE 11 | Curves $g_i(x) = g(x;\mathbf{p}_i)$ related to different choices of the model features $\mathbf{p}_i = (R_i, S_i, h_i), i = 1, \ldots, n_S$.

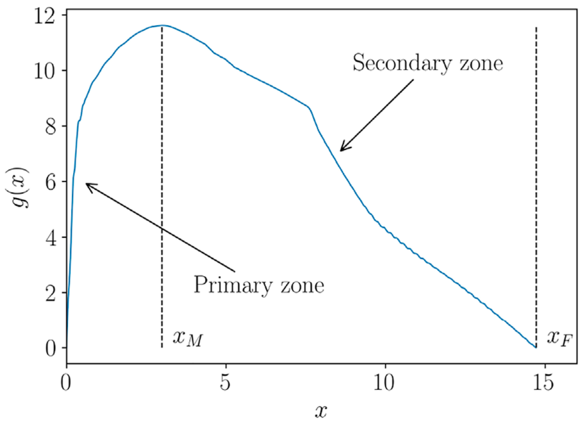

FIGURE 12 | Behavior zones, transition and end points, for one function $g(x)$.

3.4 Statistical Model Derived by Parametric Curves

With the previously built surrogate model, the curve related to any possible value of $\mathbf{p}$ can be computed in real-time, i.e. $g(x;\mathbf{p})$. In this section, this surrogate will be employed for uncertainty quantification.

We assume that each feature p^k in vector $\mathbf{p}$ is assumed characterized by a Gaussian distribution defined its mean value μ_k and its variance σ_k^2, that is $p^k \sim \mathcal{N}(\mu_k, \sigma_k^2)$. Assuming all p^k being independent, we get

$$\mathbf{p} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \boldsymbol{\mu} = (\mu_k)_{k=1}^d, \ \boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma}), \ \boldsymbol{\sigma} = (\sigma_k^2)_{k=1}^d,$$

where diag(•) is the diagonal matrix of diagonal •.

The aim is linking the sensitivity over the input features with the one over the output curve. This means computing some estimators of the average M and the variance Σ of the curve descriptors for different choices of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, and from them, by using the regressions presented in **Subsection 2.3**, build the set

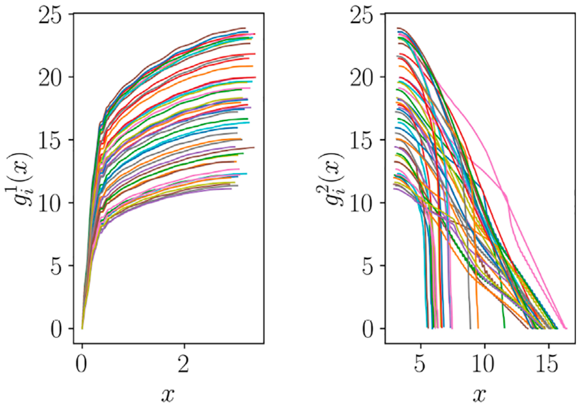

FIGURE 13 | Functions $g_i^1(x) \equiv g^1(x;\mathbf{p}_i)$ (left) and $g_i^1(x) \equiv g^2(x;\mathbf{p}_i)$ (right), with $\mathbf{p}_i = (R_i, S_i, h_i)$, for $i = 1,\ldots,n_S$.

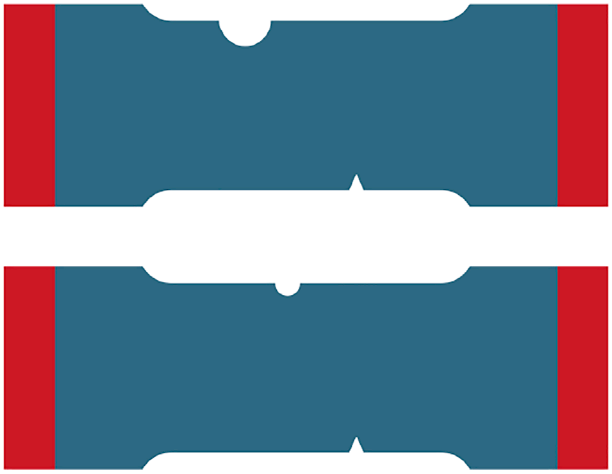

FIGURE 14 | Two different parameters configurations. Top: $R = 7.59$, $S = 18.23$, $h = 0.84$; bottom: $R = 3.75$, $S = 5.58$, $h = 1.51$ (all dimensions are provided in mm). The red zone is the part subject to rigid body constraints.

of statistical surrogates:

$$\begin{cases} \mathcal{S}_{g(x;\mathbf{p})} : (\boldsymbol{\mu},\boldsymbol{\sigma}) \to \left(\overline{M}_{g(x;\mathbf{p})}, \overline{\Sigma}_{g(x;\mathbf{p})}\right), \\ \mathcal{S}_{\mathcal{O}(\mathbf{p})} : (\boldsymbol{\mu},\boldsymbol{\sigma}) \to \left(\overline{M}_{\mathcal{O}(\mathbf{p})}, \overline{\Sigma}_{\mathcal{O}(\mathbf{p})}\right). \end{cases} \tag{5}$$

where $\mathcal{O}(\mathbf{p})$ denotes any QoI involved in the curves parametrization (i.e., an output depending on the input parameters; e.g., x_E, x_F, α_1, β_1 and β_2 in the example presented before) and $\overline{M}$ and $\overline{\Sigma}$ the corresponding estimators for mean and variance, respectively. This allows calculating the envelopes, for a given confidence, of the curves, as sketched in **Figure 8**.

To build the surrogate (5), for instance for the curve descriptor $\mathcal{O}(\mathbf{p})$, a training dataset of N_s points shall be generated:

$$\left\{\left(\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j\right), \left(\overline{M}_{\mathcal{O}(\mathbf{p}_j)}, \overline{\Sigma}_{\mathcal{O}(\mathbf{p}_j)}\right)\right\}_{j=1}^{N_s}.$$

This can be achieved by means of a Monte Carlo sampling, which gives the estimators of mean and variance for the curves $g(x;\mathbf{p}_j(\boldsymbol{\mu}_j,\boldsymbol{\sigma}_j))$, and of any descriptor $\mathcal{O}(\mathbf{p}_j)$, for $j = 1,\ldots,N_s$.

The whole procedure is summarized in **Algorithm 1**.

Algorithm 1 Statistical sensing based on parametric curves

```
Input:
1. $f^X(\mathbf{p})$, $\mathbf{p} = (p^1,\ldots,p^d)$: curves regression model;
2. $N_s$: number of training points for the statistical surrogate model $\mathcal{S}_{\mathcal{O}(\mathbf{p})}$;
3. $N_{\mathrm{MC}}$: number of Monte Carlo sampling points.
Output:
$(M_{\mathcal{O}(\mathbf{p})}, \Sigma_{\mathcal{O}(\mathbf{p})})$: regression model for mean and variance of curve descriptor $\mathcal{O}(\mathbf{p})$.
for $j = 1,\ldots,N_s$ do
    Randomly sample (e.g., LHS) the model features means and variances
        $(\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j), \quad \boldsymbol{\mu}_j = (\mu_{j,k})_{k=1}^d, \ \boldsymbol{\sigma}_j = (\sigma_{j,k}^2)_{k=1}^d.$
    Perform a Monte Carlo sampling of the curves statistical descriptor $\mathcal{O}(\mathbf{p})$:
    1. generate a population of $N_{\mathrm{MC}}$ vectors of features $\mathbf{p}_j = (p_j^k)_{k=1}^d$, by sampling $N_{\mathrm{MC}}$ points from
        $p_j^k \sim \mathcal{N}(\mu_{j,k}, \sigma_{j,k}^2), \ k = 1,\ldots,d;$
    2. generate the population of the corresponding $N_{\mathrm{MC}}$ curves (and of any QoI involved in their parametrization), by using the curves surrogate $f^X(\mathbf{p})$, that is,
        $g(x;\mathbf{p}_{j,l}), \mathcal{O}(\mathbf{p}_{j,l}) = f^X(\mathbf{p}_{j,l}), \ l = 1,\ldots,N_{\mathrm{MC}};$
    3. compute the population mean and variance to obtain the corresponding Monte Carlo estimators for the curve $g(x;\mathbf{p}_j)$ and its descriptor $\mathcal{O}(\mathbf{p}_j)$:
        $(\bar{M}_{g(x;\mathbf{p}_j)}, \bar{\Sigma}_{g(x;\mathbf{p}_j)}), \quad (\bar{M}_{\mathcal{O}(\mathbf{p}_j)}, \bar{\Sigma}_{\mathcal{O}(\mathbf{p}_j)}).$
end for
Using the previously built population $\{(\bar{M}_{\mathcal{O}(\mathbf{p}_j)}, \bar{\Sigma}_{\mathcal{O}(\mathbf{p}_j)})\}_{j=1}^{N_s}$, train a regression model for the statistical sensing of $\mathcal{O}(\mathbf{p})$ involved in the curve parametrization:
    $\mathcal{S}_{\mathcal{O}(\mathbf{p})} : (\boldsymbol{\mu},\boldsymbol{\sigma}) \to \left(\bar{M}_{\mathcal{O}(\mathbf{p})}, \bar{\Sigma}_{\mathcal{O}(\mathbf{p})}\right).$
Same procedure holds for the whole curve $g(x;\mathbf{p})$.
Given a new couple $(\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*)$ of model features means and variances, corresponding to the features $\mathbf{p}^*$, one can obtain a Confidence Interval –CI– at a given confidence level for the output. For instance, at level 0.95, one can build a CI for the curve $g(x;\mathbf{p}^*)$:
    $g(x;\mathbf{p}^*) \in \left[\bar{M}_{g(x;\mathbf{p}^*)} - 2\sqrt{\bar{\Sigma}_{g(x;\mathbf{p}^*)}}, \bar{M}_{g(x;\mathbf{p}^*)} + 2\sqrt{\bar{\Sigma}_{g(x;\mathbf{p}^*)}}\right].$
```

:

Figure 9 shows the parametric curve and its statistical sensing, for a given choice of the input features distribution parameters. Confidence Intervals have been computed using **Algorithm 1**, for the curve and the rupture point.

3.5 Statistical Model Derived From Measures

In this Section we consider that for different choices of the problem features $\mathbf{p}_i$, the measure $g^m(x;\mathbf{p}_i)$ is collected. We assume that measures contain a significant uncertainty, modeled again, without loss of generality, by a Gaussian distribution of null average and variance σ, that is, $\mathcal{N}(0,\sigma^2)$, with the variance assumed independent of the features $\mathbf{p}$.

In these circumstances applying a regression to fit those values $g^m(x;\mathbf{p}_i)$, that is $f^{X,m}(\mathbf{p}_i) = g^m(x;\mathbf{p}_i)$, according to the techniques described in **Subsection 2.3** is not a valuable route. The most valuable solution consists of looking for the baseline regression $f^X(\mathbf{p})$ such that the deviation $\mathcal{D}_i = f^{X,m}(\mathbf{p}_i) - f^X(\mathbf{p})$ follows the distribution $\mathcal{N}(0,\sigma^2)$, where both the regression parameters involved in $f^X(\mathbf{p})$ and the variance (if not known a priori) are calculated. In some cases the sensor calibration allows identifying σ^2.

The just described procedure is very close to standard Bayesian inference.

3.6 Model Enrichment

When two regression models are known, for the sake of simplicity assumed scalar, one related to a physics based model $f^{X,\text{model}}(\mathbf{p})$ and the second one to the measures $f^{X,\text{measure}}(\mathbf{p})$, both associated with the average values in case of uncertainty in the model

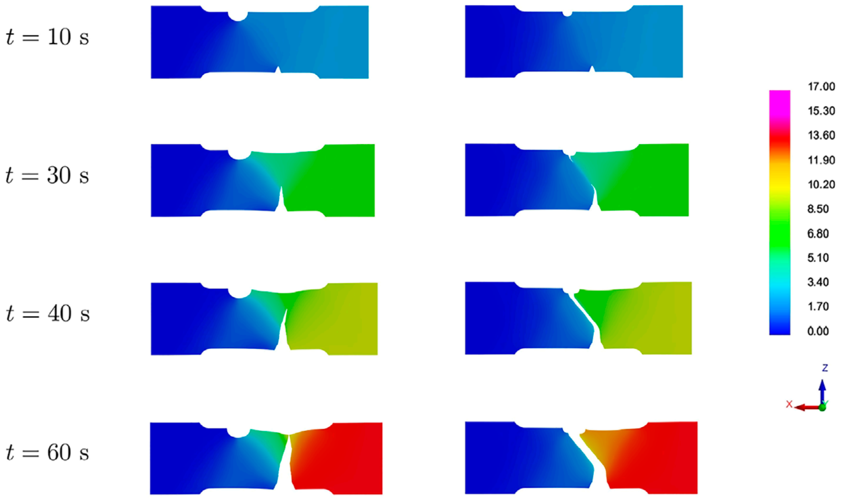

FIGURE 15 | Bifurcation in the parametric space causing completely different crack propagation dynamics.

and the measures, one could define the gap model $\Delta f^X(\mathbf{p})$ from $f^{X,\text{measure}}(\mathbf{p}) - f^{X,\text{model}}(\mathbf{p}) \equiv \Delta f^X(\mathbf{p})$.

Thus, the enriched model reads

$$f^{X,\text{enrich}}(\mathbf{p}) = f^{X,\text{model}}(\mathbf{p}) + \Delta f^X(\mathbf{p}).$$

As in general the nonlinear character of $f^{X,\text{measure}}(\mathbf{p})$ is expected being much higher than the one of the gap, $\Delta f^X(\mathbf{p})$, a more valuable route consists in calculating the discrete gap $\mathcal{D}(\mathbf{p}_i) = f^{X,m}(\mathbf{p}_i) - f^{\text{model}}(\mathbf{p}_i)$ and then calculate the regression $\widetilde{\Delta f}^X(\mathbf{p})$ fitting the discrete deviations, and the associated enriched model $\tilde{f}^{X,\text{enrich}}(\mathbf{p})$

$$\tilde{f}^{X,\text{enrich}}(\mathbf{p}) = f^{X,\text{model}}(\mathbf{p}) + \widetilde{\Delta f}^X(\mathbf{p}).$$

4 DATA ALIGNMENT AND DATA CLUSTERING

Here we focus on the study of crack propagation in notched specimens loaded in tension, whose geometry is sketched in **Figure 10**. The test piece has a V-shaped notch defect which is always at the same location (almost bottom-middle). On the other side of the test piece there is a half-circle groove. The goal is to predict the crack propagation from the defect based off of different locations (S) and radii (R) of the groove, as well as different test piece thicknesses (h). Depending on the location of the groove, the crack will propagate differently from the defect to the groove.

We have considered a sparse DoE (Latin Hypercube) of 50 points in the 3-dimensional parametric space $\Omega = I_R \times I_S \times I_h$, with the parameters bounds specified in **Table 2**. Numerical simulations (carried out in VPS software from ESI Group) employ an Explicit Analysis and the EWK rupture model (Kamoulakos, 2005), using a mesh of 1096218 solid elements.

We focus on the prediction of the Force-Displacement curves plotted in **Figure 11**, which are considered as the generic functions $g(x)$, following the same notation of **Section 3**.

It can be observed that all the curves present a similar pattern in the first zone, monotonically increasing, while the response appears much different in the secondary zone. A first pre-processing step consists in splitting the zones as illustrated in **Figure 12**, where x_M denotes the point where the curve reaches its maximum value, while x_F its endpoint.

Cutting the curves, we obtain the two groups of functions plotted in **Figure 13**, which are of course not aligned. However, they can be expressed as functions of normalized coordinates y and z, respectively, and aligned following the dilatation procedure discussed in **Section 3**.

Once the alignment has been performed, using the usual nonlinear regression techniques of **Subsection 2.3** and same notations of **Section 3**, two regression models, one for each group, can be established:

$$\begin{cases} g^1(x;\mathbf{p}) := g(x \in [0, x_M(\mathbf{p})]) = f_1^X(\mathbf{p}) \\ g^2(x;\mathbf{p}) := g(x \in [x_M(\mathbf{p}), x_F(\mathbf{p})]) = f_2^X(\mathbf{p}). \end{cases} \tag{6}$$

In **Eq. 6**, for the sake of clarity, we have specified x_M and x_F since these points are involved into the parametrization of the functions $g^1(x)$ and $g^2(x)$, respectively, and thus expressed parametrically.

As we have previously pointed out, the second group of functions $g_i^2(x)$, for $i = 1, \ldots, n_s$, presents really different shapes depending on the features $\mathbf{p}_i$. When bifurcations occur in the parametric space, the system responses related to two choices of the model parameters can be completely different. In such cases, a standard nonlinear regression over the full space can lead to inaccurate and nonphysical solutions. To enhance the accuracy of the model $f_2^X(\mathbf{p})$, a more valuable route consists in exploring the parametric space prior to interpolation. This can be done via a clustering of the system responses. Once the clusters

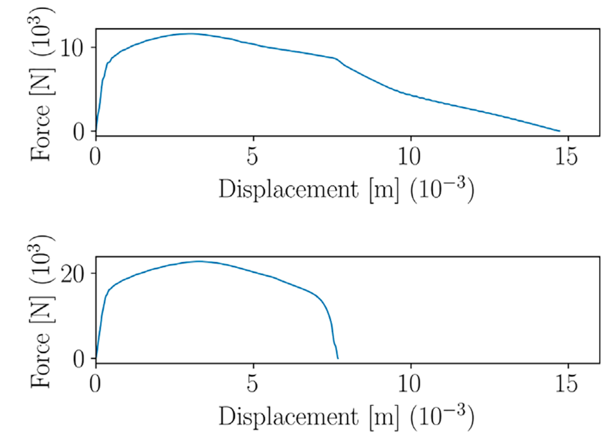

FIGURE 16 | Force-Displacement curves corresponding to the two parameters configurations in **Figure 14**.

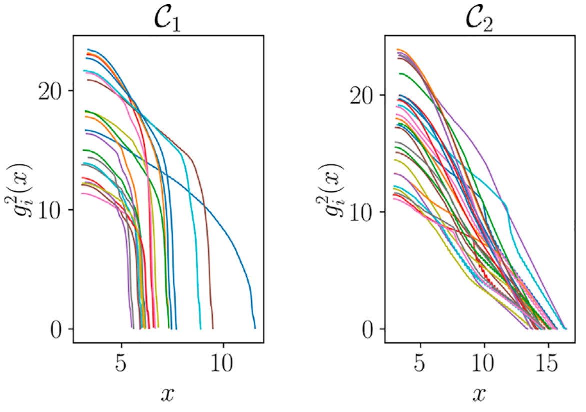

FIGURE 17 | Functions $g_i^2(x)$ of **Figure 13** (right) after clustering, for $i = 1, \ldots, n_s$.

have been established, several regression sub-models can be built, minimizing the risk of mixing spurious effects coming from other clusters.

4.1 Clustering

To exemplify the bifurcation problem in the parametric space, we consider two different configurations of the model parameters, resulting into the specimens shown in **Figure 14**.

Figure 15 shows four snapshots of the displacement field related to the specimens in **Figure 14**, under axial tensile loading. The crack propagation follows two completely different patterns, drastically influencing the Force-Displacement curve, as shown in **Figure 16**.

The clustering step can be performed automatically by using a hierarchical clustering based on the curves shape or on the location of damaged elements into the finite element mesh. Once the clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ have been established, two regression

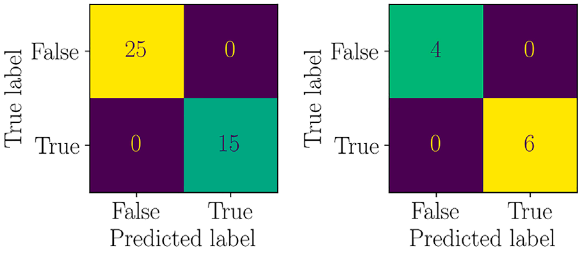

FIGURE 18 | Confusion Matrices for the SVM classifier (left: training data, right: test data).

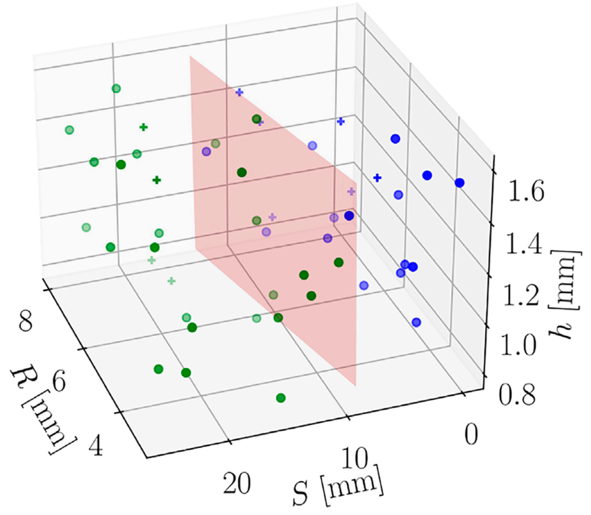

FIGURE 19 | Parametric space and classified points (marker + is used for test points). The red plane is the separation surface.

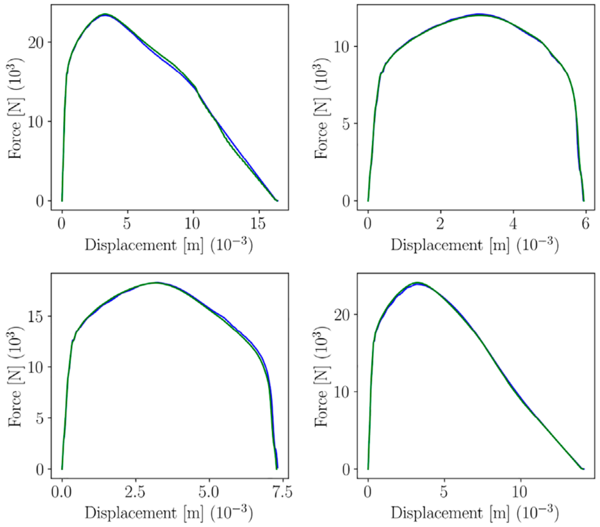

FIGURE 20 | sPGD predictions (green line) versus true curve (blue line) for training data.

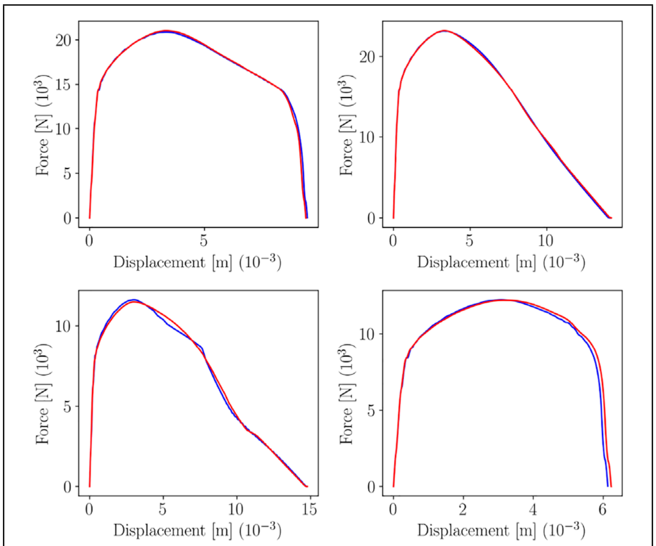

FIGURE 21 | sPGD predictions (red line) versus true curve (blue line) for test data.

submodels can be trained, one for each cluster, and **Eq. 6** becomes

$$\begin{cases} g^1(x;\mathbf{p}) = f_1^X(\mathbf{p}) \\ g^2(x;\mathbf{p}) = \begin{cases} f_{2,1}^X(\mathbf{p}) & \text{for } \mathcal{C}_1 \\ f_{2,2}^X(\mathbf{p}) & \text{for } \mathcal{C}_2. \end{cases} \end{cases} \tag{7}$$

Figure 17 shows the functions in the secondary zone after the clustering.

In particular, one can remark that fracture occurs early on for tests belonging to cluster $\mathcal{C}_1$ and the final part of the curve is characterized by a steep slope. On the contrary, tests belonging to cluster $\mathcal{C}_2$ have an endpoint displacement around 15 mm and present a shallow slope. The clustering allows to avoid averaging such different dynamics, clearly enhancing the quality of the regressor.

4.2 Curves Reconstruction and Classification

For a newly defined choice of model features $\mathbf{p}^*$, the curve $g(x;\mathbf{p}^*)$ is obtained via

$$g(x;\mathbf{p}^*) = \begin{cases} g^1(x;\mathbf{p}^*), & 0 \le x \le x_M(\mathbf{p}^*) \\ g^2(x;\mathbf{p}^*), & x_M(\mathbf{p}^*) < x \le x_F(\mathbf{p}^*), \end{cases}$$

where g^1 and g^2 are obtained through **Eq. 7**.

The training of the regression models has been performed using 40 points of the DoE, remaining 10 have been used for testing. Moreover, a Support Vector Machine classifier (a Random Forest classifier could also be used, for instance) has been trained to select the best regression submodel to predict $g^2(x;\mathbf{p}^*)$. Such classifier has shown perfect accuracy, as shown by the Confusion Matrices in **Figure 18**. Moreover, **Figure 19** shows the separating surface and classified points in the 3-dimensional parametric space.

Figures 20, **21** represent the plots of predictions for train and test, respectively, for 4 data points.

5 CONCLUSION

In this paper we have focused on several nontrivial issues encountered when a whole curve shall be predicted from a given number of features. A major argument is the data alignment to achieve physics-consistent interpolations among curves and the data clustering to detect bifurcations in the parametric space. The proposed methodologies rely on adopting specific parametrizations of the curve and a physics-based pre-processing prior to the application of any regression technique. We have also suggested a reduced order parametrization of the curve via POD coefficients, requiring the prediction of a few scalar quantities (i.e., the POD coefficients) instead of the whole curve. Here, without loss of generality, we have preferred sPGD-based nonlinear regressions, these being efficient in high-dimensional parametric spaces under the scarce data limit constraint. Indeed, since our data come from numerical simulations of complex engineering problems, due to the high computational complexity of the offline simulations, not much data are usually available. Moreover, one important achievement of the work is the definition of a statistical sensing for uncertainty propagation based on the parametric model.

We have focused on two applications in computational mechanics: 1) plastic materials with parametric hardening law, 2) crack propagation in parametric notched specimens. However, these methodologies can be applied to any time series or generic curve stem from any context. For instance, in our current research, we are successfully applying these techniques to solve many other problems (to cite some, the study of a two-phase flow dynamics in a heated channel, the composite forming processes involving a reactive resin injection molding). Moreover, we are focusing on other physics-based curves interpolation strategies based on Optimal Transport–OT–(Torregrosa et al., 2022) and other mappings.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

All the authors participated in the definition of techniques and algorithms. All authors read and approved the final manuscript.

ACKNOWLEDGMENTS

The authors acknowledge the support of ESI Group through its research chair at ENSAM ParisTech. This research is part of the programme DesCartes and is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

REFERENCES

Amsallem, D., and Farhat, C. (2008). Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity. *AIAA J.* 46, 1803–1813. doi:10.2514/1.35374

Audouze, C., De Vuyst, F., and Nair, P. B. (2013). Nonintrusive Reduced-Order Modeling of Parametrized Time-dependent Partial Differential Equations. *Numer. Methods Partial Differ. Eq.* 29, 1587–1628. doi:10.1002/num.21768

Benner, P., Schilders, W., Grivet-Talocia, S., Quarteroni, A., Rozza, G., and Silveira, L. M. (2020a). *Model Order Reduction: Applications*. Berlin: De Gruyter.

Benner, P., Schilders, W., Grivet-Talocia, S., Quarteroni, A., Rozza, G., and Silveira, L. M. (2020b). *Model Order Reduction: Snapshot-Based Methods and Algorithms*. Berlin: De Gruyter.

Benner, P., Schilders, W., Grivet-Talocia, S., Quarteroni, A., Rozza, G., and Silveira, L. M. (2020c). *Model Order Reduction: System- and Data-Driven Methods and Algorithms*. Berlin: De Gruyter.

Benner, P., Gugercin, S., and Willcox, K. (2015). A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Rev.* 57, 483–531. doi:10.1137/130932715

Borzacchiello, D., Aguado, J. V., and Chinesta, F. (2017). Non-intrusive Sparse Subspace Learning for Parametrized Problems. *Arch. Comput. Methods Eng.* 26, 303–326. doi:10.1007/s11831-017-9241-4

Chinesta, F., Ladeveze, P., and Cueto, E. (2011). A Short Review on Model Order Reduction Based on Proper Generalized Decomposition. *Arch. Comput. Methods Eng.* 18, 395–404. doi:10.1007/s11831-011-9064-7

Cristianini, N., and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge: Cambridge University Press.

de Gooijer, B. M., Havinga, J., Geijselaers, H. J. M., and Van den Boogaard, A. H. (2021). Evaluation of Pod Based Surrogate Models of Fields Resulting from Nonlinear Fem Simulations. *Adv. Model. Simul. Eng. Sci.* 8. doi:10.1186/s40323-021-00210-8

Fareed, H., and Singler, J. R. (2019). A Note on Incremental Pod Algorithms for Continuous Time Data. *Appl. Numer. Math.* 144, 223–233. doi:10.1016/j.apnum.2019.04.020

Franchini, A., Sebastian, W., and D'Ayala, D. (2022). Surrogate-based Fragility Analysis and Probabilistic Optimisation of Cable-Stayed Bridges Subject to Seismic Loads. *Eng. Struct.* 256, 113949. doi:10.1016/j.engstruct.2022.113949

Friderikos, O., Baranger, E., Olive, M., and Néron, D. (2022). *On the Stability of Pod Basis Interpolation on Grassmann Manifolds for Parametric Model Order Reduction. Comput Mech*. Cham: Springer. doi:10.1007/s00466-022-02163-0

Friderikos, O., Olive, M., Baranger, E., Sagris, D., and David, C. N. (2020). A Space-Time Pod Basis Interpolation on Grassmann Manifolds for Parametric Simulations of Rigid-Viscoplastic Fem. *MATEC Web Conf.* 318, 01043. doi:10.1051/matecconf/202031801043

Hesthaven, J. S., Rozza, G., and Stamm, B. (2016). *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Cham: Springer. doi:10.1007/978-3-319-22470-1

Hesthaven, J. S., and Ubbiali, S. (2018). Non-intrusive Reduced Order Modeling of Nonlinear Problems Using Neural Networks. *J. Comput. Phys.* 363, 55–78. doi:10.1016/j.jcp.2018.02.037

Hilberg, D., Lazik, W., and Fiedler, H. E. (1994). The Application of Classical Pod and Snapshot Pod in a Turbulent Shear Layer with Periodic Structures. *Appl. Sci. Res.* 53, 283–290. doi:10.1007/bf00849105

Ibáñez, R., Abisset-Chavanne, E., Ammar, A., González, D., Cueto, E., Huerta, A., et al. (2018). A Multidimensional Data-Driven Sparse Identification Technique: The Sparse Proper Generalized Decomposition. *Complexity* 2018, 1–11. doi:10.1155/2018/5608286

Kamoulakos, A. (2005). "The ESI-Wilkins-Kamoulakos (EWK) Rupture Model," in *Continuum Scale Simulation of Engineering Materials: Fundamentals - Microstructures - Process Applications* (Hoboken: John Wiley & Sons), 795–804. doi:10.1002/3527603786.ch43

Khatouri, H., Benamara, T., Breitkopf, P., and Demange, J. (2022). Metamodeling Techniques for Cpu-Intensive Simulation-Based Design Optimization: a Survey. *Adv. Model. Simul. Eng. Sci.* 9. doi:10.1186/s40323-022-00214-y

MacQueen, J. B. (1967). "Some Methods for Classification and Analysis of Multivariate Observations," in *Proc. Of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Editors L. M. L. Cam., and J. Neyman (California: University of California Press), 281–297.

Mainini, L., and Willcox, K. (2015). Surrogate Modeling Approach to Support Real-Time Structural Assessment and Decision Making. *AIAA J.* 53, 1612–1626. doi:10.2514/1.J053464

Mosquera, R., El Hamidi, A., Hamdouni, A., and Falaize, A. (2021). Generalization of the Neville-Aitken Interpolation Algorithm on Grassmann Manifolds: Applications to Reduced Order Model. *Int. J. Numer. Meth Fluids* 93, 2421–2442. doi:10.1002/fld.4981

Mosquera, R., Hamdouni, A., Hamdouni, A., El Hamidi, A., and Allery, C. (2018). Pod Basis Interpolation via Inverse Distance Weighting on Grassmann Manifolds. *Discrete Continuous Dyn. Syst. - S* 12, 1743–1759. doi:10.3934/dcdss.2019115

Prud'homme, C., Rovas, D. V., Veroy, K., Machiels, L., Maday, Y., Patera, A. T., et al. (2002). Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods. *J. Fluids Eng.* 124, 70–80. doi:10.1115/1.1448332

Raghavan, B., Hamdaoui, M., Xiao, M., Breitkopf, P., and Villon, P. (2013). A Bi-level Meta-Modeling Approach for Structural Optimization Using Modified Pod Bases and Diffuse Approximation. *Comput. Struct.* 127, 19–28. doi:10.1016/j.compstruc.2012.06.008

Rajaram, D., Perron, C., Puranik, T. G., and Mavris, D. N. (2020). Randomized Algorithms for Non-intrusive Parametric Reduced Order Modeling. *AIAA J.* 58, 5389–5407. doi:10.2514/1.J059616

Sancarlos, A., Champaney, V., Duval, J., Cueto, E., and Chinesta, F. (2021). *Pgd-based Advanced Nonlinear Multiparametric Regressions for Constructing Metamodels at the Scarce-Data Limit. CoRR abs/2103.05358*, ArXiv.

Simpson, T. W., Poplinski, J. D., Koch, P. N., and Allen, J. K. (2001). Metamodels for Computer-Based Engineering Design: Survey and Recommendations. *Eng. Comput.* 17, 129–150. doi:10.1007/PL00007198

Torregrosa, S., Champaney, V., Ammar, A., Herbert, V., and Chinesta, F. (2022). Surrogate Parametric Metamodel Based on Optimal Transport. *Math. Comput. Simul.* 194, 36–63. doi:10.1016/j.matcom.2021.11.010

Wang, G. G., and Shan, S. (2007). Review of Metamodeling Techniques in Support of Engineering Design Optimization. *J. Mech. Des.* 129, 370–380. doi:10.1115/1.2429697

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.