Received: 10 November 2023 | Revised: 27 February 2024 | Accepted: 17 March 2024

DOI: 10.1002/nme.7487

RESEARCH ARTICLE

WILEY

Plug-and-play adaptive surrogate modeling of parametric nonlinear dynamics in frequency domain

Phillip Huwiler[1] | **Davide Pradovera**[2,3] | **Jürg Schiffmann**[1]

[1]Laboratory for Applied Mechanical Design, Department of Mechanical Engineering, EPFL, Neuchâtel, Switzerland

[2]Department of Mathematics, University of Vienna, Vienna, Austria

[3]Department of Mathematics, KTH, Stockholm, Sweden

Correspondence
Phillip Huwiler, Laboratory for Applied Mechanical Design, Department of Mechanical Engineering, EPFL, 2000 Neuchâtel, Switzerland.
Email: phillip.huwiler@epfl.ch

Abstract

We present an algorithm for constructing efficient surrogate frequency-domain models of (nonlinear) parametric dynamical systems in a *non-intrusive* way. To capture the dependence of the underlying system on frequency and parameters, our proposed approach combines rational approximation and smooth interpolation. In the approximation effort, locally adaptive sparse grids are applied to effectively explore the parameter domain even if the number of parameters is modest or high. Adaptivity is also employed to build rational approximations that efficiently capture the frequency dependence of the problem. These two features enable our method to build surrogate models that achieve a user-prescribed approximation accuracy, without wasting resources in "oversampling" the frequency and parameter domains. Thanks to its non-intrusiveness, our proposed method, as opposed to projection-based techniques for model order reduction, can be applied regardless of the complexity of the underlying physical model. Notably, our algorithm for adaptive sampling can be used even when prior knowledge of the problem structure is not available. To showcase the effectiveness of our approach, we apply it in the study of an aerodynamic bearing. Our method allows us to build surrogate models that adequately identify the bearing's behavior with respect to both design and operational parameters, while still achieving significant speedups.

KEYWORDS

frequency domain, gas bearing, high-dimensional approximation, model order reduction, nonlinear dynamics

1 | INTRODUCTION

Frequency-domain models of (nonlinear) dynamical systems are often used by engineers in domains such as control systems, fluid mechanics, acoustics, optics, electronics and communications. A good understanding of nonlinear effects can be crucial for stability analysis, performance evaluation, and mitigation of undesirable phenomena. Specifically, with the objective of resolving complex behaviors in physical or data-driven models, high-fidelity *parametric* models are developed. The parameters are embedded within the model as "tuning knobs" of the interesting inputs to the system.

Simultaneously, engineers are often not only unwilling to accept the exploding computational cost that accompanies an up-scaling in modeling complexity, but some applications also necessitate significant reductions in computing time and

memory. This is particularly relevant in the so-called multi-query setting, where calculations have to be repeated for many different values of frequency and parameters, for example, in the context of design optimization or of real-time monitoring and control. Model order reduction (MOR) addresses this by replacing the high-fidelity model with a cheap-to-evaluate surrogate one, which (hopefully) has a small approximation error. Evaluation efficiency is not the only concern when considering modeling alternatives: surrogate costs can come in the form of variable (so-called "*online*") costs on a per-evaluation basis and overhead (so-called "*offline*") costs for building the surrogate model. Examples of the latter include (re)-training and/or development effort associated with modifications in the reference model, which could be the case for techniques exploiting specific knowledge about the inner workings of the reference model.

We wish for the reader not to lose sight of the general applicability of our algorithm to frequency-dependent, nonlinear, and parametric dynamical systems, whose abstract mathematical formulation is outlined in Section 2. Yet, we are also practitioners and see value in showcasing our approach on a tractable engineering application. For this, we narrow-in on herringbone grooved journal bearings (HGJBs), a type of non-contact bearings where the pressure distribution in a thin film of gas separates a rotating shaft from a static bearing part. A reference HGJB model is summarized in Section 4. Several numerical experiments with the HGJB are then performed in Section 5. We remind the reader that the details about the reference model are merely given to establish a context about the functioning and performance of the algorithm. They are not needed for its application. In this sense, our proposed algorithm can be thought of as *plug-and-play* in an approximation setting for *black-box* functions.

1.1 | Model reduction background and contributions

By now, a vast body of works in MOR is available, and many different approaches have been developed for building surrogate models. In the specific case of nonlinear systems, some additional difficulties arise, especially when employing the most popular class of MOR techniques, namely, projection-based approaches such as proper orthogonal decomposition (POD) or reduced basis (RB) methods (see, e.g., Reference 1 for an overview): due to the presence of nonlinear terms, it is difficult to produce a surrogate model that is both efficient and accurate. Strategies to address this issue have been studied under the umbrella term "hyperreduction", with the most known approach being the discrete empirical interpolation method (DEIM).[2] There are also two notable weaknesses of DEIM: (i) it is very problem-dependent and often requires considerable "user supervision" (limiting the extent to which it can be applied in an automated way), and (ii) it is *intrusive* (i.e., it requires knowledge of the nonlinearities and access to the high-dimensional system "state", such as, e.g., the pressure or displacement field that solves the nonlinear equation, cf. Section 2). This second item is a serious issue in many practical settings, specifically for severe nonlinearities and whenever one cannot easily access the high-fidelity state, for example, due to closed-source code. Even "less" intrusive methods like "non-intrusive RB"[3] or physics-informed neural networks[4,5] are affected by this problem. Note that further difficulties arise for intrusive *parametric* MOR methods since dedicated techniques contingent on the nature of parameter dependencies are needed to adapt the bases for given parameter values, see, for example, References 6–9.

In this work, we address this issue by developing a *non-intrusive* MOR approach for building surrogate models of quantities of interest (QoIs) arising in frequency-domain analyses of nonlinear dynamics. Our method builds the surrogate model from samples of the QoIs only, without requiring any access to –nor knowledge of– the underlying nonlinear model. This allows us to treat the high-fidelity model as a "black box".

We build a surrogate model by interpolation. More precisely, a rational interpolation framework is employed with respect to frequency, whereas smooth interpolation (e.g., by piecewise polynomials or radial basis functions) is applied to approximate the dependence on parameters. The frequency dependence is modeled via the (less regular) class of rational functions as a way to address any potential lack of smoothness of the QoIs with respect to frequency. In this context, rational functions have been often used as a general-purpose approximation class,[10–14] often yielding (quasi-)optimal approximation quality, regardless of whether such "non-smooth effects" are actually present or not. This being said, we refrain from using rational interpolation to model the joint dependence of the QoIs on frequency and parameters (as done in, e.g., References 15,16), because of the intrinsic instabilities and computational difficulties associated to multivariate rational approximation, which are ultimately due to the "*curse of dimension*".

Lastly, we consider one additional (and, in our view, crucial) issue in MOR: the selection of the sample points. Given the task of building a surrogate model, one inevitably faces the questions of how many and where (in frequency-and-parameter space) to take the expensive high-fidelity samples that the surrogate will be eventually based on. Answering these questions well is crucial to guarantee accuracy and efficiency of the surrogate model, but also to

limit the cost of the so-called "offline phase", that is, the assembly of the surrogate. Sampling is a particularly tricky issue in non-intrusive methods: the usual residual-based greedy strategies[1] cannot be employed, due to the absence of an error estimator. In practice, this hinders an efficient identification of points where the approximation error is large. In our method, we employ a fully non-intrusive approach for adaptive sampling, based on a predictor-corrector strategy: a modestly sized test set is used to assess the quality of the current surrogate, and also to determine where to add new sample points in case of insufficient accuracy.

In terms of novelty, MOR approaches similar to ours were already considered in References 17,18 for generic parametric problems. In those works, a "fully smooth" approximation framework was used, relying on, for example, piecewise polynomials or radial bases for building an interpolatory surrogate model with respect to both frequency and parameters. In this paper, we modify the method to account for the frequency component in the high-fidelity model, effectively specializing the approximation strategy from References 17,18 to frequency-domain modeling. From a practical viewpoint, this requires combining the rational-in-frequency and smooth-in-parameters approximations. The rationale behind our proposed hybrid method is threefold: (i) rational functions are better suited to address any potential lack of smoothness in the QoIs; (ii) rational functions can often achieve spectral approximation accuracy, much higher than more "local" function classes like piecewise polynomials; (iii) univariate rational approximation can be endowed with very efficient adaptive sampling strategies,[19,20] in contrast to the more costly approaches for high-dimensional adaptivity, for example, based on locally adaptive sparse grids.[21,18,22] Some numerical tests on the last two above-mentioned items are carried out in Section 5.1.

Hybrid rational/smooth approximation methods were also developed in References 23 and 24. However, several aspects distinguish our present work from those:

- In Reference 23, the focus is set on tracking how singularities with respect to frequency (the *resonating frequencies* of the target system) evolve as functions of the parameter. In the present setting, such resonant behavior is generally absent (and, if present, is not too relevant for the sake of approximation accuracy), see Section 5, thus making the method from Reference 23 misplaced. See Remarks 2 and 4 for further discussion on this.
- In Reference 24, it is assumed that the underlying dynamics are those of a quadratic dynamical system. This allows the authors to develop *intrusive* error estimators to drive a *non-localized* adaptive sampling. This is in contrast to our *localized* adaptivity, driven by a fully *non-intrusive* error indicator, which can be applied to dynamics with arbitrary nonlinearities.

2 | FREQUENCY-DOMAIN PROBLEM FRAMEWORK

We start in the time domain and consider a time-evolving physical system, possibly subject to some input. Such a system may be modeled as a *parametric* time-invariant nonlinear ordinary differential equation system (ODES)

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}(t;\mathbf{p}) = \boldsymbol{\phi}(\mathbf{x}(t;\mathbf{p});\mathbf{p}), \tag{1}$$

endowed with suitable initial conditions. The *state* $\mathbf{x} \in \mathbb{R}^n$ is a collection of n *degrees of freedom*, which characterize the instantaneous configuration (e.g., positions, velocities, temperatures, pressures, voltages, etc.) of the system. The right-hand side $\boldsymbol{\phi}$ drives the evolution of the state in response to both internal effects and external forces, including the system's dependencies on a set of input parameters $\mathbf{p}$ having a practical significance to the application. Typical examples of parameters are system design (e.g., geometry and material properties) and operational variables (e.g., external forces and boundary conditions). We note that one may be interested in more general systems, for example, including higher-order time derivatives. Still, one can often cast such problems as first-order ODES like Equation (1), for example, by augmentation. While, for convenience, we proceed assuming a first-order structure, it is of no consequence to the applicability of our algorithm.

Simulating the time-evolution of the state $\mathbf{x}$, for example, by discretizing Equation (1) with time-stepping schemes, can be demanding, due to (i) the large size n of the state, (ii) the fine time-resolution needed to follow the (nonlinear) behavior of $\boldsymbol{\phi}$, and (iii) the potentially long simulation time to reach a steady state or periodic limit, if damping is low. These issues can, at least in part, be counteracted by moving to the frequency domain. To this aim, direct transformation of $\boldsymbol{\phi}$ to the frequency domain can be done by, for example, Fourier or Laplace transforms. Notable for their frequent usage

in solving engineering problems are a subset of perturbation-based methods that introduce small oscillatory excitations in $\boldsymbol{\phi}$. These perturbation techniques often leverage assumptions about the form of $\mathbf{x}$, for example, (quasi-) periodicity, to aid the simplification. Overall, obtaining the frequency-domain form from the time-domain one can be a challenging task for general nonlinear systems. However, this has been the subject of extensive research. For instance, we refer to References 25–27 and 28,29,12 for reviews and examples of transform- and perturbation-based methods, respectively. Finally, we note that, sometimes, the problem is cast in the frequency domain not just out of convenience, but because frequency-domain quantities are the ultimate objective of the analysis.

In a first-order perturbation method, one assumes that the state is a small *harmonic* perturbation of a stationary state $\mathbf{x}_0$, namely,

$$\mathbf{x}(t;\mathbf{p}) \approx \mathbf{x}_0(\mathbf{p}) + \mathrm{Re}\left(\mathbf{x}_1(\omega;\mathbf{p})e^{i\omega t}\right). \tag{2}$$

The amplitude $\mathbf{x}_1$ of the perturbation term is complex to account for possible phase shifts. Note that, while $\mathbf{x}_1$ depends on the perturbation frequency $\omega \in \mathbb{R}$, $\mathbf{x}_0$ does not. Further harmonic perturbation terms with different frequencies may be added in the expansion of $\mathbf{x}$, giving rise to higher-order perturbation methods.

This expansion is then plugged into Equation (1). Under the assumption of a small perturbation $\mathbf{x}_1$, a first-order Taylor expansion of $\boldsymbol{\phi}$ is then performed, resulting in a problem of the form

$$\begin{cases} \boldsymbol{\phi}(\mathbf{x}_0(\mathbf{p});\mathbf{p}) = 0, \\ \boldsymbol{\psi}(\mathbf{x}_1(\omega;\mathbf{p}),\omega;\mathbf{x}_0(\mathbf{p}),\mathbf{p}) = 0. \end{cases} \tag{3}$$

The first equation characterizes stationary equilibrium states, whereas the second equation is frequency-dependent and identifies the perturbation amplitude *in function of the equilibrium* $\mathbf{x}_0$. Additional terms in higher-order perturbation methods would be identifiable through further equations, related to higher-order terms in a Taylor expansion of $\boldsymbol{\phi}$.

In most applications, the target of the system's analysis is not the state $\mathbf{x}$, but rather a set of *quantities of interest* (QoIs), functions of $\mathbf{x}$ whose choice is driven by the specific application. For instance, the QoIs may characterize the (averaged) boundary values of the system state (such as position and pressure values for a fluid-structure interaction[30]) at an interface or interconnection point with other systems. Other examples of QoIs are point evaluations of the state, to be compared with real-time sensor measurements in the context of, for example, feedback control or system-health monitoring.[31]

Generally, QoIs depend on the whole $\mathbf{x}$, that is, in the context of Equation (2), on both equilibrium state $\mathbf{x}_0$ and perturbation $\mathbf{x}_1$. For this reason, QoIs are, in general, functions of both perturbation frequency ω and parameters $\mathbf{p}$, and their computation requires solving both equations in Equation (3). We denote generic QoIs by

$$\xi : \mathbb{R}^{1+n_p} \ni (\omega,\mathbf{p}) \mapsto \xi(\omega,\mathbf{p}) \in \mathbb{C}^{n_\xi}. \tag{4}$$

Multiple QoIs are allowed, by letting $\xi(\omega,\mathbf{p})$ be vector-valued ($n_\xi > 1$).

This being said, for our later discussion, it is important to remark that, in some cases, QoIs may depend on the equilibrium state $\mathbf{x}_0$ and parameters $\mathbf{p}$ only. For instance, in some applications, for example, predator-prey or oscillator models, the equilibrium $\mathbf{x}_0$ is a QoI in itself. In a similar way, it is sometimes of interest to compute time-averaged quantities (over infinite time intervals), for which the oscillating component of Equation (2) disappears. See Section 4 for further examples. The corresponding QoIs are functions of $\mathbf{p}$, but not of ω. Moreover, we note that evaluating these QoIs requires solving only the first equation in Equation (3), and not the second one. We will denote such frequency-independent QoIs by

$$\zeta : \mathbb{R}^{n_p} \ni \mathbf{p} \mapsto \zeta(\mathbf{p}) \in \mathbb{C}^{n_\zeta}. \tag{5}$$

Again, $\zeta(\mathbf{p})$ is allowed to be vector-valued ($n_\zeta > 1$).

> *Remark* 1. Above and in the rest of this work, we assume that we are able to compute QoIs at arbitrary values of ω and $\mathbf{p}$, by (i) expensively solving the frequency-domain problem in Equation (3) and then (ii) post-processing the computed $\mathbf{x}$. In some situations, instead of solving a problem in the frequency domain, it is possible to obtain frequency-domain information by discrete Fourier transform of data gathered in the time domain (see, e.g., Reference 32). For instance, this might be relevant in applications where a frequency-domain model is out of reach, for example, because it is too complex to obtain or to simulate.

3 | APPROXIMATION SETUP

Consider the abstract setup introduced in the previous section. We are interested in finding an approximation for the two following functions: the generic QoIs $\xi : \mathbb{R}^{1+n_p} \to \mathbb{C}^{n_\xi}$ and the frequency-independent QoIs $\zeta : \mathbb{R}^{n_p} \to \mathbb{C}^{n_\zeta}$, see Equations (4) and (5). (To keep our presentation as general as possible, we even allow n_ξ or n_ζ to be 0, in case only one type of QoIs is relevant.) We wish the respective approximations of ξ and ζ, which we denote by $\widetilde{\xi}$ and $\widetilde{\zeta}$ respectively, to be accurate for frequencies $\omega \in \Omega \subset \mathbb{R}$ and for parameters $\mathbf{p} \in \mathcal{P} \subset \mathbb{R}^{n_p}$, with Ω and $\mathcal{P}$ being the (application-dependent) frequency and parameter ranges, respectively.

Due to its additional dependence on ω, the approximation of ξ requires special care. This behooves us to use a non-standard framework, taking inspiration from the "marginalized" approximation strategy described in Reference 23. The main idea is to treat ω and $\mathbf{p}$ separately. First, one selects T distinct *collocation* parameter points $\mathbf{p}_1, \ldots, \mathbf{p}_T \in \mathcal{P}$. At each collocation parameter $\mathbf{p}_j$, a *local* surrogate for ξ is built: by fixing $\mathbf{p} = \mathbf{p}_j$, we are left with the rather easier task of approximating a function of ω only. This step is carried out by *minimal rational interpolation* (MRI), see Section 3.1. We denote the corresponding approximation by $\widetilde{\xi}_j : \Omega \to \mathbb{C}^{n_\xi}$, which should satisfy $\widetilde{\xi}_j(\omega) \approx \xi(\omega, \mathbf{p}_j)$ for all $\omega \in \Omega$.

Then, the *global* surrogate is constructed by combining the local surrogates, ultimately resulting in an approximation of the form

$$\widetilde{\xi}(\omega, \mathbf{p}) = \sum_{j=1}^{T} \widetilde{\xi}_j(\omega)\varphi_j(\mathbf{p}). \tag{6}$$

In the expression above, for all j, $\varphi_j : \mathcal{P} \to \mathbb{R}$ is a weight function, responsible for combining the different local surrogates. Note that, in order to recover the local surrogates at all collocation points, that is, to have

$$\widetilde{\xi}(\omega, \mathbf{p}_j) = \widetilde{\xi}_j(\omega) \approx \xi(\omega, \mathbf{p}_j) \text{ for all } \omega \in \Omega \text{ and all } j, \tag{7}$$

the weight functions should satisfy a Lagrange property: $\varphi_j(\mathbf{p}_j) = 1$ and $\varphi_j(\mathbf{p}_k) = 0$ if $j \neq k$. For instance, multivariate piecewise-linear "hat" functions and Lagrange polynomials are candidate families of weight functions.

Remark 2. As already mentioned, our approach differs from the one in Reference 23 because our focus is on ξ, not on its poles and residues. More explicitly, we see from Equation (6) that the global surrogate $\widetilde{\xi}$ is obtained by superposition of the local surrogates $\widetilde{\xi}_j$. Instead, Reference 23 suggests building $\widetilde{\xi}$ through a superposition of poles and residues of $\widetilde{\xi}_j$, namely,

$$\widetilde{\xi}_j(\omega) = \sum_i \frac{\tilde{r}_j^i}{\omega - \tilde{\lambda}_j^i} \quad \rightsquigarrow \quad \widetilde{\xi}(\omega, \mathbf{p}) = \sum_i \frac{\sum_{j=1}^T \tilde{r}_j^i \varphi_j(\mathbf{p})}{\omega - \sum_{j=1}^T \tilde{\lambda}_j^i \varphi_j(\mathbf{p})}.$$

In Reference 23, this idea is justified by the approximation target having poles in the frequency range Ω, thus making an accurate identification of such poles a priority.

In our setting, we assume the approximation target ξ to be sufficiently smooth over Ω (more details on this assumption can be found in Remark 4). Accordingly, the poles $\tilde{\lambda}_j^i$ lose importance and meaning. In effect, tracking the evolution of the poles with respect to $\mathbf{p}$ would result in spurious and numerically unstable effects, whereas Equation (6) is accurate and stable.

Constructing an approximation for ζ is usually easier, since ω is absent. In this work, for simplicity, we employ a structure that is rather similar to that of $\widetilde{\xi}$ in Equation (6), namely,

$$\widetilde{\zeta}(\mathbf{p}) = \sum_{j=1}^{T} \zeta(\mathbf{p}_j)\varphi_j(\mathbf{p}). \tag{8}$$

Note that, while $\widetilde{\xi}$ is obtained as a $\mathbf{p}$-superposition of *approximations* $\widetilde{\xi}_j$, $\widetilde{\zeta}$ is built as a $\mathbf{p}$-superposition of *exact* values $\zeta(\mathbf{p}_j)$. Also, we remark that, since the weight functions φ_j are assumed to satisfy a Lagrange property, we recover exact interpolation: $\widetilde{\zeta}(\mathbf{p}_j) = \zeta(\mathbf{p}_j)$ for all j. We note that a rather similar MOR approach for building $\widetilde{\zeta}$ is the one described in Reference 18, where radial basis functions are used as weight functions for interpolation.

In general, it makes sense to choose the number T of collocation points differently for $\widetilde{\xi}$ and for $\widetilde{\zeta}$, for example, if ξ depends on $\mathbf{p}$ in a more (or less) "complicated" way than ζ does. This might even correspond to completely disjoint sets of collocation points $\mathbf{p}_j$ for ξ and ζ, and to different weight functions φ_j. In presenting our approach, we mostly ignore this possibility to avoid an overly heavy notation. At the same time, we wish to mention here that our ultimate algorithm of choice, see Sections 3.2 and 3.3, usually employs different values of T for ξ and ζ.

Remark 3. Before proceeding, we wish to discuss the cost of evaluating the surrogates $\widetilde{\xi}$ and $\widetilde{\zeta}$ at new values of (ω and) $\mathbf{p}$. In the case of $\widetilde{\xi}$, we see from Equation (6) that we only need to evaluate T weight functions φ_j, which is very inexpensive, and T local surrogates $\widetilde{\xi}_j$, which, as we will show, are also very cheap to evaluate. In particular, their evaluation cost is independent of the cost of solving the high-fidelity problem.

On the other hand, the evaluation of $\widetilde{\zeta}$ requires evaluating the T weight functions, but also obtaining the T values of ζ at the collocation points. This second group of operations can be significantly sped up by employing an *offline-online* decomposition. More explicitly, since the T values $\{\zeta(\mathbf{p}_j)\}_{j=1}^{T}$ are independent of the new parameter $\mathbf{p}$ at which $\widetilde{\zeta}$ is to be evaluated, we can (expensively) compute them once and then store them. Afterwards, when $\widetilde{\zeta}$ is to be evaluated, it suffices to cheaply access the values $\zeta(\mathbf{p}_j)$ from memory.

3.1 | Minimal rational interpolation

Here we consider the problem of building a local surrogate $\widetilde{\xi}_j : \Omega \to \mathbb{C}^{n_\xi}$ at some collocation point $\mathbf{p}_j$: the target is $\widetilde{\xi}_j(\omega) \approx \xi(\omega, \mathbf{p}_j)$ for all $\omega \in \Omega$. For this, we employ the MRI method introduced in Reference 14, which we summarize here in its barycentric formulation.

Assume that S_j *support* frequencies $\omega_{j,1}, \dots, \omega_{j,S_j} \in \Omega$ are fixed. Note the presence of the index j, a reminder that different collocation points may have different support frequencies. In fact, we will soon outline how such support frequencies can be *adaptively* selected, in a process that may yield different results at each collocation point $\mathbf{p}_j$.

Once the costly samples $\{\xi(\omega_{j,i}, \mathbf{p}_j)\}_{i=1}^{S_j}$ have been computed, we define the local surrogate as

$$\widetilde{\xi}_j(\omega) = \sum_{i=1}^{S_j} \xi(\omega_{j,i}, \mathbf{p}_j) \underbrace{\frac{q_{j,i}}{\omega - \omega_{j,i}} \Bigg/ \sum_{k=1}^{S_j} \frac{q_{j,k}}{\omega - \omega_{j,k}}}_{\phi_{j,i}(\omega)}. \tag{9}$$

The *rational* weight functions $\phi_{j,i}$ are defined in *barycentric* form, with $q_{j,1}, \dots, q_{j,S_j} \in \mathbb{C}$ being coefficients that will be defined in the next paragraph. Note that this choice of weight functions satisfies the Lagrange property for any (nonzero) $q_{j,i}$, so that, in particular, we always have exact reconstruction of the samples: $\widetilde{\xi}_j(\omega_{j,i}) = \xi(\omega_{j,i}, \mathbf{p}_j)$ for all i.

The barycentric coefficients $q_{j,i}$ are computed by solving the minimization problem

$$\begin{aligned} \min_{q_{j,1}, \dots, q_{j,S_j}} \quad & \left\| \sum_{i=1}^{S_j} q_{j,i} \xi(\omega_{j,i}, \mathbf{p}_j) \right\| \\ \text{subject to} \quad & \sum_{i=1}^{S_j} |q_{j,i}|^2 = 1, \end{aligned} \tag{10}$$

where $\|\cdot\|$ denotes the Euclidean norm. We refer to Reference 33 for more details on why the barycentric coefficients are chosen as the solution of this problem and on how to easily compute them via singular value decomposition. Alternative approaches for finding the weights $q_{j,i}$ could be based, for example, on the Loewner framework[34] or the adaptive Antoulas–Anderson algorithm.[13]

The number S_j and the locations of the support frequencies $\omega_{j,i}$ are crucial for the approximation properties and efficiency of the MRI surrogate: too few samples yield an inaccurate approximation, whereas too many make the training of the local surrogate overly expensive, and can even result in numerical instabilities.[20,35] A practical and effective way of choosing the support frequencies is the *greedy* MRI (gMRI) method, introduced in Reference 19.

In gMRI, the set of support frequencies is incrementally built, starting from the two endpoints of the interval Ω. Each new support frequency ω_{j,S_j+1} is selected based on the current surrogate $\widetilde{\xi}_j$, specifically, based on the denominator of the weight functions $\phi_{j,i}$, see Equation (9). The termination condition is based on a user-defined tolerance tol_ω, which determines the desired accuracy in frequency of the local surrogate.

Remark 4. We have already mentioned, notably in Remark 2, that we assume the approximation target ξ to be sufficiently smooth with respect to frequency. Sometimes the smoothness of ξ follows from physical considerations involving the underlying dynamics, for example, in terms of stability or passivity. Namely, if the poles of ξ (the resonating frequencies of the system) are constrained away from the frequency range Ω, ξ will necessarily be well-behaved there. However, in more general settings, it may be a priori unknown whether ξ is smooth or not.

Our setting allows us to perform a heuristic "on-the-fly smoothness check": once the rational approximation $\widetilde{\xi}_j$ has been found, for example, by gMRI, it is possible to find its poles by standard linear algebra techniques.[13] Then, we may empirically classify the function as "smooth" if all the surrogate poles are sufficiently far away from Ω. Unfortunately, at this stage, it is the authors' opinion that the above heuristic classification is heavily problem-dependent, and that a case-by-case tuning of a "pole-distance threshold" for the "smoothness check" must be carried out under careful user supervision. A more thorough, systematic investigation of this idea, including also the development of countermeasures to be used if a lack of smoothness is detected, is out of the scope of the present work, but is envisioned as future research.

3.2 | Adaptive parameter sampling strategy

In this section, we restrict our attention to the approximation of ξ. More details on our strategy for building $\widetilde{\zeta}$ follow in Section 3.3. Also, for simplicity, from here onward we assume that the parameter range $\mathcal{P} \subset \mathbb{R}^{n_p}$ is a hypercube. This is the most common situation in practice, since a certain range is usually prescribed independently for each of the n_p parameter components $\mathbf{p} = (p_1, \dots, p_{n_p})$.

Building a good and efficient approximation $\widetilde{\xi}$ requires a careful choice of the T collocation points $\mathbf{p}_j$. To this aim, we take inspiration from the sampling approach described in References 18,23, adaptively selecting the collocation points by exploring $\mathcal{P}$ via a *locally adaptive sparse grid* (LASG) structure.

LASGs are able to explore hypercubes even in high dimension at a relatively quick pace. Some comparable methods are locally refined tensor grids and quasi-random (e.g., Halton points or Latin hypercube) sampling. See, for example, References 36,37. However, tensor grids suffer greatly from the so-called "curse of dimension" and, as a consequence, are much more inefficient than LASG even for modest n_p (e.g., > 2). On the other hand, quasi-random sampling is very competitive even in the high-dimensional case, but does not allow for localized adaptive refinements. We refer to References 21,18,22,23 for more details on LASGs.

In our sampling strategy, we rely on a non-intrusive predictor-corrector approach to "grow" the set of collocation points using a tree-like structure. Given a certain *training set* (i.e., the set of collocation points $\mathbf{p}_j$ where $\widetilde{\xi}_j$ is available), a disjoint *test set* is defined. For instance, the test set can be defined in a procedural way, as the "discrete LASG-neighborhood" of the training set. See sect. 3.4 of Reference 23 for more details.

A *prediction* step is carried out at each test point $\mathbf{p}_\star$: the current surrogate model (which is based on training points only) is evaluated at $\mathbf{p} = \mathbf{p}_\star$. The discrepancy between surrogate and high-fidelity model at $\mathbf{p}_\star$ is then quantified through the *prediction error* (see Equation 11 below). If the prediction error is larger than a user-defined tolerance tol_ξ, the test point $\mathbf{p}_\star$ is flagged for *correction*. This means that $\mathbf{p}_\star$ is added to the training points, thus deflating the prediction error at $\mathbf{p}_\star$ by the interpolation property in Equation (7). The test set is then automatically refined around $\mathbf{p}_\star$ for the next iteration. An example of the overall procedure is shown in Figure 1 for $n_p = 2$. We can indeed observe local refinements of the LASG near points that are flagged for correction. The algorithm terminates when the prediction error is smaller than the tolerance tol_ξ at all test points.

Remark 5. Within a prediction-correction iteration, each test parameter $\mathbf{p}_\star$ is explored independently. This means that the above strategy is amenable to a parallel implementation.

Above, by "prediction error" at some parameter point $\mathbf{p} = \mathbf{p}_\star$, we mean the following quantity:

$$e_{\star,\xi} = \sqrt{\frac{1}{S}\sum_{i=1}^{S} \frac{\left\| \widetilde{\xi}(\omega_i^{\mathrm{test}}, \mathbf{p}_\star) - \xi(\omega_i^{\mathrm{test}}, \mathbf{p}_\star) \right\|^2}{\left\| \xi(\omega_i^{\mathrm{test}}, \mathbf{p}_\star) \right\|^2}}. \tag{11}$$

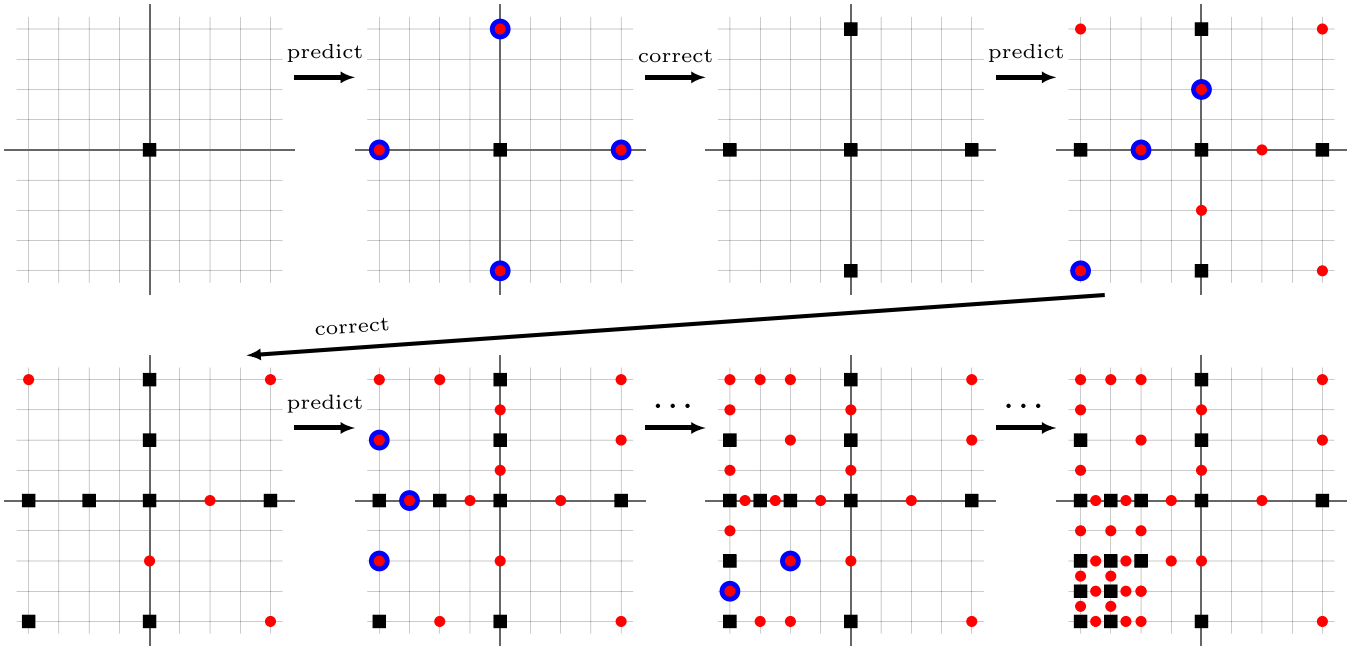

FIGURE 1 Example of evolution of the LASG in the adaptive algorithm for two parameters $\mathbf{p} = (p_1, p_2)$. The components p_1 and p_2 are on the x- and y-axes, respectively. The black squares are the current training set (the collocation points). The red circles are the test set (the points where the prediction error is computed). Points flagged for correction at each step are circled in blue.

Essentially, we look at the relative error between approximated and exact QoIs, taking the average (in the root-mean-square sense) over a set of user-defined "test frequencies" $\omega_1^{\text{test}}, \dots, \omega_S^{\text{test}} \in \Omega$.

Computing the prediction error by Equation (11) might be rather expensive if S is modest or large, since S high-fidelity samples of ξ must be computed. To alleviate this issue, we replace the exact QoI at $(\omega_i^{\text{test}}, \mathbf{p}_\star)$ with a proxy, namely, an *ad hoc* approximation $\widetilde{\xi}_\star(\omega_i^{\text{test}})$ computed by gMRI at $\mathbf{p}_\star$. If $\widetilde{\xi}_\star$ is a sufficiently good approximation of $\xi(\cdot, \mathbf{p}_\star)$ (notably, if it is closer to $\xi(\cdot, \mathbf{p}_\star)$ than $\widetilde{\xi}(\cdot, \mathbf{p}_\star)$ is), the resulting approximated prediction error

$$\bar{e}_{\star,\xi} = \sqrt{\frac{1}{S}\sum_{i=1}^{S} \frac{\left\| \widetilde{\xi}(\omega_i^{\text{test}}, \mathbf{p}_\star) - \widetilde{\xi}_\star(\omega_i^{\text{test}}) \right\|^2}{\left\| \widetilde{\xi}_\star(\omega_i^{\text{test}}) \right\|^2}} \tag{12}$$

can be expected to be close to the (expensive) exact one in Equation (11).

It is useful to note that, if a test point $\mathbf{p}_\star$ is flagged for correction, then the gMRI surrogate $\widetilde{\xi}_\star$ that we propose to use in Equation (12) has to be computed anyway, since it becomes one of the local surrogates in the expansion of the new $\widetilde{\xi}$, see Equation (6). With this observation, one can avoid repeated computations in the training phase.

> *Remark* 6. Since $\widetilde{\xi}$ is built based on the local surrogates $\widetilde{\xi}_j \approx \xi(\cdot, \mathbf{p}_j)$, it is crucial that such local surrogates be accurate enough. Otherwise, the prediction error might stagnate at an excessive value, resulting in a non-terminating $\mathbf{p}$-sampling loop. Importantly, if the local surrogates are computed by gMRI, we must set the tolerance tol_ω to a sufficiently smaller value than tol_ξ.

3.3 | Post-enrichment technique for frequency-independent QoIs

As already mentioned in Section 3, the approximation of ζ is fundamentally simpler than that of ξ. Indeed, no frequency modeling is necessary, and the approximation $\widetilde{\zeta}$ consists of a linear combination of values of ζ at some collocation points, see Equation (8). Still, computing the values $\zeta(\mathbf{p}_j)$ for $j = 1, \dots, T$ is a rather expensive task. We can speed it up thanks to the following observation: if $\xi(\omega, \mathbf{p}_j)$ has been computed at some ω, this means that the high-fidelity problem in Equation

(3) has been solved at the corresponding frequency and parameters. In particular, the steady state $\mathbf{x}_0(\mathbf{p}_j)$ has already been computed. As such, since evaluating ζ only requires post-processing of $\mathbf{x}_0$, the computation of $\zeta(\mathbf{p}_j)$ is very cheap, usually much less expensive than solving the high-fidelity problem from scratch.

Since our target is to obtain an approximation of *both* ξ and ζ, this allows us to employ the following strategy:

1. Build an approximation $\widetilde{\xi}$, for example, using the adaptive strategy from Section 3.2. This requires evaluating ξ at all the adaptively selected collocation points $\mathbf{p}_j$ for $j = 1, \dots, T$, at the corresponding support frequencies $\omega_{j,1}, \dots, \omega_{j,S_j} \in \Omega$.
2. At every collocation point $\mathbf{p}_j$, compute also $\zeta(\mathbf{p}_j)$. As mentioned above, depending on the implementation, this step might be very cheap, if $\zeta(\mathbf{p}_j)$ can be obtained by post-processing of the already available steady state $\mathbf{x}_0(\mathbf{p}_j)$.
3. Check if the resulting $\widetilde{\zeta}$, that is, the surrogate obtained by combining the samples of ζ at the collocation points of $\widetilde{\xi}$ (according to Equation 8), is accurate enough.
4. If $\widetilde{\zeta}$ is not accurate enough, add more collocation points. (Note that such additional points affect only $\widetilde{\zeta}$, and not $\widetilde{\xi}$!)

The addition of new collocation points for $\widetilde{\zeta}$ can be carried out by using the LASG approach from Section 3.2. However, in this case, since ω is absent, the prediction error, see Equation (11), can be more easily computed as the relative approximation error

$$e_{\star,\zeta} = \left\| \widetilde{\zeta}(\mathbf{p}_\star) - \zeta(\mathbf{p}_\star) \right\| \Big/ \left\| \zeta(\mathbf{p}_\star) \right\|. \tag{13}$$

Note that this error measure greatly penalizes inaccuracies at locations where ζ is small, since the denominator above approaches zero there. In some applications, one might consider applying a "regularized" relative error, for example,

$$e_{\star,\zeta,\delta} = \left\| \widetilde{\zeta}(\mathbf{p}_\star) - \zeta(\mathbf{p}_\star) \right\| \Big/ \max\left\{ \left\| \zeta(\mathbf{p}_\star) \right\|, \delta \right\},$$

for some (small) regularizing parameter $\delta > 0$. Similar considerations are relevant also for the error estimators used for $\widetilde{\xi}$, namely, Equations (11) and (12).

Note that a complementary strategy exists, where the surrogate $\widetilde{\zeta}$ is built first, and then $\widetilde{\xi}$ is obtained by adding more collocation points. However, in our numerical testing, approximating $\widetilde{\zeta}$ usually required more collocation points than $\widetilde{\xi}$. This explains why we present the first-$\widetilde{\xi}$-then-$\widetilde{\zeta}$ method only. Of course, depending on the application, the complementary approach may be preferable.

We summarize the overall procedure in Figure 2. In the diagram, we can see that, as described in Section 3.2, the training set grows from the center of $\mathcal{P}$ into a LASG, by progressively adding any test point $\mathbf{p}_\star$ with an excessive prediction error. Once the training phase of $\widetilde{\xi}$ ends, the final training set $\mathcal{P}_{\text{train},\xi}$ is selected as the initial training set $\mathcal{P}_{\text{train},\zeta}$ for $\widetilde{\zeta}$, which is then subject to adaptive refinements.

We note that, while training both $\widetilde{\xi}$ and $\widetilde{\zeta}$, test points $\mathbf{p}_\star$ with sufficiently small prediction errors might remain in the test set for several iterations of the predictor-corrector scheme. In such cases, the testing phase in Figure 2 requires building the local surrogate $\widetilde{\xi}_\star$ (or evaluating $\widetilde{\zeta}(\mathbf{p}_\star)$) over and over in the prediction-correction phase. These expensive operations can be avoided by storing any such $\widetilde{\xi}_\star$ (and $\widetilde{\zeta}(\mathbf{p}_\star)$). The memory requirements for this are extremely low, thus resulting in a net time save with basically no extra costs.

In Figure 2, we also employ the following trick to improve the accuracy of the surrogates. For both $\widetilde{\xi}$ and $\widetilde{\zeta}$, let us put ourselves at the end of the outer adaptivity loop, that is, when the LASG reaches convergence. The prediction error, as encoded by Equations (12) and (13), has just been computed at all test points. This means that both the "test surrogates" $\widetilde{\xi}_\star \approx \xi(\cdot, \mathbf{p}_\star)$ and the "test evaluations" $\zeta(\mathbf{p}_\star)$ are available. Such quantities can be included in the global surrogates $\widetilde{\xi}$ and $\widetilde{\zeta}$ by simply appending the test sets $\mathcal{P}_{\text{test}}$ to the training sets $\mathcal{P}_{\text{train}}$. This improves the accuracy of the two approximations with virtually no computational effort, since the added "expensive" values are already available in memory from the prediction-correction phase.

Before proceeding further, we also note that we provide an open-source MATLAB® version of the algorithm at https://github.com/pradovera/rational_lasg_matlab.

4 | FREQUENCY-DOMAIN BEARING MODEL

At this point, we are ready to explore the example of our approximation applied to the modeling of an aerodynamic bearing, specifically a *herringbone grooved journal bearing* (HGJB). Our focus in this and the next section is to establish a

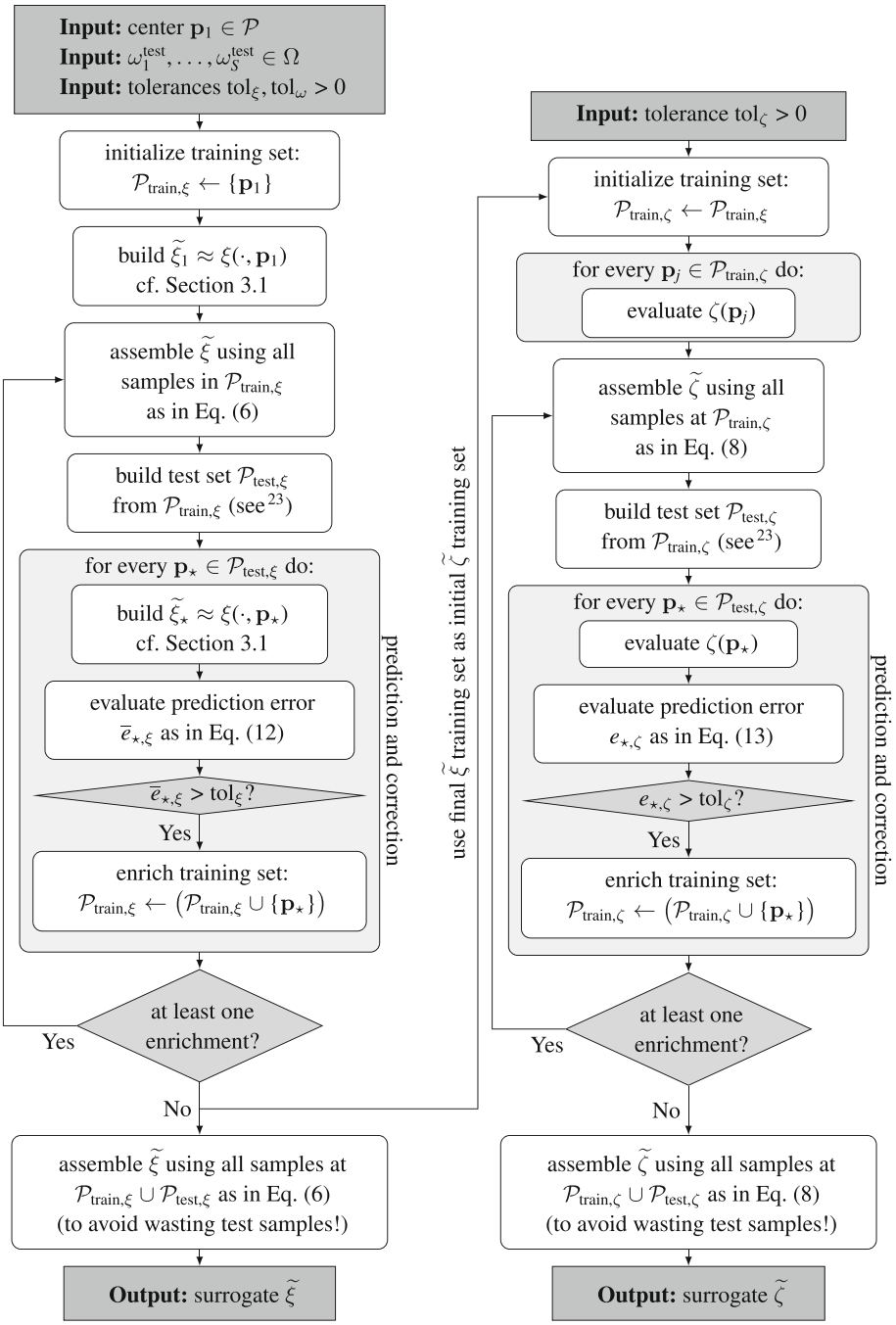

FIGURE 2 Adaptive parametric approximation of ξ, and of ζ by post-enrichment.

reference model in the framework of Section 2, referring readers to Reference 38 for a broader overview of the modeling of grooved bearings.

A schematic representation of an HGJB is shown in Figure 3 (top left). It consists of a segment of a larger rotor, rotating with angular speed ω_{rot} inside a bushing. A tight clearance between the parts (typically, on the order of microns) is occupied by the lubricating gas. The HGJB's distinguishing feature is the V-shaped groove (and ridge) pattern. This acts as a viscous pump, self-pressurizing the gas film through its own rotation, by forcing fluid toward the axial center from both ends, where ambient conditions exist. As with other aerodynamic bearings, the relative motion of two bearing parts (bushing and rotor) in HGJBs generates a pressure distribution in a thin gas film between their surfaces, thus acting as a cushion to resist contact between the bearing parts. Such bearings are characterized by a low-friction and oil-free operation, enabling small-scale high-speed turbo-machinery that is well suited for distributed energy conversion technology, for example, fuel cells or waste-heat recovery, needed for a sustainable energy transition. We refer to Reference 38

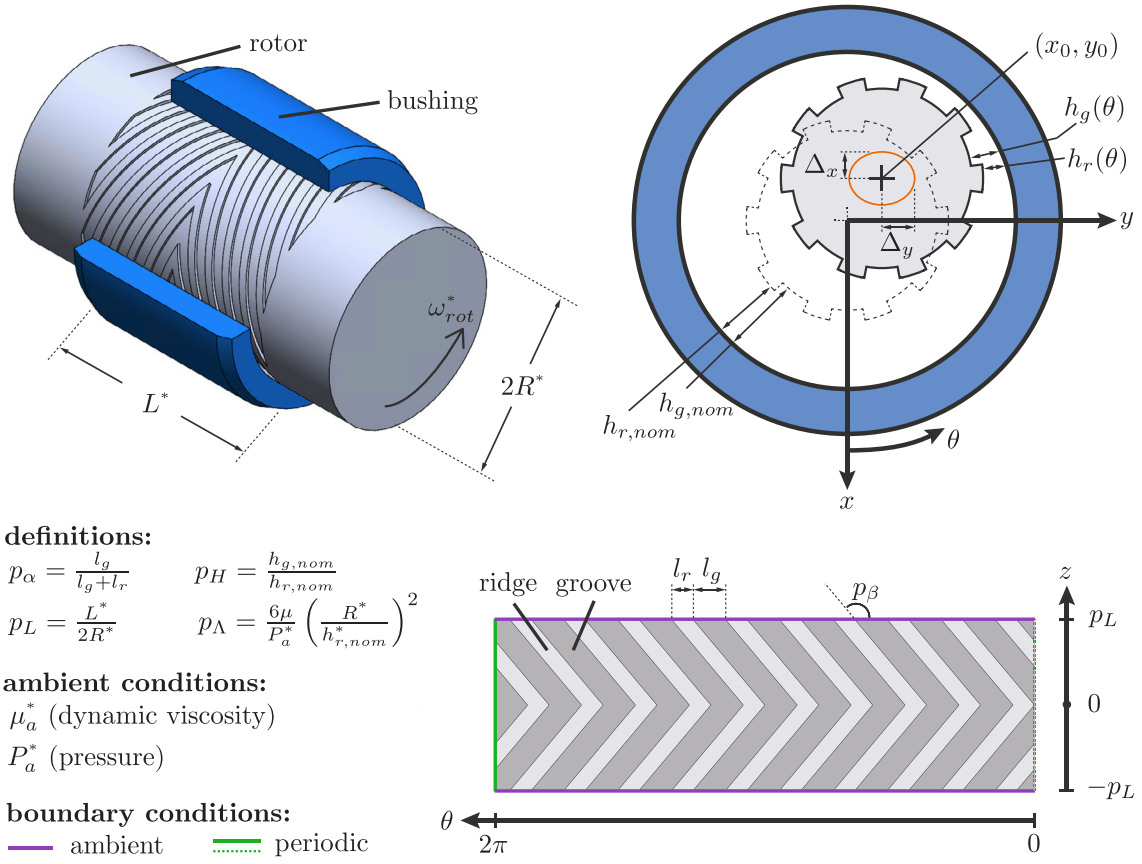

FIGURE 3 Definition of HGJB design and modeling parameters and a contextual view of a typical rotor system including two HGJB bearings.

for details on original works across various HGJB applications. Notably, compared to other aerodynamic bearings, HGJBs have the advantage of higher stiffness and load capacity. In applications, for example, in the transportation sector, gas-bearing-supported rotors also see significant external loading, including shock and vibration. Predicting bearing behavior over a range of design and operational parameters is challenging, especially in the contexts of (near-)real-time stability or response prediction during machine operation.

For our purposes, the design and operation of the HGJB can be completely described in nondimensional terms. The translating rotor has a trajectory inside the bearing, whose position, depicted in the section view, is given in bushing-fixed Cartesian (x, y)-coordinates with the origin at bearing's center (Figure 3). With the characteristic length in this frame is taken as the concentric ridge clearance $h_{r,nom}$, the magnitude of the rotor position vector cannot exceed unity, which would correspond to the (undesired) contact between the rotor and the bushing. We describe any bearing fluid film location using bushing-fixed cylindrical (θ, z)-coordinates, with z aligned with the rotor's rotation vector. The grooves are designed with a symmetric V-shape in the $\theta - z$ plane, parametrized by the angle of the groove p_β, the width ratio of the groove p_α, groove-ridge clearance ratio p_H, and the length of the bearing p_L, as depicted. It is worth mentioning that an eccentric rotor has θ-dependent local film thicknesses $h_r(\theta)$ and $h_g(\theta)$. Finally, the compressibility number p_Λ describes the combined effects of rotor speed and the fluid properties at a given operating point, and is of utmost importance for both design and operational purposes. In Figure 3, we refer to a few characteristic dimensional quantities, for which we use the superscript "$\star$", to ground our nondimensional description to a physical bearing.

4.1 | Time-domain model

At present, modeling of generic aerodynamic bearings requires solving a suitable adaptation of the Reynolds equation, the classical governing equation describing the pressure distribution within the fluid film. More specifically, in the case of

HGJBs, the groove-ridge interfaces in the θ-direction lead to discontinuities in the local film thickness $h(\theta)$ (see Figure 3). This makes it necessary to employ a very fine spatial (and temporal) meshing in numerical implementations, resulting in high simulation costs.

As an alternative, we employ the narrow groove theory (NGT), pioneered in the 1950s by Whipple.[39,40] The NGT allows a convenient modification of the Reynolds equation for bearings with periodic grooving features in the θ-direction. Two assumptions lead to the NGT result:[41] (i) locally incompressible fluid, giving a triangular pressure profile across a groove-ridge pair, and (ii) an infinite number of grooves, which transforms the saw-toothed pressure profile into a smooth one. The NGT equation describes the smooth pressure distribution with a second-order nonlinear partial differential equation given by

$$\begin{aligned} 2p_\Lambda \partial_t(a_5 P) = \; & \partial_\theta(P(a_1\partial_\theta P + a_2\partial_z P)) + \partial_z(P(a_2\partial_\theta P + a_3\partial_z P)) \\ & + a_s\big(\sin(p_\beta)\partial_\theta(a_4 P) - \cos(p_\beta)\partial_z(a_4 P)\big) - p_\Lambda \partial_\theta(a_5 P), \end{aligned} \tag{14}$$

where P is the pressure field, while the coefficients $a_d(\theta)$ with $d = 1, \ldots, 5$ and a_s are defined in Appendix A, along with the nondimensionalization scheme used. Refer to Figure 3 for definitions of parameters $\mathbf{p}$. As boundary conditions, we impose ambient pressure at the two axial edges of the bearing.

After spatial discretization of the fluid film and numerical approximation (typically, by finite-difference or-element methods) of the spatial partial derivatives, we can transform Equation (14) into the form of the first-order ODE system in Equation (1). We refer to Reference 42 for more details on the discretization. In our simulations below, we discretize the fluid film on a 65×65 rectangular grid in the (θ, z)-plane using a centered-differences scheme.

In the interest of studying the rotordynamics, the time-domain ODE system can be augmented with the equations of motion for the relevant inertial components. Specifically, in a fluid-structure-interaction fashion, our rotor is subject to the pressure of the fluid film, so that the total force is obtained by integration of P, to be added to any external forces (e.g., due to gravity).

4.2 | Frequency-domain model

To obtain a frequency-domain version of the bearing model, we follow References 43,42. Consider again Figure 3 with the rotor in an arbitrary equilibrium position (x_0, y_0) due to a static loading (e.g., inertial forces, gravity, etc.). The rotor is then perturbed to yield simple harmonic motion (with frequency ω) around the equilibrium position. Independent perturbations in the x and y directions are allowed, with magnitudes Δ_x and Δ_y respectively, which are complex to account for phase shifts. Specifically, the local film thickness takes the time-harmonic form

$$h_r(\theta, t) \approx 1 - \big(x_0 + \mathrm{Re}\big(\Delta_x e^{i\omega t}\big)\big)\cos(\theta) - \big(y_0 + \mathrm{Re}\big(\Delta_y e^{i\omega t}\big)\big)\sin(\theta). \tag{15}$$

As a consequence of this expansion, it is assumed that the total pressure solution takes the form

$$P(\theta, z, t) \approx P_0(\theta, z) + \mathrm{Re}\big(\big(\Delta_x P_{1,x}(\theta, z) + \Delta_y P_{1,y}(\theta, z)\big)e^{i\omega t}\big). \tag{16}$$

Note the presence of two independent perturbation terms $P_{1,x}$ and $P_{1,y}$, whose sum (weighted by Δ_x and Δ_y) gives the total first-order pressure perturbation. As the notation suggests, they correspond to perturbations of h_r in the x and y directions, respectively.

After injecting Equations (15) and (16) into (14) and keeping only terms up to order $\mathcal{O}(\Delta_x + \Delta_y)$, we can split the resulting equation into three separate equations by collecting and balancing coefficients of Δ_x and Δ_y (see Reference 42 for the detailed equations):

- A nonlinear steady-state equation, with no analytical solution, is given by zeroth-order terms. Its solution is the equilibrium point P_0 in Equation (16), whose discrete version is indicated by $\mathbf{x}_0$ in Equation (3). To find it, we use the Newton method after spatial discretization, and this accounts for a significant portion of the computation time.
- Two ω-dependent linear equations contain first-order perturbation terms, corresponding to Δ_x and Δ_y, respectively. Their solutions are the ω-dependent first-order perturbation fields $P_{1,x}$ and $P_{1,y}$. These equations are linear and thus, after spatial discretization, can be solved more directly by an off-the-shelf linear system solver. Still, it is required to inject the equilibrium $\mathbf{x}_0$, which must therefore be precomputed.

Note that varying the parameters $\mathbf{p}$ will generally affect all the above-mentioned equations. As such, for each parameter value of interest, it is necessary to assemble and solve the nonlinear equation, and then assemble and solve the linear equations at each excitation frequency ω of interest.

Remark 7. The frequency-domain formulation of the bearing described above is not merely a matter of modeling convenience. The typical axisymmetric bearing geometry and (quasi-)periodic nature of system excitations (e.g., rotor unbalance, external vibrations, self-excited whirl) contribute to strongly frequency-dependent responses. This observation is supported by experimental studies for the HGJB, for example, in References 44,45.

Until now, we have not yet specified how to quantify the "bearing behavior" that we wish to model with respect to the parameters $\mathbf{p}$. We do this through specific QoIs, which are used to assess the bearing's static and dynamic performance. These QoIs are extracted from the fluid film pressure solution via a post-processing step. For assessing the bearing's dynamic performance near its static equilibrium, we model the rotor as a 2D nonlinear harmonic oscillator with complex direct and cross-coupled impedances computed by integration (practically, by numerical quadrature) of the perturbed pressure field as

$$\mathrm{Z} = \begin{bmatrix} \mathrm{Z}_{xx} & \mathrm{Z}_{xy} \\ \mathrm{Z}_{yx} & \mathrm{Z}_{yy} \end{bmatrix} = -\int_{-p_L}^{p_L}\int_{0}^{2\pi} \begin{bmatrix} P_{1,x}\cos\theta & P_{1,x}\sin\theta \\ P_{1,y}\cos\theta & P_{1,y}\sin\theta \end{bmatrix} d\theta dz. \tag{17}$$

Decomposing entries of Z into real and imaginary parts ($\mathrm{Z} = \mathrm{K} + i\omega\mathrm{C}$) leads us directly to eight real coefficients, interpreted as the dynamic stiffness and damping of the bearing. The evolution of the impedance matrix with respect to frequency ω and design/operating parameters $\mathbf{p}$ can be directly used to obtain critical information needed in harmonic-response calculations and in stability analyses. When considered in the context of more complex rotordynamic systems, for example, with multiple bearings and/or system degrees-of-freedom, these bearing dynamic coefficients can also be integrated into the respective impedance matrices of the larger rotordynamic system for more sophisticated performance analyses.

Moreover, when the system is at the static equilibrium at (x_0, y_0), the force transmitted to the rotor due to the static pressure distribution exactly offsets the external static load applied. Having solved the equilibrium equation for the static pressure, we can obtain the entries of the static force vector by integration (in practice, quadrature) of the rise in the static pressure as

$$\mathbf{f} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = -\int_{-p_L}^{p_L}\int_{0}^{2\pi} (P_0 - 1) \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} d\theta dz. \tag{18}$$

The load that the bearing can support at equilibrium provides practitioners with useful information on the bearing's static performance. Specifically, a large magnitude of the static force $\mathbf{f}$ means high bearing load capacity, which is generally desirable to keep the rotor operating safely.

A high-level process summary for obtaining the QoIs from the reference model is included in Figure 4.

Remark 8. Above, to obtain a frequency-domain formulation, we have perturbed the NGT equations around a *generic* equilibrium rotor position. In the literature, it is common to find perturbation methods that act around the bearing's concentric position only. Such models, which we refer as "concentric NGT" (cNGT) models, trade accuracy to gain computational efficiency. Specifically, cNGT models predict the bearing stiffness and damping properties (cf. Equation 17) as functions of perturbation frequency ω and parameters $\mathbf{p}$ without consideration of static eccentricity effects. Such a concentric analysis generally yields a pessimistic stability and load capacity estimation. The timings of an efficient implementation of cNGT based on the theory of Reference 46 are compared with our surrogate in Section 5.

5 | NUMERICAL RESULTS

In many applications (e.g., in optimal design or real-time control), one needs efficient and accurate prediction tools for bearing behavior in response to a variety of sources of excitation. However, the nonlinear frequency-domain model

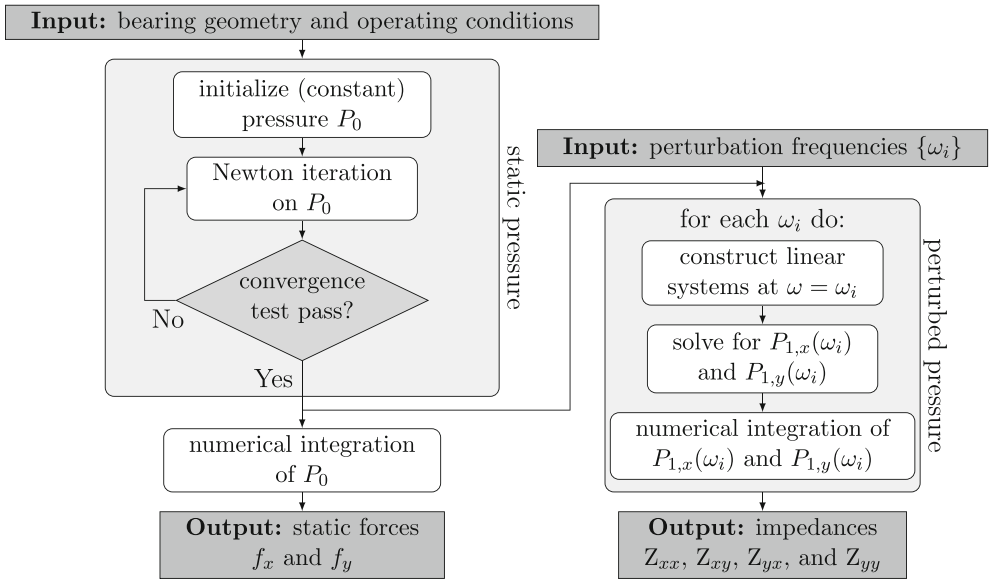

FIGURE 4 Eccentric NGT HGJB reference model numerical procedure.

described in the previous section is often too expensive to be directly applied in such multi-query settings. We propose to replace the high-fidelity and high-cost model with a surrogate one, obtained with our proposed adaptive strategy from Section 3. In the following sections, some experiments are carried out in this direction, for several choices of parameters $\mathbf{p}$. Our numerical tests were carried out in MATLAB® R2022b on a desktop computer with a 6-core 3.50 GHz Intel® processor.

In all cases, the QoIs are the impedance parameters and the static forces, see Equations (17) and (18). More precisely, we seek approximations for the frequency-and-parameter-dependent

$$\xi := [Z_{xx}, Z_{xy}, Z_{yx}, Z_{yy}]^\top : \mathbb{R}^{1+n_p} \to \mathbb{C}^4$$

and for the parameter-dependent

$$\zeta := \mathbf{f} = [f_x, f_y]^\top : \mathbb{R}^{n_p} \to \mathbb{R}^2.$$

While we test several input variable sets, perturbation frequency and eccentricity are common inputs in all our simulations, because they are the very essence of the eccentric perturbation frequency-domain model we wish to approximate, see Section 4. In our formulation, ω represents the nondimensional ratio between the (dimensional) excitation frequency and the rotor speed (see Appendix A). Considering the consequential nature of an unbalanced rotor, it is of practical relevance to ensure that the ω domain of interest includes the rotor frequency, that is, in nondimensional terms, when $\omega = 1$. Concerning eccentricity, for convenience we consider equilibrium eccentricities only in the right direction; that is, $x_0 \geq 0$ and $y_0 = 0$ in Equation (15). This can be done without loss of generality due to the bearing's symmetry.

5.1 | (1 + 1)D numerical results

For our first numerical experiment, we consider a frequency $\omega \in \Omega = [10^{-5}, 1.701]$ and a single parameter $\mathbf{p} := x_0 \in \mathcal{P} = [0, 0.95]$, the eccentricity in the x direction. The remaining parameters of the bearing model are set to their nominal values, which are reported in Table 1. Since Ω contains several orders of magnitude, we space the $S = 1001$ test frequencies $\omega_1^{\text{test}}, \ldots, \omega_S^{\text{test}}$ logarithmically over Ω. We set $\text{tol}_Z = \text{tol}_\mathbf{f} = 10^{-3}$ and $\text{tol}_\omega = \text{tol}_Z/3$, and we prescribe piecewise-linear "hat basis functions" for $\mathbf{p}$-interpolation (φ_j in Equations 6 and 8).

Our algorithm samples the parameter space $\Omega \times \mathcal{P}$ adaptively. The resulting sample points are shown in Figure 5 (left). With respect to frequency, we can observe that, except for the left endpoint $\omega = 10^{-5}$ (which is always sampled by gMRI),

TABLE 1 Nominal values and ranges of bearing parameters in numerical simulations.

		Range	
Parameter	**Nominal value**	**Min**	**Max**
ω	—	10^{-5}	1.701
x_0	0	0	0.95
p_Λ	25	1	40
p_α	0.65	0.3	0.7
p_β	19°	−50°	−10°
p_L	1.0	0.7	2.0
p_H	3.2	1.0	4.0

Note: See Section 4 for a more detailed description of the parameters and of their significance in the HGJB model.

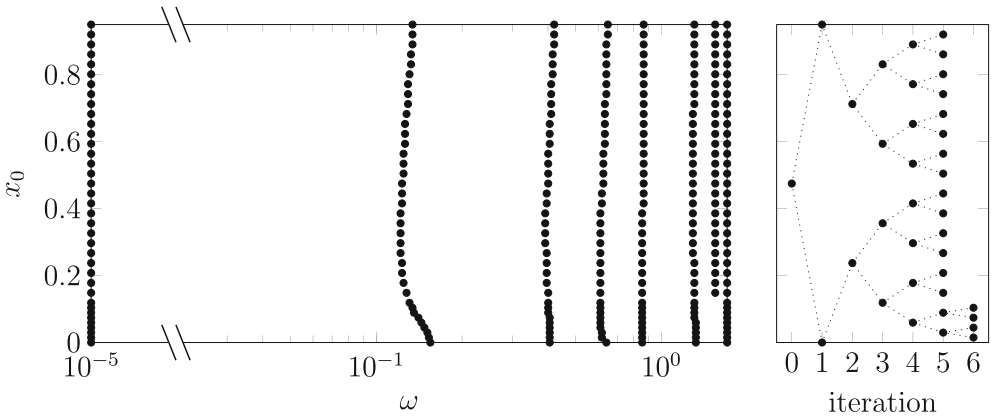

FIGURE 5 On the left: (ω, x_0) sample points. On the right: x_0 sample points as the Z predictor-corrector loop advances. The LASG tree structure is superimposed as dotted lines.

all the sampled frequencies are clustered at large frequencies, in the interval $[0.1, 1.701]$. Also, we note that the number and locations of the sampled frequencies depend on the value of x_0.

In Figure 5 (right), we can see the **p**-sample points that are added at each greedy iteration. For the first five iterations, we can observe uniform refinements, whereas, at the final iteration, localized refinements are performed around small values of x_0. This means that the approximation of Z is particularly difficult there, a fact that was unknown *a priori*.

To validate our approximation, we evaluate $\widetilde{\mathrm{Z}}$ on a uniform grid of 100×50 (ω, x_0)-points. There, we also expensively compute the exact Z. The approximated $\widetilde{\mathrm{Z}}$ is shown in Figure 6, where we focus on two stiffness entries ($\mathrm{K}_{xx} = \mathrm{Re}(\mathrm{Z}_{xx})$ and $\mathrm{K}_{xy} = \mathrm{Re}(\mathrm{Z}_{xy})$) and on two damping entries ($\mathrm{C}_{xx} = \mathrm{Im}(\mathrm{Z}_{xx})/\omega$ and $\mathrm{C}_{xy} = \mathrm{Im}(\mathrm{Z}_{xy})/\omega$). We show the absolute error $|\widetilde{\mathrm{Z}} - \mathrm{Z}|$, which is quite small in all cases, namely, below 10^{-3}.

As a final test to investigate the behavior of the employed LASGs, we look at the prediction error indicators $e_\star$ in Equations (11) and (13) over the parameter space $\mathcal{P}$. In Figure 7 (left), we evaluate $e_\star$ on a grid of new test points, which do not fall within the set of support parameters. We see that the error values are uniformly below the prescribed tolerance. Moreover, the gap between tolerance and actual error is rather small: for most values of x_0, the error is only around an order of magnitude smaller than the tolerance. This is actually a very beneficial property. Indeed, if the gap between tolerance and prediction error were larger, it would be an indication that our algorithm takes too many samples in the training phase, resulting in less efficient training, and also in more costly evaluation of the surrogate model.

Additionally, we recall that our algorithm is "aware" of such indicators only for values of x_0 in the training and test set. Moreover, the impedance prediction error is available only in "surrogate" form, see Equations (11) and (12). Nonetheless, the prescribed tolerance seems nicely attained over the whole parameter range.

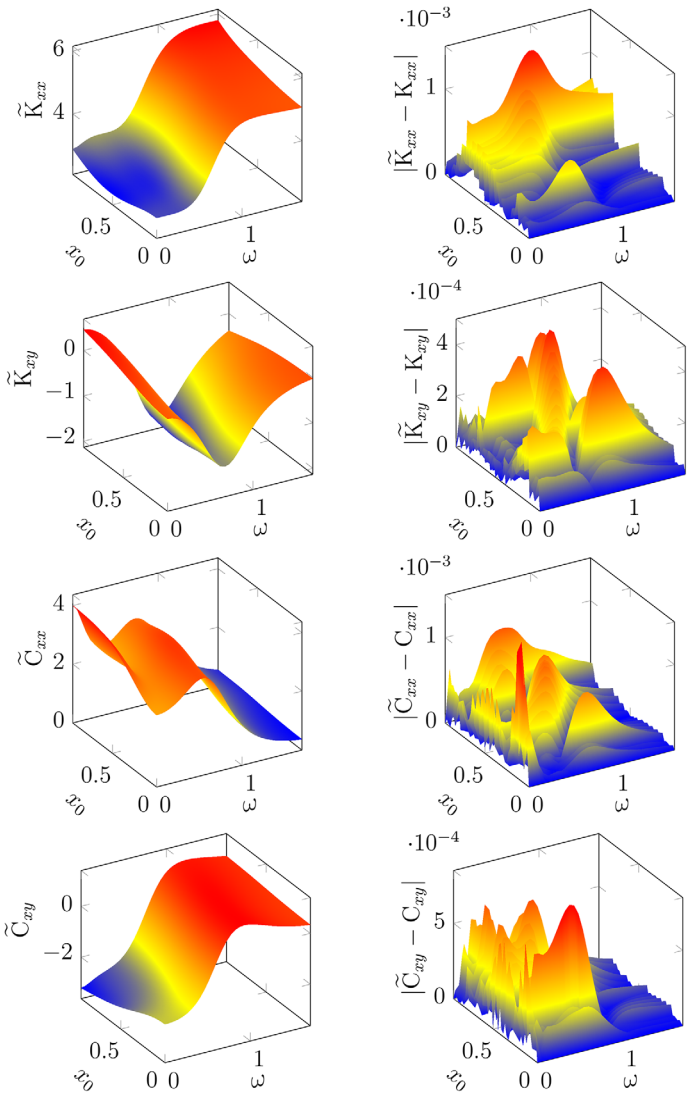

FIGURE 6 Predicted values and errors for some stiffness and damping entries.

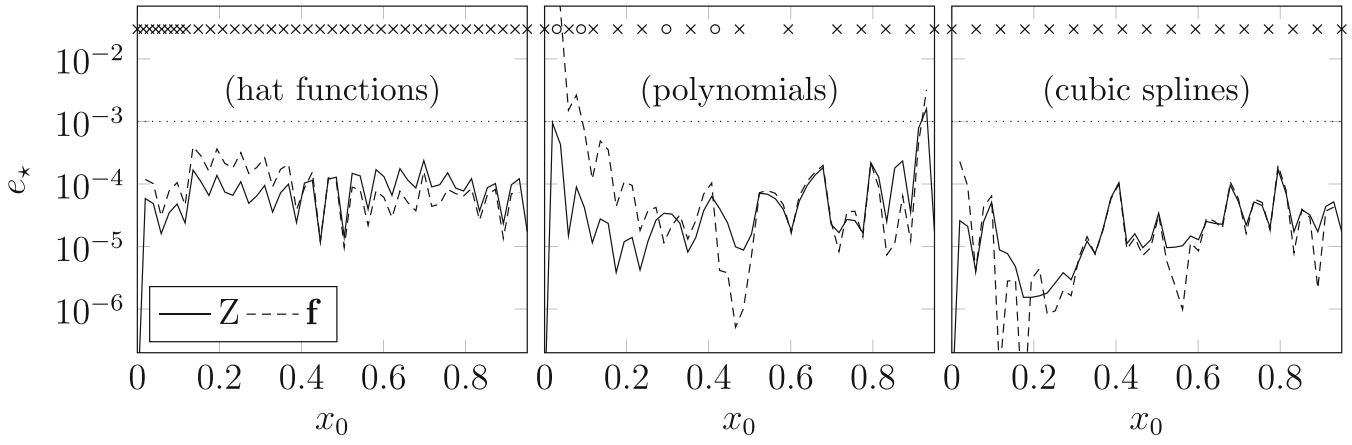

FIGURE 7 Validation errors for impedance and static forces. From left to right, the approximations $\widetilde{Z}$ and $\widetilde{\mathbf{f}}$ are obtained with piecewise-linear, polynomial, and spline **p**-interpolation. The tolerances are $\text{tol}_Z = \text{tol}_{\mathbf{f}} = 10^{-3}$, as denoted by the horizontal dotted line. On top of both plots, the locations of x_0 sample points are shown as crosses, with dots denoting points reserved for $\widetilde{\mathbf{f}}$.

5.1.1 | Comparing more (and less) global approximation strategies

We now want to discuss some of the consequences of our choice of interpolation strategy over the parameter x_0. We recall that, in order to obtain our results above, we have employed piecewise-linear hat functions. Such basis is local, in the sense that each local value, that is, $\widetilde{Z}_j$ for $\widetilde{Z}$ and $\mathbf{f}(\mathbf{p}_j)$ for $\widetilde{\mathbf{f}}$, affects the approximation only around the collocation point $\mathbf{p}_j$. This leads to some advantages: it can be used to approximate general non-smooth (but continuous) functions and it is cheap to evaluate. However, it also has one main disadvantage: we are building an approximation that is locally only linear with respect to x_0, which may not be too accurate.

Under smoothness assumptions, global bases can be employed to gain accuracy. To show this empirically, we repeat our experiments by replacing hat functions with global Lagrange polynomials. The training phase of our algorithm ends much earlier: only 17 samples of x_0 are needed to build the approximation, compared to the 37 needed with the previous setup. Despite the lower number of sample points, the approximation error (quantified by $e_{\star,Z}$ and $e_{\star,\mathbf{f}}$) is fairly similar to the previous case for most values of x_0, as can be seen in Figure 7 (middle).

However, we can clearly notice some Runge oscillations near both endpoints of $\mathcal{P}$, where the tolerance seems not to be attained. These numerical instabilities are intrinsically related to polynomial interpolation over scattered points. Specifically, (locally refined subsets of) uniformly spaced points may have extremely large Lebesgue constants.[47] As such, we could expect our algorithm, endowed with global polynomial $\mathbf{p}$-interpolation, to fail whenever the collocation points are unfavorably placed. Indeed, in such case, the numerical instabilities might cause the tolerance to never be attained, by introducing oscillations near the endpoints of the parameter domain.

With splines,[47] one can sometimes achieve higher accuracy than with local bases (like hat functions) and better stability than with global bases (like polynomials). To test this, we repeat our simulation, using cubic splines to interpolate over $\mathbf{p}$. The corresponding algorithm reaches convergence after exploring 17 $\mathbf{p}$-points, which is about half as many as with hat functions. The corresponding accuracy (again, quantified by $e_{\star,Z}$ and $e_{\star,\mathbf{f}}$) is shown in Figure 7 (right). It is uniformly below the tolerance, without showing any unstable behavior. This being said, splines cannot be easily generalized to higher dimensions ($n_p \geq 2$) if the sampling set is unstructured. Due to adaptivity, this makes splines unavailable in our method as soon as more than one parameter is present. In effect, this means that piecewise-linear hat functions remain the natural choice for a robust interpolation framework over general LASGs.

5.1.2 | Validation of a gMRI + LASG approximation

As a final test involving the (1 + 1)D model, we compare our method to the state-of-the-art non-intrusive adaptive algorithm from Reference 18. There, LASGs are used to adaptively sample over *all* parameters, that is, in our framework, over frequency and parameters alike. Effectively, this means that the rational approximation routine via gMRI is skipped, at the cost of increasing the number of parameters by 1. In comparing the two approaches, we refer to our method as "hybrid", since it combines rational and hat functions, whereas we dub the reference state-of-the-art method "joint".

To make the comparison more fair, in the joint method, we treat frequency on a linear scale, as opposed to the logarithmic one used for the hybrid approach. (Using a logarithmic scale would exacerbate the sampling issues that we are about to describe.) Moreover, since the joint approach does not require a "frequency tolerance", we use the larger "tol" value as tolerance for both frequency and eccentricity.

The joint greedy method requires eight iterations to converge, as opposed to the six needed by our hybrid approach. In turn, this corresponds to a set of 622 collocation (ω, x_0)-points (at 61 distinct eccentricity values), many more than the 287 (ω, x_0)-points (at 37 distinct eccentricity values) used by the hybrid approach. We show the collocation points of the joint method in Figure 8, which should be visually compared to Figure 5 (left), containing the samples of the hybrid approach. We can conclude that the joint method is less effective in achieving the target tolerance, when compared to our hybrid approach. In our view, there are two main reasons for this:

- When used to approximate the frequency dependence of the QoIs, hat functions, due to their locality and lower degree, converge more slowly than rational functions. From this point of view, our proposed hybrid method improves the joint one by increasing the approximation efficiency in one of the parameters, namely, frequency. Notably, in contrast to our observations concerning polynomials in Section 5.1.1, the chosen higher-order approximation strategy in frequency is numerically stable, due to the beneficial properties of the barycentric expansion, see Equation (9).

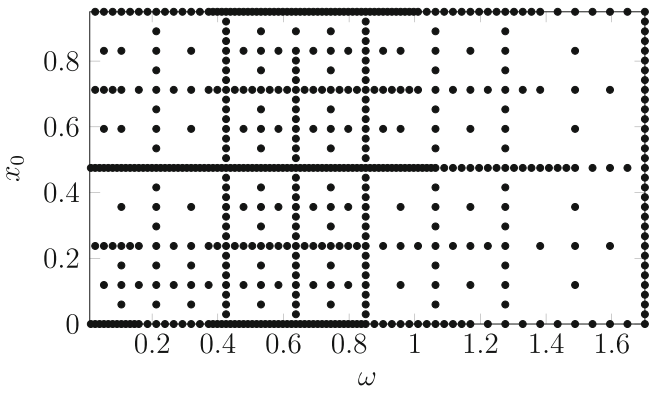

FIGURE 8 (ω, x_0) points sampled by the "joint" method.

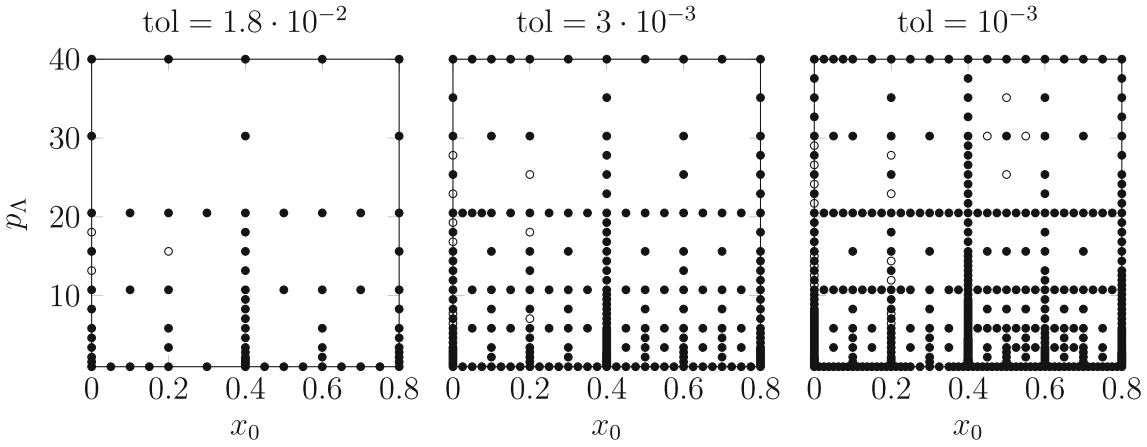

FIGURE 9 Adaptively chosen training **p**-points for three values of tol, which are reported on top of the plots. The white dots are added by the **f** predictor-corrector loop.

- The greedy MRI method performs adaptive sampling more effectively than LASGs. In particular, both gMRI and LASGs use a predictor-corrector approach, evaluating the current approximation error over a set of test parameter points. However, the size of the test set used in gMRI is smaller, consisting of a *single* point at each iteration. This enables a quicker adaptivity routine, which is more parsimonious in terms of total number of samples taken. (Of course, the main limitation of gMRI is its being limited to the approximation of *univariate* functions.)

5.2 | (1 + 2)D numerical results

We consider a slightly more complex setup with operational relevance. The compressibility number p_Λ is added as a parameter, on top of frequency ω and static eccentricity x_0: $\mathbf{p} := (x_0, p_\Lambda)$. The effect of a static loading on the bearing is captured by x_0, while p_Λ envelopes the combined effects of the specific fluid conditions and rotor speed at a given operating point. See Table 1 for the ranges of ω, x_0, and p_Λ, as well as for the nominal values of the remaining bearing parameters.

In this section, we specifically wish to investigate the effect of the tolerances (tol_Z, $\mathrm{tol}_\mathbf{f}$, and tol_ω) on the behavior of the algorithm, as well as on the quality of the resulting surrogate model. For simplicity, we fix the relationships $\mathrm{tol}_Z = \mathrm{tol}_\mathbf{f} =: \mathrm{tol}$ and $\mathrm{tol}_\omega = \mathrm{tol}/3$, so that we only have one free tolerance value to vary. Each value of tol requires an independent training, and yields a different surrogate model. We train six surrogates, corresponding to tolerance levels between 10^{-3} and $2.7 \cdot 10^{-2}$. For reference, in Figure 9 we display the sets of training **p**-points that are explored for some values of tol. We can observe anisotropic refinements near small values of p_Λ and mid-to-large values of x_0.

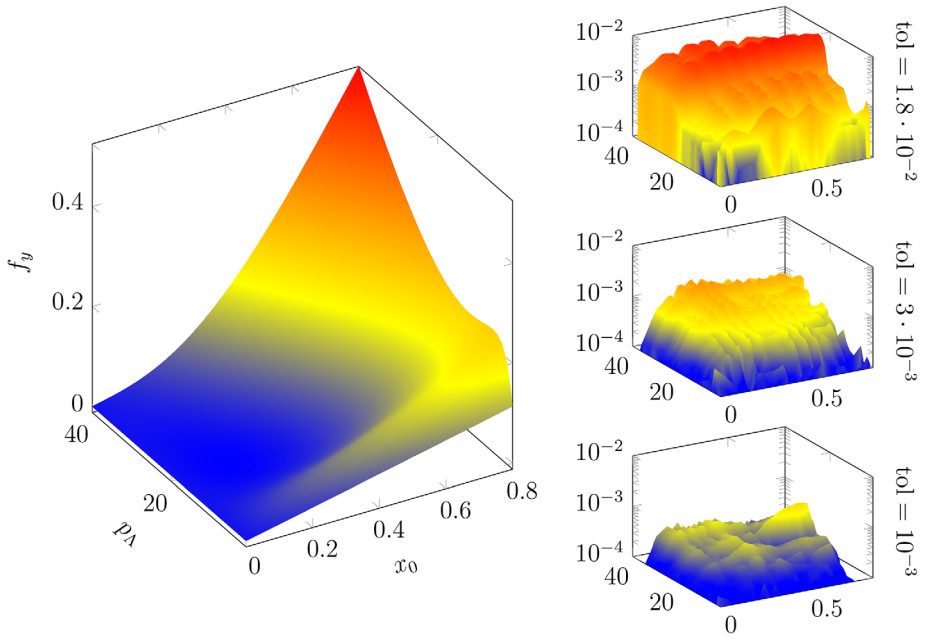

FIGURE 10 On the left: f_y computed with the high-fidelity training model. On the right: absolute errors $\left|\widetilde{f}_y - f_y\right|$ for surrogates obtained at different tol levels. The same z-scale and color scale are used for all plots on the right-hand side.

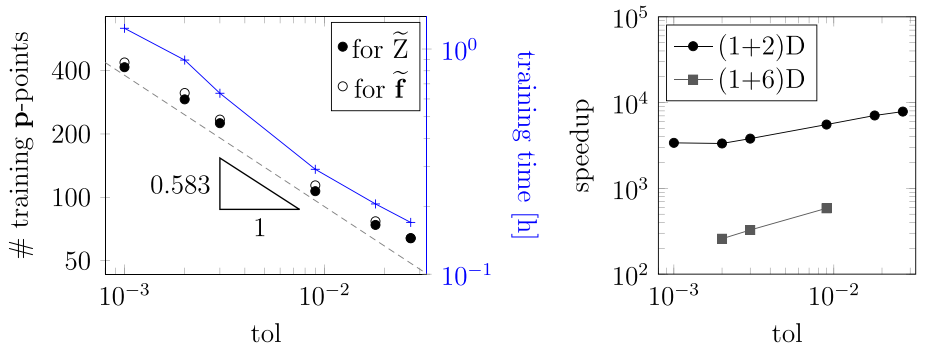

FIGURE 11 Timing results for different values of tol. The left and right plots refer to the offline and online phases, respectively. Left plot: the black dots represent the sizes of the training sets, for $\widetilde{\mathrm{Z}}$ and $\widetilde{\mathbf{f}}$ (full and empty dots, resp.), with the scale on the left axis. The wall-clock time for the training phase is shown as a full blue line, with the scale on the right axis. A trend line is also included, showing the average slope of the training cost. Right plot: online speedup for the (1 + 2)D and (1 + 6)D surrogates.

We assess the accuracy of our approximations by evaluating the surrogates and the exact model on a 30×30 uniform grid of (x_0, p_Λ)-points. The frequency-independence of $\mathbf{f}$ affords us the convenience of plotting the results. In Figure 10, we visualize the exact f_y, the y-component of $\mathbf{f}$, and the evolution of the corresponding surrogate absolute error for selected values of tol. We can observe that the error peaks, in absolute terms, are driven down across the grid as tol is decreased: adding collocation points improves the surrogate approximation in these critical areas of the domain.

Further, we look at the trade-offs between training cost, online evaluation cost, and desired accuracy of a surrogate model for our current application. In Figure 11, we show two different kinds of scaling with respect to tol.

5.2.1 | Offline cost

In Figure 11 (left), we show how the number of explored training **p**-points increases as tol decreases. In the same plot, we also show that the training time is approximately proportional to the size of the training set. This confirms the fact

that computing high-fidelity samples is the bottleneck of the training phase. A trend line is also shown, investigating how the training cost scales with respect to the target accuracy tol. The observed rate ($\sim \text{tol}^{-0.583}$) is particularly interesting if compared to the theoretical approximation rate by hat functions in 1D, namely, $\sim \text{tol}^{-0.5}$, and in 2D over a tensor grid (as opposed to a sparse grid), namely, $\sim \text{tol}^{-1}$. This comparison reminds us of the fact that sparse grids can often recover approximation rates similar to those in just one dimension, being only weakly affected by the "curse of dimension".

5.2.2 | Online cost

In Figure 11 (right), we display the average "online speedup" of the surrogate model. To assess the online speedup, each of the trained surrogates and the high fidelity training model are evaluated at 300 sample points chosen by a Latin hypercube sampling in $\mathbf{p}$-space. Then the "speedup" is defined as the factor (averaged over the 300 sample points) by which evaluating the surrogates $\widetilde{Z}$ and $\widetilde{\mathbf{f}}$ is less expensive than computing these values with the high-fidelity training model. Since Z also depends on ω, for fairness to the high-fidelity models, we compute the impedance speedup by comparing evaluations of $\widetilde{Z}$ and Z at 100 different values of ω, for each fixed $\mathbf{p}$.

We observe that the speedup decreases as we go to smaller values of tol: the surrogate model becomes more complex there, since more collocation points are required. Still, even in the worst case, the speedup is always on the order of 10^3.

As a final note on the online costs, we also compare our surrogate models with the simplified cNGT bearing model referred to in Remark 8, which neglects eccentricity dependence. Despite its simplicity (one of the parameters is completely ignored), the cNGT model evaluation is more than 55 times slower than all our surrogates.

5.3 | 5.4 (1 + 6)D numerical results

In this section, we present results of our most complex surrogate, a complete parametric model of the classical HGJB. Both design and operational parameters are allowed to vary: $\mathbf{p} := (x_0, p_\Lambda, p_\alpha, p_\beta, p_L, p_H)$. The parameter ranges for each of the (1 + 6)-parameters are given in Table 1. We train three surrogates, differing in the tolerance level $\text{tol} \in \{2 \cdot 10^{-3}, 3 \cdot 10^{-3}, 9 \cdot 10^{-3}\}$.

For the purpose of validating the surrogates against the high-fidelity model, we sample the approximation errors over a set of 300 points generated by Latin hypercube sampling. At each of the sample points, the metrics $\bar{e}_{\star,\xi}$ and e_ζ (see Section 3.2) are calculated, in order to quantify the errors between surrogate and high-fidelity values of Z and $\mathbf{f}$. Figure 12 shows the corresponding empirical probability density functions. We observe that, as tol decreases, the error distributions are shifted accordingly.

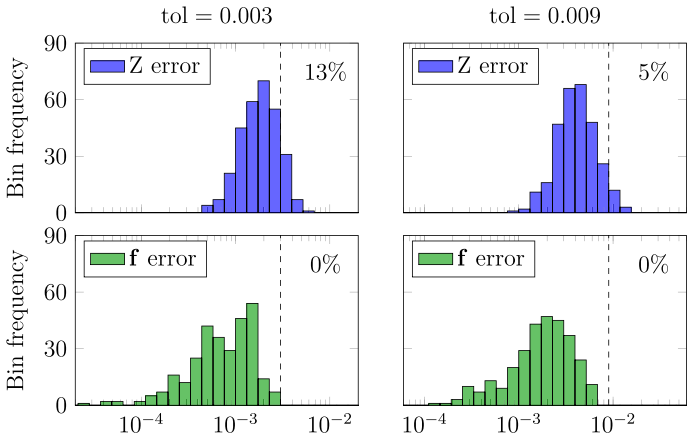

FIGURE 12 Histograms of testing errors, measured at 300 new values of $\mathbf{p}$. Impedance and static forces errors are shown in the top and bottom rows, respectively. Two different values of tol are considered in the two columns. The dashed vertical lines are placed at tol. The proportion of samples that do *not* satisfy the tolerance is indicated in the top right of each plot.

TABLE 2 Summary of computational costs for training (1 + 6)D surrogates.

		Value of tol		
		$2 \cdot 10^{-3}$	$3 \cdot 10^{-3}$	$9 \cdot 10^{-3}$
Building $\widetilde{Z}$	#training **p**-points	69,745	50,043	19,584
	Training time [h]	248	184	68
Post-refinement for $\widetilde{\mathbf{f}}$	#training **p**-points	29,900	22,690	9407
	Training time [h]	102	80	31

For Z, only 13% and 5% of the errors in our random sample on the training domain exceed the respective tol values. Even when the tolerance is not attained, the errors remain approximately within a $(2 \cdot \text{tol})$-level. This behavior is to be expected, since our way of enforcing the tolerance cannot prevent larger-than-desired errors to arise far from the test set used in the predictor-corrector step. On the other hand, **f** behaves better, with all error samples falling below tol.

Further, the tol level qualitatively falls within or just above the upper tail of the distributions. The close proximity of each distribution to its "tol" value speaks to the effectiveness of greedy sampling. Indeed, in the interest of achieving both a quick training phase and a favorable speedup factor, we want the algorithm to use as few training points as necessary. Error distributions much lower than their training tol level would have suggested that fewer collocation points could have been used to get the desired model accuracy.

In Figure 11 (right), we also show the online speedup factors for the (1 + 6)D surrogates. As in the previous example, the speedups are computed based on the 300 **p**-points used in post-processing. We see that, despite four additional parameters, our algorithm still maintains a speedup on the order of 10^2. Even in the worst case, there is still a significant speedup ($> 5\times$) compared to the simplified cNGT model, which neglects the effect of static eccentricity (cf. Remark 8).

As a final point, we revisit training costs. The training times and the required number of training points are reported in Table 2. As in the (1 + 2)D case, our surrogates maintain a proportionality between training time and the number of training points required. The training set size is observed to scale as $\sim \text{tol}^{-0.839}$. This is a further justification for the use of sparse grids as a means to combat the "curse of dimension" in our application. Compared to theoretical 6D tensor grid scaling ($\sim \text{tol}^{-3}$) our observed rate again is more similar to theoretical 1D scaling ($\sim \text{tol}^{-0.5}$).

6 | CONCLUSIONS

In this work, we have proposed a strategy for building surrogate models of QoIs arising from generic parametric frequency-domain problems, as exemplified by the costly nonlinear systems resulting from the study of gas-lubricated bearing dynamics under harmonic perturbation. Non-intrusiveness is paramount in our approach: we do not make any assumptions on the structure of the target high-fidelity problem, so that our method can be applied even if the underlying model is accessible only via "samples" obtained through closed-source code.

Central to our approach and its originality is the combination of rational approximation and smooth interpolation to account for the influences of frequency and other parameters on the underlying dynamical system. Rational approximations are built in a *greedy* manner to efficiently capture the frequency-dependent aspects of the problem. Via locally adaptive sparse grids, we can deal even with a modest number of additional parameters.

Numerical results show, by example, that with our method one can build accurate surrogate models, with significant speedups compared to their reference high-fidelity model. The obtained surrogates have the potential to enable (near-)real-time behavior characterizations of (nonlinear) dynamical systems where they would otherwise be unfeasible, due to the high cost of the underlying model. We look to engineers and practitioners in fields like real-time control and design optimization to leverage potential gains in model efficiency into advances in performance, efficiency, and reliability of real-world applications. Notably, tasks such as anomaly detection and stability analysis (even with feedback from measurements), or optimization of a rich physical model become attainable "online".

The price to pay for a reliable and cheap surrogate model is a lengthy training phase, which however can be performed only once and "offline", that is, before the surrogate is deployed. The training phase can be especially costly when the number of parameters is large, due to the necessity to (non-intrusively) explore high-dimensional parameter domains with

sparse grids. In our numerical tests, adaptivity allowed us to tackle problems depending on up to seven parameters, which would have otherwise been intractable, for example, with tensorized sampling grids. Still, in modest-to-high-dimensional settings, adaptivity can only partially reduce the unavoidable training burden.

It is in this context that we envision the most interesting further research directions, specifically in the study of more effective (adaptive) sampling strategies in high-dimension. Also, since we are sampling with the objective of interpolating, there is also the question of the interpolation basis: in our experiments, we have employed piecewise-linear hat functions, because they are intrinsically hierarchical and naturally suited for approximation over LASGs. Still, their weakness is the lack of accuracy. Alternatives like polynomials, splines, or radial basis functions are often unstable or unwieldy when it comes to high-dimensional *unstructured* interpolation. On the other hand, approximation methods rooted in "machine learning", like Gaussian process regression or neural networks may be more promising. We feel that, in this direction, there is much room for research, both in theory and in numerical practice.

ACKNOWLEDGMENTS

The authors extend their heartfelt thanks to Dimitri Goutaudier for his invaluable feedback on a preliminary version of this work. His contributions significantly improved the quality of this manuscript. Open access funding provided by Ecole Polytechnique Federale de Lausanne.

FUNDING INFORMATION

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

An open-source MATLAB® version of the algorithm is available at https://github.com/pradovera/rational_lasg_matlab.

ORCID

Phillip Huwiler https://orcid.org/0009-0006-3170-6446
Davide Pradovera https://orcid.org/0000-0003-0398-1580

REFERENCES

1. Hesthaven JS, Rozza G, Stamm B. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer Briefs in Mathematics. Springer International Publishing; 2016.
2. Chaturantabut S, Sorensen DC. Nonlinear model reduction via discrete empirical interpolation. *SIAM J Sci Comput*. 2010;32(5):2737-2764. doi:10.1137/090766498
3. Guo M, Hesthaven JS. Reduced order modeling for nonlinear structural analysis using Gaussian process regression. *Comput Methods Appl Mech Eng*. 2018;341:807-826. doi:10.1016/j.cma.2018.07.017
4. Hesthaven JS, Ubbiali S. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J Comput Phys*. 2018;363:55-78. doi:10.1016/j.jcp.2018.02.037
5. Renganathan SA, Maulik R, Rao V. Machine learning for nonintrusive model order reduction of the parametric inviscid transonic flow past an airfoil. *Phys Fluids*. 2020;32(4):047110. doi:10.1063/1.5144661
6. Panzer H, Mohring J, Eid R, Lohmann B. Parametric model order reduction by matrix interpolation. *At-Automatisierungstechnik*. 2010;58(8):475-484. doi:10.1524/auto.2010.0863
7. Baur U, Beattie C, Benner P, Gugercin S. Interpolatory projection methods for parameterized model reduction. *SIAM J Sci Comput*. 2011;33(5):2489-2518. doi:10.1137/090776925
8. Zimmermann R. A locally parametrized reduced-order model for the linear frequency domain approach to time-accurate computational fluid dynamics. *SIAM J Sci Comput*. 2014;36(3):B508-B537. doi:10.1137/130942462
9. Goutaudier D, Nobile F, Schiffmann J. A new method to interpolate POD reduced bases–application to the parametric model order reduction of a gas bearings supported rotor. *Int J Numer Methods Eng*. 2023;124(18):4141-4170. doi:10.1002/nme.7305
10. Walsh JL. *Interpolation and Approximation by Rational Functions in the Complex Domain*. American Mathematical Society; 1960.
11. Saff E, Varga R. *Padé and Rational Approximation: Theory and Applications*. Academic Press Inc; 1977.
12. Hassini MA, Arghir M. A simplified nonlinear transient analysis method for gas bearings. *J Tribol*. 2012;134(1):1. doi:10.1115/1.4005772

13. Nakatsukasa Y, Sète O, Trefethen LN. The AAA algorithm for rational approximation. *SIAM J Sci Comput*. 2018;40(3):A1494-A1522. doi:10.1137/16M1106122
14. Pradovera D. Interpolatory rational model order reduction of parametric problems lacking uniform inf-sup stability. *SIAM J Numer Anal*. 2020;58(4):2265-2293. doi:10.1137/19M1269695
15. Ionita AC, Antoulas AC. Data-driven parametrized model reduction in the Loewner framework. *SIAM J Sci Comput*. 2014;36(3):A984-A1007. doi:10.1137/130914619
16. Schneider F, Papaioannou I, Ehre M, Straub D. Polynomial chaos based rational approximation in linear structural dynamics with parameter uncertainties. *Comput Struct*. 2020;233:106233. doi:10.1016/j.compstruc.2020.106223
17. Brumm J, Scheidegger S. Using adaptive sparse grids to solve high-dimensional dynamic models. *Econometrica*. 2017;85(5):1575-1612. doi:10.3982/ECTA12216
18. Alsayyari F, Perkó Z, Lathouwers D, Kloosterman JL. A nonintrusive reduced order modelling approach using proper orthogonal decomposition and locally adaptive sparse grids. *J Comput Phys*. 2019;399:108912. doi:10.1016/j.jcp.2019.108912
19. Pradovera D, Nobile F. Frequency-domain non-intrusive greedy model order reduction based on minimal rational approximation. *Sci Comput Electr Eng*. 2021;36:159-167. doi:10.1007/978-3-030-84238-3_16
20. Pradovera D, Nobile F. A technique for non-intrusive greedy piecewise-rational model reduction of frequency response problems over wide frequency bands. *J Math Ind*. 2022;12:2. doi:10.1186/s13362-021-00117-4
21. Ma X, Zabaras N. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J Comput Phys*. 2009;228(8):3084-3113. doi:10.1016/j.jcp.2009.01.006
22. Obersteiner M, Bungartz HJ. A generalized spatially adaptive sparse grid combination technique with dimension-wise refinement. *SIAM J Sci Comput*. 2021;43(4):A2381-A2403. doi:10.1137/20M1325885
23. Nobile F, Pradovera D. Non-intrusive double-greedy parametric model reduction by interpolation of frequency-domain rational surrogates. *ESAIM: Math Model Numer Anal*. 2021;55(5):1895-1920. doi:10.1051/m2an/2021040
24. Römer U, Bollhöfer M, Sreekumar H, Blech C, Christine LS. An adaptive sparse grid rational Arnoldi method for uncertainty quantification of dynamical systems in the frequency domain. *Int J Numer Methods Eng*. 2021;122(20):5487-5511. doi:10.1002/nme.6761
25. Oppenheim AV, Willsky AS, Nawab SH. *Signals & Systems*. Prentice-Hall Signal Processing Series. 2nd ed. Prentice Hall; 1997.
26. Worden K, Manson G, Tomlinson GR. A harmonic probing algorithm for the multi-input VOLTERRA series. *J Sound Vib*. 1997;201(1):67-84. doi:10.1006/jsvi.1996.0746
27. Cardona A, Lerusse A, Géradin M. Fast Fourier nonlinear vibration analysis. *Comput Mech*. 1998;22(2):128-142. doi:10.1007/s004660050347
28. Peng ZK, Lang ZQ, Billings SA, Tomlinson GR. Comparisons between harmonic balance and nonlinear output frequency response function in nonlinear system analysis. *J Sound Vib*. 2008;311(1):56-73. doi:10.1016/j.jsv.2007.08.035
29. Nayfeh AH. *Introduction to Perturbation Techniques*. John Wiley & Sons; 2011.
30. Bisplinghoff RL, Ashley H, Halfman RL. *Aeroelasticity*. Courier Corporation; 2013.
31. Raheed A, San O, Kvamsdal T. Digital twin: values, challenges and enablers from a modeling perspective. *IEEE Access*. 2020;8:21980-22012. doi:10.1109/ACCESS.2020.2970143
32. Tang H, Liao YH, Cao JY, Xie H. Fault diagnosis approach based on Volterra models. *Mech Syst Signal Process*. 2010;24(4):1099-1113. doi:10.1016/j.ymssp.2009.09.001
33. Bonizzoni F, Pradovera D, Ruggeri M. Rational-approximation-based model order reduction of Helmholtz frequency response problems with adaptive finite element snapshots. *Math Eng*. 2023;5(4):1-38. doi:10.3934/MINE.2023074
34. Mayo AJ, Antoulas AC. A framework for the solution of the generalized realization problem. *Linear Algebra Appl*. 2007;425:634-662. doi:10.1016/j.laa.2007.03.008
35. Aumann Q, Gosea IV. Practical challenges in data-driven interpolation: dealing with noise, enforcing stability, and computing realizations. 2023. doi:10.48550/arxiv.2301.04906
36. Saliby E, Pacheco F. An empirical evaluation of sampling methods in risk analysis simulation: quasi-Monte Carlo, descriptive sampling, and Latin hypercube sampling. Winter Simulation Conference Proceedings. 2, 1606-1610. 2002.
37. Kucherenko S, Albrecht D, Saltelli A. Exploring multi-dimensional spaces: a Comparison of Latin Hypercube and Quasi Monte Carlo Sampling Techniques. 2015. doi:10.48550/arxiv.1505.02350
38. Gu L, Guenat E, Schiffmann J. A review of grooved dynamic gas bearings. *Appl Mech Rev*. 2020;72(1):010802. doi:10.1115/1.4044191
39. Whipple RTP. *Theory of the Spiral Grooved Thrust Bearing with Liquid or Gas Lubricant*. Great Britain Atomic Energy Research Establishment; 1951.
40. Whipple RTP. *The Inclined Groove Bearing*. United Kingdom Atomic Energy Authority. Research Group. Atomic Energy Research Establishment; 1958.
41. Constantinescu VN, Castelli V. On the local compressibility effect in spiral-groove bearings. *J Lubrication Technol*. 1969;91(1):79-86. doi:10.1115/1.3554902
42. Guenat E, Schiffmann J. Effects of humid air on aerodynamic journal bearings. *Tribol Int*. 2018;127:333-340. doi:10.1016/j.triboint.2018.06.002
43. Lund JW. Calculation of stiffness and damping properties of gas bearings. *J Lubrication Technol*. 1968;90(4):793-803. doi:10.1115/1.3601723
44. Malanoski SB. Experiments on an ultrastable gas journal bearing. *J Lubrication Technol*. 1967;89(4):433-438. doi:10.1115/1.3617021
45. Castelli V, Vohr JH. *Performance Characteristics of Herringbone- Grooved Journal Bearings Operating at High Eccentricity Ratios and with Misalignment*. NASA; 1967.

46. Vohr JH, Chow CY. Characteristics of herringbone-grooved, gas-lubricated journal bearings. *J Basic Eng*. 1965;87(3):568-576. doi:10.1115/1.3650607
47. Phillips GM. *Interpolation and Approximation by Polynomials. 41 of CMS Books in Mathematics*. Springer; 2004.

How to cite this article: Huwiler P, Pradovera D, Schiffmann J. Plug-and-play adaptive surrogate modeling of parametric nonlinear dynamics in frequency domain. *Int J Numer Methods Eng*. 2024;e7487. doi: 10.1002/nme.7487

APPENDIX A. REFERENCE MODEL DETAILS

The nondimensional formulation of the NGT equation given by Equation (14) and the frequency-domain formulation in Section 4.2 can be derived using the scalings

$$\begin{aligned} &P = P^\star/P_a^\star, \quad \omega = \omega^\star/\omega_{rot}^\star, \quad t = t^\star\omega_{rot}^\star, \\ &h_r = h_r^\star/h_{r,nom}^\star, \quad \text{and} \quad h_g = h_g^\star/h_{r,nom}^\star, \end{aligned} \tag{A1}$$

where quantities with a "$\star$" as superscript are dimensional, while quantities without superscript are not. We refer to Figure 3 for context.

For brevity, we introduced coefficients in Equation (14), which can be expanded as

$$\begin{aligned} &a_1 = \frac{b_1 + b_2 \sin^2 p_\beta}{b_3}, \quad a_2 = \frac{b_2 \sin p_\beta \cos p_\beta}{b_3}, \quad a_3 = \frac{b_1 + b_2 \cos^2 p_\beta}{b_3}, \\ &a_4 = \frac{h_g^3 - h_r^3}{b_3}, \quad a_5 = p_\alpha h_g + (1 - p_\alpha) h_r, \quad \text{and} \\ &a_s = -p_\Lambda p_\alpha (1 - p_\alpha)(p_H - 1) \sin p_\beta \end{aligned} \tag{A2}$$

using the quantities

$$b_1 = h_r^3 h_g^3, \quad b_2 = \left(h_g^3 - h_r^3\right)^2 p_\alpha (1 - p_\alpha), \quad \text{and} \quad b_3 = (1 - p_\alpha) h_g^3 + p_\alpha h_r^3. \tag{A3}$$

We note that $a_1, \ldots, a_5$ are space-dependent coefficients and thus they have a perturbation form obtained by injection of Equation (15) in the development of the frequency-domain model in Section 4.2.