



Improving accuracy in parametric reduced-order models for classical guitars through data-driven discrepancy modeling

Pierfrancesco Cillo^{1,*}, Alexander Brauchler¹, Sebastian Gonzalez², Pascal Ziegler¹, Fabio Antonacci², Augusto Sarti², and Peter Eberhard¹

¹ Institute of Engineering and Computational Mechanics, University of Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart, Germany

² Musical Acoustic Lab at the Violin Museum of Cremona, DEIB-Politecnico di Milano, Via Bell'Aspa 3, 26100 Cremona, Italy

Received 8 March 2024, Accepted 22 August 2024

Abstract – Recently developed high-fidelity finite element (FE) models represent a state-of-the-art approach for gaining a deeper understanding of the vibrational behavior of musical instruments. They can also be used as virtual prototypes. However, certain analyses, such as optimization or parameter identification, necessitate numerous model evaluations, resulting in long computation times when utilizing the FE model. Projection-based parametric model order reduction (PMOR) proves to be a powerful tool for enhancing the computational efficiency of FE models while retaining parameter dependencies. Despite their advantages, projection-based methods often require complete system matrices, which may have limited accessibility. Consequently, a systematic discrepancy is introduced in the reduced-order model compared to the original model. This contribution introduces a discrepancy modeling method designed to approximate the parameter-dependent effect of a radiating boundary condition in an FE model of a classical guitar that cannot be exported from the commercial FE software Abaqus. To achieve this, a projection-based reduced-order model is augmented by a data-driven model that captures the error in the approximation of eigenfrequencies and eigenmodes. Artificial neural networks account for the data-driven discrepancy models. This methodology offers significant computational savings and improved accuracy, making it highly suitable for far-reaching parametric studies and iterative processes. The combination of PMOR and neural networks demonstrate greater accuracy than using either approach alone.

This paper extends our prior research presented in the proceedings of Forum Acusticum 2023, offering a more comprehensive examination and additional insights.

Keywords: Guitar, Data-driven model, Virtual prototype, Discrepancy modeling, Model order reduction

1 Introduction

In the last decades, various modeling approaches have been explored for the bodies of string instruments, such as finite difference models [1] and multibody models [2–4]. However, the finite element method (FEM) has predominated and has been used for a large number of studies [5]. The FEM has been used to model the guitar at various stages of its development process, to study the fluid-structure interaction between the body and the enclosed air [6, 7], and to compare different kinds of musical instruments [8–10]. Other works focus on model-based conservation of culturally valuable instruments and the influence of design parameters on the instrument's natural frequencies [11–17].

Despite their broad potential, detailed finite element models are computationally intensive. This makes them

impractical for extensive parametric studies, such as parameter identification or optimization [18], where thousands or more configurations need to be analyzed.

Projection-based parametric model order reduction (PMOR) is a technique used to reduce the computation time of FEM simulations while preserving the parameter dependence of the model. This technique involves projecting the original high-dimensional model onto a lower-dimensional subspace that captures the dominant features of the model [19].

Projection-based methods necessitate access to the complete system matrices, which can pose a limitation when utilizing certain commercial finite element software. Specifically, when employing the commercial finite element software Abaqus [20], it is not feasible, for instance, to export the full system matrices if the finite element model

*Corresponding author: pierfrancesco.cillo@itm.uni-stuttgart.de

incorporates specific features like acoustic structural coupling or radiating boundary conditions (BCs). To address this challenge, the finite element model must undergo slight adaptations before applying model order reduction. The implementation of these adjustments, combined with the PMOR procedure, inevitably results in a systematic discrepancy between the reduced-order model and the original one.

In conjunction with our earlier work presented at Forum Acusticum 2023 [21], of which this is an extension, this contribution presents a novel workflow to obliterate the error between a full-order guitar model and its reduced-order counterpart. This is achieved by modeling the discrepancy between the reduced-order and full-order models. A data-driven approach approximating the parameter-dependent errors in eigenfrequencies and eigenmodes is developed. This involves employing discrepancy modeling, or closure modeling, using deep neural networks as a closure learning framework for reduced-order systems [22–24]. Artificial neural networks (NNs) have the potential to predict certain features of the vibrational behavior of musical instruments [25–27]. To the best of the authors’ knowledge, PMOR and NN methods have never been integrated for the accurate prediction of musical instrument behaviors. This novel combination offers enhanced prediction accuracy over using either approach alone.

The workflow of this contribution involves several key steps to develop an efficient and accurate model. First, a full-order finite element model of the guitar is created to capture its detailed dynamic behavior. From this, a reduced-order model is derived to speedup the computations. Both models are then evaluated for 1000 different parameter configurations to generate a comprehensive dataset. NNs are trained on this dataset to learn the discrepancies between the reduced and full-order models. Finally, the trained NNs are used to correct the reduced-order model, thereby enhancing its accuracy by addressing the identified gaps. While the initial preprocessing step of evaluating the models for 1000 configurations is computationally intensive, it is a one-time effort. Nonetheless, this implies that the method is advantageous only when the number of model evaluations is greater than a given threshold. Therefore, in applications requiring a sufficiently large number of evaluations, the approach becomes more efficient than repeatedly solving the full-order model.

In contrast to [21], this study presents a more detailed description of the order reduction procedure. Further, substantial information has been added concerning the various steps essential for deriving the model. These extended insights are pivotal for a comprehensive understanding of the decision-making process behind the procedure. Additionally, the work has been enriched in terms of findings. Specifically, a comparative analysis has been conducted between the proposed model and a surrogate model employing solely a NN to characterize the guitar’s modal parameters. This extension enables the establishment of the preeminence of our developed model over a purely data-driven model utilizing only NNs.

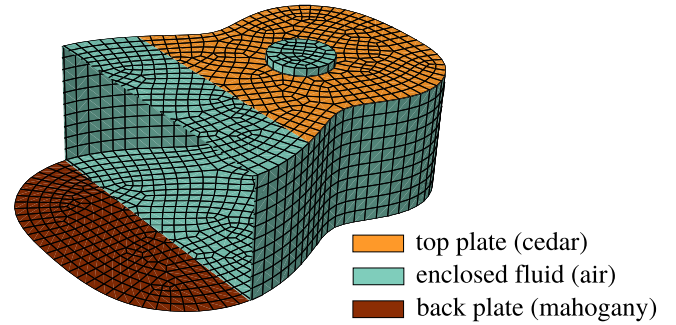


Figure 1. Assembled and meshed finite element model of a classical guitar with different sections for the different parts.

2 Model description

In this section, a finite element model of a classical guitar is presented and a detailed description is provided. This model will be referred to as the full-order model. Projection-based PMOR is then applied to the full-order model, resulting in a reduced-order, or surrogate, model.

2.1 Full-order model

The geometry modeling and finite element discretization are implemented using the commercial software Abaqus. This software allows for the definition of material characteristics, the imposition of acoustic and structural boundary conditions, and the automatic generation of the mesh.

Since it is not the aim of this contribution to study a particular instrument, but to show the feasibility of the method, a simplified model of the guitar is used. Details about the model can be found in [28].

The model consists of three parts: a plane top plate or soundboard with a sound hole, a planar back plate with the same shape as the top plate, and the air inside the cavity, shaped and sized like a classical guitar. The materials for the top and back plates are chosen to be cedar and mahogany, respectively. Figure 1 shows the geometry of the guitar model after the three parts have been meshed.

The fluid-structure interaction between the plates and the air cavity is accounted for by applying a tie constraint between the surface of each plate and the corresponding underlying surface of the air cavity. Using a tie constraint, the pressure degree of freedom (DOF) of each node of the surfaces of the air cavity is approximated by the interpolated values of structural displacements at the nearby plate nodes times the area of the air node [20].

Rather than modeling the actual sides of the guitar, a homogeneous Dirichlet boundary condition is imposed on the edges of both the top and back plates. A homogeneous Neumann boundary condition has been imposed on the sides of the air cavity. According to [7], the air cavity has been modified with a length correction corresponding to the sound hole, as depicted in Figure 1. This modification allows for the consideration of effects due to the external

Table 1. Nominal material parameter values for cedar and mahogany.

	ρ [kg/m ³]	E_L [GPa]	E_R [GPa]	E_T [GPa]	ν_{LR}	ν_{LT}	ν_{RT}	G_{LR} [GPa]	G_{LT} [GPa]	G_{RT} [GPa]
Cedar	320	8.47	0.686	0.466	0.378	0.296	0.403	0.737	0.728	0.042
Mahogany	420	10.7	1.18	0.534	0.297	0.641	0.264	0.939	0.630	0.224

air. Acoustic infinite elements are applied to the surface of the sound hole to simulate sound radiation in that specific location as if the surrounding environment were infinitely large. The model results in $N = 19908$ total DOFs.

The woods of the guitar plates can be considered as orthotropic materials [29], characterizing their behavior with ten material parameters for each plate: density ρ , three Young's moduli E_L , E_T , and E_R , three Poisson ratios ν_{LT} , ν_{LR} , and ν_{TR} , and three shear moduli G_{LT} , G_{LR} , and G_{TR} . Thus, the finite element model of the solid parts is parameterized with 20 material parameters, i.e., 10 for each plate. The subscripts L, T, and R denote the longitudinal, tangential, and radial directions with respect to the wood growth rings. The material parameters are considered homogeneous throughout the plates, following the approach in [30]. The nominal parameter values, taken from [31], are listed in Table 1. The air cavity is characterized by a density $\rho_F = 1.2$ kg/m³ and a bulk modulus $K_F = 142$ kPa.

The final system of equations of motion for the full-order model reads

$$\mathbf{M}_{FE}(\hat{\mathbf{p}})\ddot{\mathbf{q}} + \mathbf{D}_{FE}(\hat{\mathbf{p}})\dot{\mathbf{q}} + \mathbf{K}_{FE}(\hat{\mathbf{p}})\mathbf{q} = \mathbf{f}, \quad (1)$$

where $\mathbf{M}_{FE} \in \mathbb{R}^{N \times N}$, $\mathbf{D}_{FE} \in \mathbb{R}^{N \times N}$ and $\mathbf{K}_{FE} \in \mathbb{R}^{N \times N}$ are the mass, damping, and stiffness matrices of the full-order model. The vector $\mathbf{f} \in \mathbb{R}^N$ represents the external forces acting on the system, while $\mathbf{q} \in \mathbb{R}^N$ contains the displacements of the N nodal DOFs. The parameter dependency of the models is highlighted by the array $\hat{\mathbf{p}} \in \mathbb{R}^{20}$ containing the material parameters of both plates.

It is important to note that Abaqus can only evaluate discrete values of $\hat{\mathbf{p}}$, therefore, an analytical parametric version of the system matrices does not exist. Additionally, when coupling the structure and the fluid, the system matrices are not accessible for export in Abaqus. Approaches to address these limitations will be discussed in the following section.

The eigenfrequencies and eigenmodes of the system, as a function of the material parameters, are computed solving the eigenvalue problem

$$(\mathbf{K}_{FE}(\hat{\mathbf{p}}) - \omega_m^2 \mathbf{M}_{FE}(\hat{\mathbf{p}}))\boldsymbol{\phi}_m = \mathbf{0} \quad (2)$$

with ω_m and $\boldsymbol{\phi}_m$ being the m -th eigenfrequency and eigenmode of the system, respectively.

The use of acoustic infinite elements enables the consideration of stiffness and mass contributions in the extraction of eigenfrequencies and eigenmodes, while damping effects are neglected [20]. Nevertheless, a previous study [30] demonstrates that the employment of acoustic infinite elements yields results consistent with experimental data.

The eigenfrequencies and eigenvectors were computed using the Abaqus embedded eigenvalue solver. To compute

the first 50 eigenfrequencies and eigenmodes with the full-order model, the computational time is approximately 14 s on a workstation equipped with an AMD Ryzen 9 5950X 16-Core Processor and 128 GB of RAM.

2.2 Surrogate model

The full-order model undergoes projection-based model order reduction, resulting in a surrogate model. However, this reduction method relies on the analytical parametric version of the system matrices, which is not available from Abaqus. Additionally, the system matrices at discrete parameter values cannot be exported from the commercial tool if the model incorporates acoustic infinite elements and fluid-structure interaction. Therefore, a few preliminary steps are required, namely:

- (a) Modification of the boundary condition applied to the sound hole;
- (b) Fluid-structure interaction implementation following the export of the structure and fluid matrices individually;
- (c) Analytical formulation of parametric system matrices.

One approach to tackle step a), in order to obtain accessible system matrices, is to replace the acoustic infinite elements at the sound hole with regular acoustic elements, to which a Dirichlet boundary condition is applied, i.e., a pressure of $p = 0$ on the entire sound hole surface. Regarding step b), the matrices of the structural and acoustical parts are exported separately, and the fluid-structure interaction is computed using a numerical solver implemented in Matlab, which operates independently of Abaqus, as described in [28].

Within the scope of step c), it is beneficial to assume a so-called affine parameter dependence (APD) form of the system matrices [32]. It means that the parametric system matrices

$$\begin{aligned} \mathbf{M}(\mathbf{p}) &= \mathbf{M}_0 + \sum_{j=1}^{J_M} a_j(\mathbf{p}) \mathbf{M}_j, \\ \mathbf{K}(\mathbf{p}) &= \mathbf{K}_0 + \sum_{j=1}^{J_K} b_j(\mathbf{p}) \mathbf{K}_j \end{aligned} \quad (3)$$

are expressed as a sum of products of scalar parameter-dependent functions $a_j, b_j \in \mathbb{R}$ and constant matrices $\mathbf{M}_0, \mathbf{M}_j, \mathbf{K}_0, \mathbf{K}_j \in \mathbb{R}^{N \times N}$. In previous research this turned out to be a reasonable assumption [18]. From the fundamental FE equations [33] it can be seen that the mass matrix depends linearly on the density ρ . So, the affinely parameter dependent matrix of the system reads

$$\mathbf{M}_{\text{apd}} = \mathbf{M}_0 + \rho_C \mathbf{M}_1 + \rho_M \mathbf{M}_2, \quad (4)$$

Where ρ_C and ρ_M are the densities of the top and back plate, respectively. Obtaining the APD formulation of the stiffness matrix of an orthotropic material requires a more elaborate procedure. The scalar parameter-dependent functions correspond to the elements of the inverse of the compliance matrix $\mathbf{E}^{-1}$, correlating stress ς and strain κ , see [31]. For the details of the computation, the reader is referred to [34]. This procedure results in $J_M = 2$ parameter-dependent parametric functions and $J_M + 1$ unknown constant matrices $\mathbf{M}_0$, $\mathbf{M}_j$ for the mass, and $J_K = 18$ parameter-dependent parametric functions and $J_K + 1$ unknown constant matrices $\mathbf{K}_0$, $\mathbf{K}_j$ for the stiffness.

The entries of multiple discrete system matrices can be expressed as a linear system of equations to estimate the entries of the parameterized matrices. For the stiffness matrix, this results into the following system of equations

See the Equation (5) bottom of the page

where $k_{FE}^{r,c}(\hat{\mathbf{p}}_i)$ are the matrix entries of row r and column c exported from Abaqus for $I = 1, \dots, J_K + 1$ discrete parameter points. Then, $k_0^{r,c}$ to $k_{J_K}^{r,c}$ are the entries of the unknown matrices $\mathbf{K}_0$, $\mathbf{K}_j$. Therefore, the entries can be computed as

$$\mathbf{k}^{r,c} = \mathbf{A}_{\text{stiff}}^{-1} \mathbf{b}^{r,c}_{\text{stiff}}. \quad (6)$$

The entries $\mathbf{m}^{r,c}$, of the unknown mass matrices $\mathbf{M}_0$, $\mathbf{M}_j$ are computed in the same fashion by solving

$$\mathbf{m}^{r,c} = \mathbf{A}_{\text{mass}}^{-1} \mathbf{b}^{r,c}_{\text{mass}} \quad (7)$$

The parameter-dependent damping matrix is computed using the proportional Rayleigh approach, as

$$\mathbf{D}(\mathbf{p}) = \alpha \mathbf{M}(\mathbf{p}) + \beta \mathbf{K}(\mathbf{p}), \quad (8)$$

where α and β represent the Rayleigh coefficients, which are set here to $\alpha = 0.44$ 1/s and $\beta = 2.4 \cdot 10^{-8}$ s.

After assembling the affinely dependent parametric system matrices, the resulting equations of motion, written as a second-order input-output system, read

$$\begin{aligned} \mathbf{M}_{\text{apd}}(\mathbf{p}) \ddot{\mathbf{q}} + \mathbf{D}_{\text{apd}}(\mathbf{p}) \dot{\mathbf{q}} + \mathbf{K}_{\text{apd}}(\mathbf{p}) \mathbf{q} &= \mathbf{B} \mathbf{u}, \\ \mathbf{y} &= \mathbf{C} \mathbf{q}. \end{aligned} \quad (9)$$

The system inputs $\mathbf{u} \in \mathbb{R}^k$ are distributed on the nodal DOFs via the input matrix $\mathbf{B} \in \mathbb{R}^{N \times k}$. The desired system output points contained in the vector $\mathbf{y} \in \mathbb{R}^j$ are retrieved

via the output matrix $\mathbf{C} \in \mathbb{R}^{j \times N}$. The parameter dependency is represented by the variable vector $\mathbf{p} \in \mathbb{R}^{20}$.

The core idea of model order reduction is, to obtain a reduced vector of DOFs $\mathbf{q}_r \in \mathbb{R}^n$ from which it is possible to retrieve an approximation of the full-order solution $\mathbf{q}$ by back-projecting it using a projection matrix $\mathbf{V} \in \mathbb{R}^{N \times n}$. This means that

$$\mathbf{q} \approx \mathbf{V} \mathbf{q}_r \quad (10)$$

must hold for $n \ll N$. By substituting equation (10) into equation (9), the resulting system is expressed as

$$\mathbf{M}_{\text{apd}}(\mathbf{p}) \mathbf{V} \ddot{\mathbf{q}}_r + \mathbf{D}_{\text{apd}}(\mathbf{p}) \mathbf{V} \dot{\mathbf{q}}_r + \mathbf{K}_{\text{apd}}(\mathbf{p}) \mathbf{V} \mathbf{q}_r = \mathbf{B} \mathbf{u} + \boldsymbol{\epsilon}, \quad (11)$$

where the term $\boldsymbol{\epsilon} \in \mathbb{R}^N$ represents the residual from the approximation. It is possible to get rid of this term by left-multiplying the system of equations with the transpose of another projection matrix $\mathbf{W} \in \mathbb{R}^{N \times n}$, where the rows of $\mathbf{W}^T$ are orthogonal to the residual term $\boldsymbol{\epsilon}$, such that $\mathbf{W}^T \boldsymbol{\epsilon} = \mathbf{0}$. The resulting system reads

$$\begin{aligned} \underbrace{\mathbf{W}^T \mathbf{M}_{\text{apd}}(\mathbf{p}) \mathbf{V}}_{\mathbf{M}_r(\mathbf{p})} \ddot{\mathbf{q}}_r + \underbrace{\mathbf{W}^T \mathbf{D}_{\text{apd}}(\mathbf{p}) \mathbf{V}}_{\mathbf{D}_r(\mathbf{p})} \dot{\mathbf{q}}_r + \underbrace{\mathbf{W}^T \mathbf{K}_{\text{apd}}(\mathbf{p}) \mathbf{V}}_{\mathbf{K}_r(\mathbf{p})} \mathbf{q}_r \\ = \underbrace{\mathbf{W}^T \mathbf{B}}_{\mathbf{B}_r} \mathbf{u} + \underbrace{\mathbf{W}^T}_{\mathbf{0}} \boldsymbol{\epsilon} = \underbrace{\mathbf{C} \mathbf{V}}_{\mathbf{C}_r} \mathbf{q}_r. \end{aligned} \quad (12)$$

The matrices $\mathbf{M}_r \in \mathbb{R}^{n \times n}$, $\mathbf{D}_r \in \mathbb{R}^{n \times n}$, $\mathbf{K}_r \in \mathbb{R}^{n \times n}$, $\mathbf{B}_r \in \mathbb{R}^{n \times k}$ and $\mathbf{C}_r \in \mathbb{R}^{j \times n}$ represent the reduced-order mass, damping, stiffness, input, and output matrices.

The identification of appropriate bases $\mathbf{V}$ and $\mathbf{W}^T$ with $n \ll N$ is needed such that the original system is well approximated. A worthwhile choice is to find appropriate bases with the so-called moment-matching methods. In [35], it is shown that using the projection matrices

See the Equation (13) bottom of the page

computed at discrete parameter expansion point $\mathbf{p}_i$ and discrete frequency expansion point ω_i , the reduced order transfer function is equivalent to the full-order transfer function. The bases in (13) can be concatenated as

$$\begin{aligned} \tilde{\mathbf{V}} &= [\mathbf{V}_1(\omega_1, \mathbf{p}_1), \mathbf{V}_2(\omega_2, \mathbf{p}_2), \dots] \quad \text{and} \\ \tilde{\mathbf{W}} &= [\mathbf{W}_1(\omega_1, \mathbf{p}_1), \mathbf{W}_2(\omega_2, \mathbf{p}_2), \dots] \end{aligned} \quad (14)$$

$$\underbrace{\begin{bmatrix} k_{FE}^{r,c}(\hat{\mathbf{p}}_1) \\ k_{FE}^{r,c}(\hat{\mathbf{p}}_2) \\ \vdots \\ k_{FE}^{r,c}(\hat{\mathbf{p}}_{J_K+1}) \end{bmatrix}}_{\mathbf{b}_{\text{stiff}}^{r,c}} = \underbrace{\begin{bmatrix} 1 & b_1(\hat{\mathbf{p}}_1) & b_2(\hat{\mathbf{p}}_1) & \dots & b_{J_K}(\hat{\mathbf{p}}_1) \\ 1 & b_1(\hat{\mathbf{p}}_2) & b_2(\hat{\mathbf{p}}_2) & \dots & b_{J_K}(\hat{\mathbf{p}}_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & b_1(\hat{\mathbf{p}}_{J_K+1}) & b_2(\hat{\mathbf{p}}_{J_K+1}) & \dots & b_{J_K}(\hat{\mathbf{p}}_{J_K+1}) \end{bmatrix}}_{\mathbf{A}_{\text{stiff}}} \underbrace{\begin{bmatrix} k_0^{r,c} \\ k_1^{r,c} \\ \vdots \\ k_{J_K}^{r,c} \end{bmatrix}}_{\mathbf{k}^{r,c}}, \quad (5)$$

$$\begin{aligned} \text{span}(\mathbf{V}_i) &= \text{span}((-\omega_i^2 \mathbf{M}(\mathbf{p}_i) + i\omega_i \mathbf{D}(\mathbf{p}_i) + \mathbf{K}(\mathbf{p}_i))^{-1} \mathbf{B}(\mathbf{p})) \text{ and} \\ \text{span}(\mathbf{W}_i) &= \text{span}(\mathbf{C}(\mathbf{p})(-\omega_i^2 \mathbf{M}(\mathbf{p}_i) + i\omega_i \mathbf{D}(\mathbf{p}_i) + \mathbf{K}(\mathbf{p}_i))^{-1}), \end{aligned} \quad (13)$$

to form global bases for the projection, in order to guarantee matching transfer functions at all these points. However, this procedure is likely to produce some linear dependent columns, which should be prevented. To ensure that the final projection matrices are solely composed by orthogonal vectors, the truncated singular value decompositions (SVDs) are computed, as

$$\mathbf{U}_V \Sigma_V \mathbf{N}_V = \tilde{\mathbf{V}} \text{ and } \mathbf{U}_W \Sigma_W \mathbf{N}_W = \tilde{\mathbf{W}}. \quad (15)$$

The final bases for the PMOR are computed as the truncated left singular vectors $\mathbf{V} = \mathbf{U}_V(:, 1:n)$ and $\mathbf{W} = \mathbf{U}_W(:, 1:n)$. After the truncation, the conformity between the reduced and full-order transfer function only holds approximately. By selecting a sufficiently large number of expansion points, it is possible to obtain a reduced-order model that ensures a good approximation of the full-order model within the parameter and frequency range of interest.

Only the dependence on six material parameters has been taken into account, while the others have been fixed to their nominal values. The density ρ and the longitudinal Young's modulus E_L have been chosen as they have a high influence on the eigenmodes [18], along with one of lesser influence, namely the longitudinal-tangential shear modulus G_{LT} .

Different combinations of parameter and frequency expansion points have been explored through a random search. The combination offering the best balance between accuracy and dimensionality has been selected. As parameter expansion points, the first $n_{\text{param}} = 50$ samples of a Sobol sequence [36] centered around the nominal values of the free material parameters are used. As frequency expansion points, $n_{\omega} = 25$ points have been chosen equally distributed between 40 Hz and 320 Hz.

Also, the inputs and outputs need to be specified in the matrices $\mathbf{B}$ and $\mathbf{C}$. The same positions are chosen for input and output, namely two on the back plate and three on the top plate. The translation in all the three directions has been considered, resulting in $n_{\text{input}} = 15$.

The concatenated bases obtained from (14) are of the size $\tilde{\mathbf{V}}, \tilde{\mathbf{W}}^T \in \mathbb{R}^{37500 \times 19908}$ where $37500 = 2 n_{\text{param}} n_{\omega} n_{\text{input}}$. Figure 2 depicts the decaying singular values of the projection matrices. The final system order is chosen to be $n = 2471$, since both matrices exhibit a very small decay in singular values for the subsequent values. Therefore, it can be inferred that increasing the system order would not provide significant additional information about the system.

As a result, the first $n = 2471$ left singular vectors compose the final projection matrices $\mathbf{V}, \mathbf{W}^T \in \mathbb{R}^{2471 \times 19908}$. The final reduced-order system has $n = 2471$ DOFs, which is a number significantly smaller than the $N = 19908$ DOFs of the full-order model.

The model order reduction procedure, as well as the preliminary steps a), b) and c), involve additional approximations from the initial full-order model. Consequently, the discrepancy between the final surrogate model and the full-order model will be a result of the accumulation of errors from each approximation step.

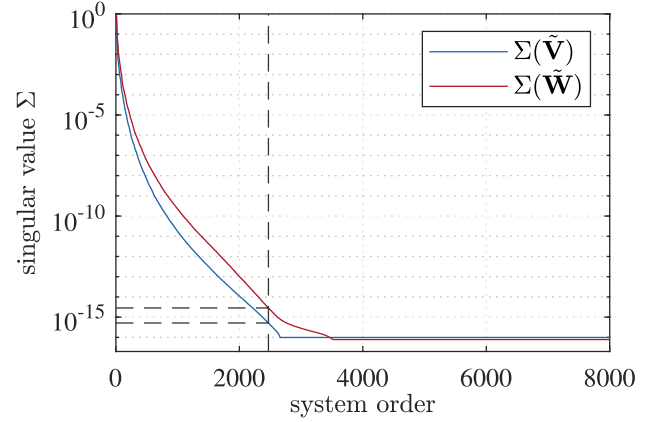


Figure 2. Decaying singular values of the projection matrices $\tilde{\mathbf{V}}$ and $\tilde{\mathbf{W}}$. The dashed lines mark the finally selected system order.

The eigenfrequencies $\omega_{r,m}$ and eigenmodes $\phi_{r,m}$ of the reduced-order model are found solving the eigenvalue problem

$$(\mathbf{K}_r(\mathbf{p}) - \omega_{r,m}^2 \mathbf{M}_r(\mathbf{p})) \phi_{r,m} = \mathbf{0}. \quad (16)$$

On the same workstation previously mentioned, the computational time for computing the first 50 eigenfrequencies and eigenmodes is approximately 1.7 s. This is more than eight times faster than doing the same computation for the full-order model.

3 Discrepancy modeling

In this section, a data-based approach to learn the discrepancy between the full-order and surrogate models in terms of eigenfrequencies and eigenmodes is proposed. Two distinct discrepancy models are developed based on neural networks: one for the eigenfrequencies and one for the eigenmodes.

To train the NNs, a dataset is generated by solving the eigenvalue problems (2) and (16) for 1000 different parameter configurations, computing the first 50 eigenfrequencies and eigenmodes. The configurations are computed using a quasi-random Sobol sequence bounded between $\pm 30\%$ of the nominal values. The hyperparameters of the two networks are tuned independently through a random hyperparameter search, using the mean squared error on the dataset as the performance criterion.

3.1 Eigenfrequencies correction

The parameter-dependent difference between the eigenfrequencies of the two models is written as

$$g_m(\mathbf{p}) = \omega_m(\mathbf{p}) - \omega_{r,m}(\mathbf{p}), \quad (17)$$

and a function $\tilde{g}_m(\mathbf{p})$ is searched, for each considered mode m , that approximates the parameter dependent eigenfrequency error $g_m(\mathbf{p})$. This allows to compute an approximation of the eigenfrequencies of the reduced-order model as

$$\tilde{\omega}_m(\mathbf{p}) = \omega_{r,m}(\mathbf{p}) + \tilde{g}_m(\mathbf{p}) \approx \omega_m(\mathbf{p}). \quad (18)$$

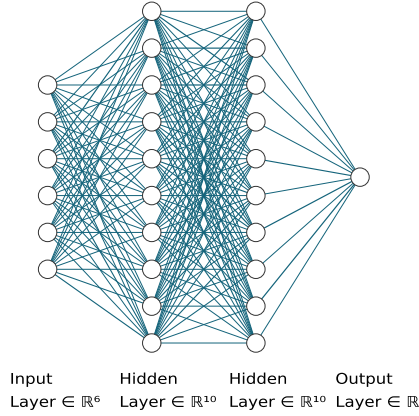


Figure 3. Schematic neural network structure for learning the discrepancy in the eigenfrequencies.

To model $\tilde{g}_m$, a fully-connected multi-layer perceptron is employed, featuring two hidden layers of dimension $\mathbb{R}^{10}$. The input layer contains the parameter vector $\mathbf{p} \in \mathbb{R}^6$, and the output layer contains the approximated eigenfrequency error $\tilde{g}_m \in \mathbb{R}$, as depicted in Figure 3. The activation function of the inner layers of the proposed network is a Rectified Linear Unit function, while the output layer has a linear activation function. The backpropagation during training is implemented using a Broyden-Fletcher-Goldfarb-Shanno quasi-Newton algorithm. One neural network is trained for each mode m over the training set. The training concludes after 1000 iterations to prevent overfitting. Eighty percent of the data is used to train the networks, while the remaining twenty percent is reserved as a test set. After it has been trained, the evaluation of the network takes approximately 30 ms for each mode.

Finally, the performance of the proposed method is compared with the sole use of a neural network to predict the parameter-dependent eigenfrequencies. To achieve this, a neural network with the same architecture as the one previously described is trained with ω_m in the output layer. The eigenfrequencies predicted by the model solely employing the neural network are denoted as $\hat{\omega}_m$.

3.2 Eigenmodes correction

The eigenmodes error term is considered as

$$\Delta_\mu = \bar{\phi}_\mu(\mathbf{p}) - \bar{\phi}_{r,\mu}(\mathbf{p}), \quad (19)$$

where $\bar{\phi}_\mu$ and $\bar{\phi}_{r,\mu}$ represent the μ -th normalized eigenmodes. The parameter-dependent function $\Delta_\mu(\mathbf{p})$ that approximates the discrepancy between the eigenmodes, in order to compute a corrected version of the reduced-order eigenmodes, is

$$\tilde{\phi}_\mu(\mathbf{p}) = \bar{\phi}_{r,\mu}(\mathbf{p}) + \tilde{\Delta}_\mu(\mathbf{p}) \approx \bar{\phi}_\mu(\mathbf{p}). \quad (20)$$

3.2.1 Particle swarm optimization

Training a neural network can be computationally expensive when considering all the structural nodal

displacements of the modeshapes. To handle this, a criterion is needed to select a subset of nodal displacements for comparing the modeshapes of the models.

In order to achieve this, a number D of nodal displacements that adequately approximate the relationship between the modeshapes of the full-order and reduced-order models must be determined. Only the surface node displacements (displacement in the direction normal to the plates of the guitar) are considered, as they are more accessible for possible comparisons with experimental measurements. The modeshapes are compared using the Modal Assurance Criterion (MAC) [37].

The search focuses on finding the minimum error between the diagonals of two different MAC matrices. The first matrix represents the MAC between the modeshapes of the two models computed on all structural nodal displacements, while the second matrix considers only the surface displacement of D nodes. To identify an optimal set of nodes, a Particle Swarm Optimization (PSO) algorithm [38] is employed.

In the PSO algorithm, each particle within the swarm is associated with a position in the search space. The algorithm operates in discrete time steps, updating the positions and velocities of the particles with the objective of minimizing a specified objective function at each iteration. The algorithm is known for its simplicity, efficiency, and ability to explore complex search spaces [39], and it is inspired by the behavior of bird flocking, fish schooling, and swarming theory in general. Each particle in the swarm represents a potential solution to the optimization problem. The particles adjust their positions based on their own experience (personal best) and the collective knowledge of the swarm (global best), allowing the swarm to converge toward optimal solutions through collaborative exploration of the solution space.

In this specific scenario, the search space corresponds to the array containing all the surface nodes. The positions of the particles correspond to D indices of the aforementioned array. The position and velocity of the particles are updated at each iteration with the aim of minimizing a specified objective function $J(\mathbf{x})$. This objective function is computed for each time step, where $\mathbf{x}$ is the array containing the position of all the particles. The objective function if we choose the first 50 modeshapes is proposed as

$$J(\mathbf{x}) = \sqrt{\sum_{\mu=1}^{50} \left| \frac{\text{MAC}_{\mu\mu} - \text{MAC}(\mathbf{x})_{\mu\mu}}{\text{MAC}_{\mu\mu}} \right|^2}, \quad (21)$$

where $\text{MAC}_{\mu\mu}$ represents the diagonal elements of the MAC matrix considering all the structural DOFs, and $\text{MAC}(\mathbf{x})_{\mu\mu}$ represents the MAC matrix considering the D surface node displacements at position $\mathbf{x}$.

The computation involves analyzing the modeshapes of the full-order and surrogate model with the material parameters fixed at their nominal values. This process is conducted with varying numbers of nodes in a range between 100 and 200. The solution with the lowest value of the objective function is selected, namely $D = 160$. Although

the chosen configuration is adequate for approximating the difference between the modeshapes of the models, it cannot be ruled out that this particular solution of the PSO is a local minimum rather than a global one.

3.2.2 Dictionary of modeshapes

Eigenmodes exhibit continuous variations within the parameter space, and moreover, they appear and disappear in certain parameter ranges. Therefore, it is crucial to use a method to classify the modeshapes, exploring the regions within the parameter space where they occur. To achieve this, a so-called *dictionary* of modeshapes is established. This dictionary encompasses all modeshapes within the dataset, which are compared using the MAC. Similar modeshapes are grouped together in clusters within the dictionary. The dictionary is initially populated with modeshapes corresponding to parameter values fixed at their nominal values. Subsequently, each modeshape in the dataset is compared to the items in the dictionary. If the MAC is greater than 0.8, indicating a high similarity, the modeshape and its corresponding parameter configuration are stored. Otherwise, a new record is created, and the modeshape is added to this new item. Each record forms a category to which a modeshape can belong. Using this methodology, a regression model is learned for each dictionary item, rather than for each mode number, as done for the eigenfrequencies. Therefore, the symbol μ in equations (19) and (20) does not denote the mode number, as for the eigenfrequencies, but it represents the dictionary item number.

3.2.3 Network architecture

A feedforward NN has been employed as the learning method. The input layer of the network has a dimension of $\mathbb{R}^6$ and contains the parameter array $\mathbf{p}$. The network comprises four hidden layers. The first two have dimension $\mathbb{R}^{20}$, while the last two have dimension $\mathbb{R}^{160}$. A schematic of the NN architecture is illustrated in Figure 4. The output layer contains the approximated error for one dictionary item, with a dimension of $\mathbb{R}^{160}$. In the first three hidden layers, each neuron is characterized by a hyperbolic tangent transfer function. The neurons in the last hidden layer use a linear transfer function.

The first three hidden layers of the NN are fully connected. Notably, between the two large hidden layers of dimension $\mathbb{R}^{160}$, the weights are not learned but are predefined in a neighborhood relation that can be expressed in matrix form with the entries

$$w_{ji}(\mathbf{x}_j, \mathbf{x}_i) = \begin{cases} 0.9 \left(1 - \frac{\|\mathbf{x}_j - \mathbf{x}_i\|_2}{\delta}\right) + 0.1, & \text{if } \|\mathbf{x}_j - \mathbf{x}_i\|_2 \leq \delta, \\ 0, & \text{if } \|\mathbf{x}_j - \mathbf{x}_i\|_2 > \delta, \end{cases} \quad (22)$$

Where $\|\mathbf{x}_j - \mathbf{x}_i\|_2$ is the Euclidean distance between coordinates of two nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ of the mesh, and $\delta = 5$ cm represents the threshold distance. Nodes within the threshold distance from each other receive weights that increase linearly from 0.1 to 1 as the Euclidean distance between

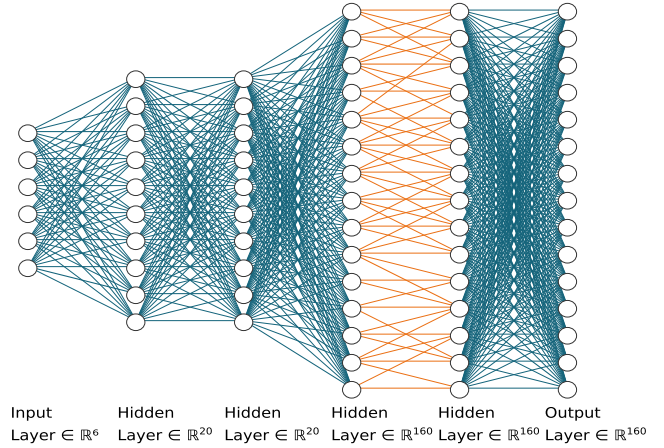


Figure 4. Schematic neural network structure for learning the discrepancy in the eigenmodes. Connections in orange highlight predefined weights.

them decreases. Nodes outside the threshold distance will have weight $w_{ji}(\mathbf{x}_j, \mathbf{x}_i) = 0$. The use of predefined weights grants a significant reduction in training time. To validate the usage of predefined weights inside the neural network, its performance is compared to that of a fully connected network with the same architecture.

One NN for each dictionary item is trained using the Levenberg-Marquardt backpropagation algorithm. In contrast to the NN employed for the eigenfrequencies, this algorithm has been chosen because of the larger NN size, due to its fast convergence feature [40].

The layer weights are initialized randomly, and the training finishes after 100 iterations. The difference in the number of iterations, compared to the NN used for eigenfrequencies, is due to the different size of the networks. Each training iteration for the eigenmode NN can take up to 900 times longer than for the eigenfrequency NN. A total of 100 iterations was found to provide the best trade-off between approximation accuracy and training time. All the training runs converged and their error plateaued. Eighty percent of the data is used for training the networks, while the remaining twenty percent is used as test set. After training, evaluating the NN takes 0.1 s for each dictionary eigenmode.

4 Results

The preprocessing steps, which involve the computation of the projection matrices $\mathbf{V}$ and $\mathbf{W}^T$ the evaluation of the full-order and the surrogate models for 1000 parameter configurations to build the dataset, and the training of the NNs, take about 15 h. Hence, despite the reduced online effort, the usage of the model combining PMOR and NN is advantageous over the full-order model only when performing more than 5600 evaluations.

On the test set, the eigenfrequencies ω^* are computed for three different study cases: $\omega^* = \omega_{r,m}$ in case no correction is applied to the surrogate model, $\omega^* = \hat{\omega}_m$ in case the

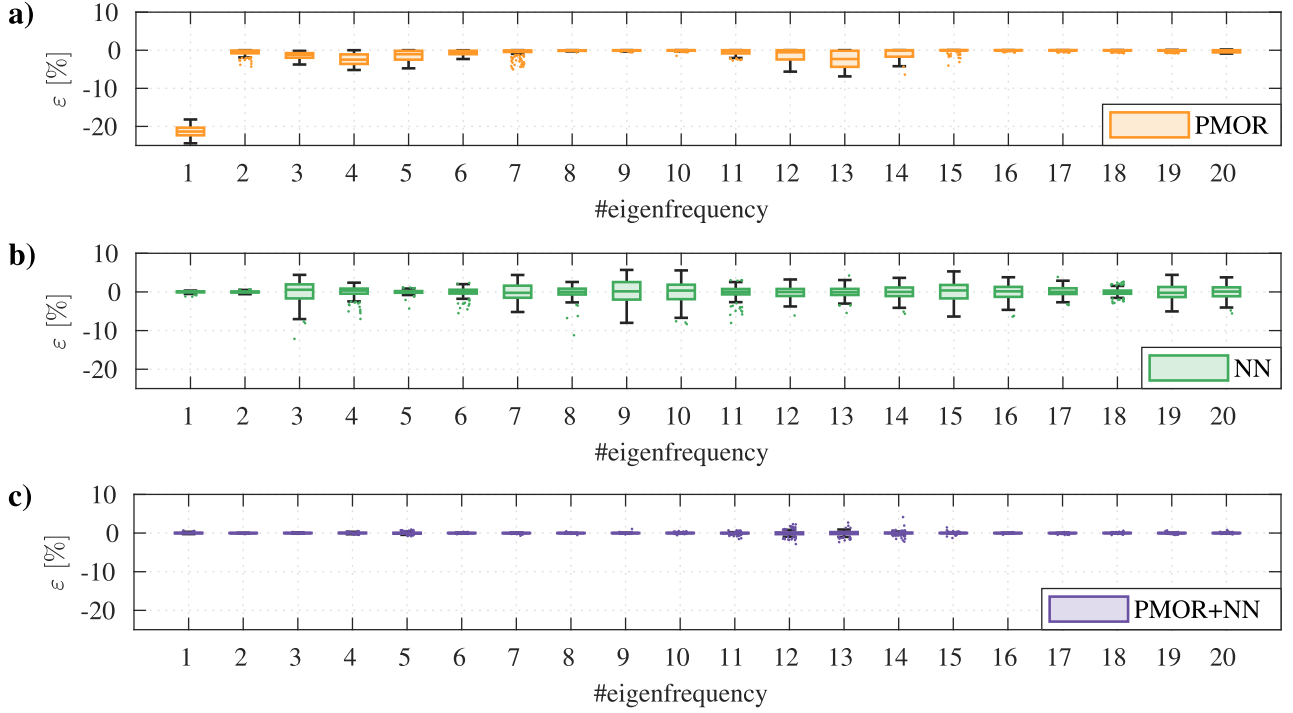


Figure 5. Box plots representing the relative eigenfrequency errors for the first 20 modes using the PMOR model (a), the NN model (b), and the PMOR model combined with the NN discrepancy model (c).

model employing the sole NN is used, and $\omega^* = \tilde{\omega}_m$ computed using equation (18) if the eigenfrequencies of the surrogate model are corrected via discrepancy modeling. For all cases, the relative eigenfrequency error ε_m is calculated as

$$\varepsilon_m = \frac{\omega_m - \omega^*}{\omega_m}. \quad (23)$$

The results are visualized using box plots, where each box contains the data between the 25th and 75th percentiles, with a central mark denoting the median. Whiskers extending from the box represent a distance of 1.5 times the interquartile range. Outliers, data points lying beyond this range, are depicted as scattered points.

In Figure 5, the box plots illustrate the relative eigenfrequency error for the first 20 modes across all study cases. The eigenfrequencies of the PMOR model have a systematically lower value than the ones of the full-order model. The most significant errors in the PMOR model occur in eigenfrequencies 1 to 7 and 11 to 14, with the first eigenfrequency having the highest relative error, averaging -21.35% .

The method combining PMOR and NN shows a marked improvement over using PMOR alone. This combined approach effectively adjusts all eigenfrequencies, shifting the median value toward zero and reducing the variance. After applying the discrepancy model correction, the relative error for the eigenfrequencies is falls within $-1\% < \varepsilon < 1\%$, with an average value of 0.11% . Given that the average relative error before correction is 1.73% , discrepancy modeling reduces the average error by 94% . Even the discrepancy of modes with relatively low absolute

relative error is further reduced. A comparison between the combined method and the method employing only the NN reveals a clear advantage of the former. The sole NN method produces error distributions with median values generally farther from zero, averaging 1.21% , and systematically higher variance values. In contrast, the combined PMOR and NN method produces results that are over 10 times more accurate.

On the test set, the modeshapes are computed using both the PMOR model and the combined PMOR and NN model. Their performances are evaluated by calculating their correlation with the modeshapes of the full-order model. This correlation is computed as the normalized scalar product between two modeshapes.

The evaluation of modeshape discrepancy modeling performance is not straightforward. Therefore, Figure 6 provides a visual representation of a modeshape correction example, specifically for the 9th dictionary eigenmode of a particular parameter configuration $\hat{\mathbf{p}}$. In this instance, the correction is evident as the possibly imprecise modeshape from the reduced-order model is adjusted through the discrepancy model, resulting in a final modeshape much closer to the modeshape of the full-order model. The correlation increases from 0.94 before correction to 0.99 after correction.

Figure 7 shows the distribution of the correlations between the full-order and reduced-order model, both before and after the discrepancy model correction, for the first 20 dictionary modes computed on the test set. The data are presented as box plots. It is important to note that the eigenmodes in this figure do not directly correspond to

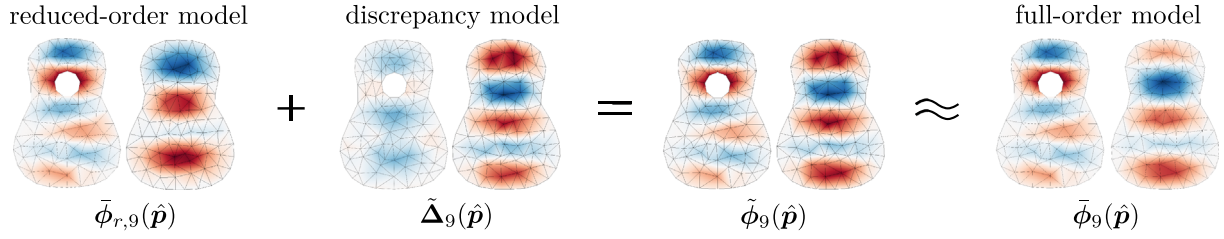


Figure 6. Visual example of modeshape correction for the 9th dictionary eigenmode of a specific parameter configuration $\hat{p}$.

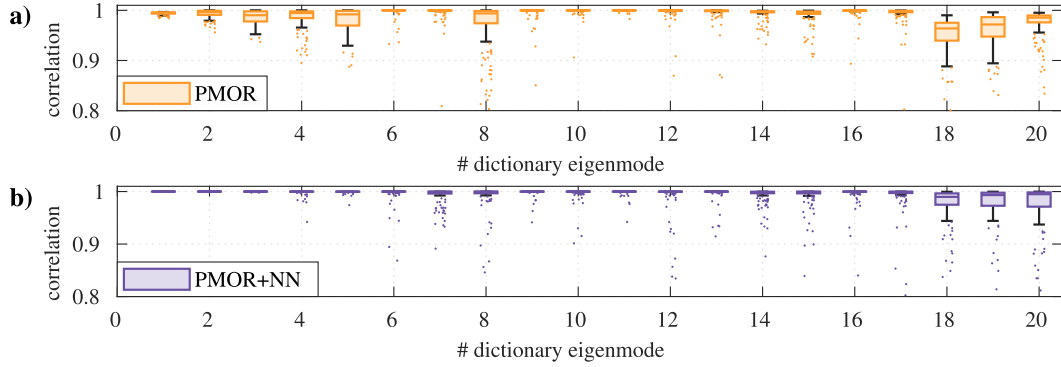


Figure 7. Box plots representing the correlation values between full-order and surrogate model for the first 20 dictionary modes using the PMOR model (a), and the PMOR model combined with the NN discrepancy model (b).

the eigenfrequencies in Figure 5, as the ordering here is determined by the dictionary of modeshapes. For dictionary items where the modeshapes are already well approximated with an average correlation greater than 0.99, the discrepancy model correction does not result in significant improvement. In fact, for dictionary items 6, 7, 9 to 11, and 14, the corrected model performs slightly worse, showing a higher number of outliers. However, for other dictionary items, where the distribution of the correlation values was not concentrated around one, the correction leads to overall improvement, shifting the median values closer to 1 and reducing variances. The only exception is dictionary item 20, where the median value is closer to 1 after correction, but the variance is nearly doubled.

The performance of the NN with predefined weights are compared with the performance of a fully-connected NN with the same architecture. Specifically, the distributions of the correlation values of the modeshapes obtained using the two different networks for the discrepancy modeling are compared. The distributions, with respect to the two methods, are situated within a highly comparable range. The difference is minimal and not appreciable. In spite of this, on the aforementioned workstation a NN with predefined weights grants a speedup in training of 100 with respect to the fully connected NN.

5 Conclusions

The developed data-driven method for enhancing a reduced-order model of a classical guitar by modeling the discrepancy between full-order and reduced-order models

shows promising results. By integrating projection-based parametric model order reduction with neural network discrepancy modeling, our methodology addresses the inherent discrepancies between full-order and reduced-order models, particularly in predicting eigenfrequencies and eigenmodes, by leveraging data-driven techniques. The significant reduction in error for the first 50 eigenfrequencies, particularly for crucial lower-order modes in characterizing the guitar's sound, makes this approach valuable [41]. The method provides a substantial improvement over the surrogate model without any drawbacks.

A crucial aspect of the methodology is the establishment of a dictionary of modeshapes, which facilitates the classification and comparison of modeshapes within the parameter space. By categorizing modeshapes based on their similarity, discrepancy models can be developed for each category, allowing for more precise and targeted corrections. However, the effectiveness of the method varies among different modeshapes. It is crucial to exercise great care in its application, focusing on specific modeshapes where it proves beneficial. Despite its efficacy, there is room for further refinement, potentially through an improved assembly procedure of the dictionary of modeshapes that could leverage a classification learning model for assembling clusters.

In terms of neural network architecture, an approach is employed that incorporates predefined weights between nodes based on spatial proximity. This solution significantly reduces training time while maintaining comparable performance to fully connected networks, enabling efficient and accurate discrepancy modeling without compromising computational resources.

The results demonstrate that the integration of PMOR with neural network discrepancy modeling offers several advantages over traditional PMOR or data-driven approaches alone. Specifically, the approach achieves improved predictive accuracy by systematically correcting errors in eigenfrequencies and eigenmodes, resulting in more reliable structural analysis results. Moreover, the discrepancy modeling approach effectively reduces variance in both eigenfrequency predictions and modeshape comparisons, resulting in more consistent and reliable predictions across different parameter configurations.

Due to the computationally intensive preprocessing step, the approach combining PMOR and NN is beneficial over the full-order model only when the number of evaluations exceeds a certain threshold, which varies depending on the model. While not universally applicable, the proposed framework is well-suited for applications requiring numerous model evaluations, such as optimization or probabilistic uncertainty quantification [18].

In this study, the discrepancy modeling was applied to parameter-dependent eigenfrequencies and eigenmodes of full-order and surrogate models. Another direction for exploration could involve applying the discrepancy modeling method directly to the mass and stiffness matrices of the model. This would involve learning the missing terms in the affine parametric matrices, resulting in an additional parameter-dependent matrix term in the summation, capturing the discrepancy between the two models.

Future research could extend the application of this method to a fully-detailed finite element guitar model, such as those developed in [30, 42–44]. This extension could lead to better-approximated efficient models, enhancing our understanding of existing instruments.

Acknowledgments

This research was partially supported by the German Research Foundation DFG (Project No. 455440338). The authors are grateful for the support. The authors express their gratitude to Ingeborg Wenger for engaging in valuable discussions and providing thoughtful remarks, particularly concerning the application of artificial neural networks.

Conflicts of interest

The authors declare no conflicts of interest.

Data availability statement

The datasets are available from the corresponding author on reasonable request.

Author contribution statement

P.C.: Methodology, software, data curation, visualization, investigation, writing-original draft; A.B.: Conceptualization, methodology, software, writing-reviewing and editing, supervision; S.G.: Methodology, writing-reviewing and editing, supervision; P.Z.: Supervision, funding acquisition, writing-reviewing and editing; F.A.: Supervision, funding acquisition, writing-reviewing and editing; A.S.: Supervision, funding acquisition, writing-reviewing

and editing; P.E.: Supervision, project administration, funding acquisition, writing-reviewing and editing.

References

1. R. Bader: Computational mechanics of the classical guitar, Springer Berlin, Heidelberg, Germany, 2006.
2. O. Christensen: Quantitative models for low frequency guitar function. *Journal of Guitar Acoustics* 6 (1982) 10–25.
3. J.E. Popp: Four mass coupled oscillator guitar model. *The Journal of the Acoustical Society of America* 131, 1 (2012) 829–836.
4. R. Mores: Sound tuning in asymmetrically braced guitars. *The Journal of the Acoustical Society of America* 149, 2 (2021) 1041–1057.
5. E. Kaselouris, M. Bakarezos, M. Tatarakis, N.A. Papadogiannis, V. Dimitriou: A review of finite element studies in string musical instruments. *Acoustics* 4, 1 (2022) 183–202.
6. M.J. Elejabarrieta, A. Ezcurra, C. Santamaria: Coupled modes of the resonance box of the guitar. *The Journal of the Acoustical Society of America* 111, 5 (2002) 2283–2292.
7. A. Ezcurra, M. Elejabarrieta, C. Santamaria: Fluid-structure coupling in the guitar box: numerical and experimental comparative study. *Applied Acoustics* 66, 4 (2005) 411–425.
8. J.A. Torres, R.R. Boullosa: Influence of the bridge on the vibrations of the top plate of a classical guitar. *Applied Acoustics* 70, 11–12 (2009) 1371–1377.
9. D. Salvi, S. Gonzalez, F. Antonacci, A. Sarti: Modal analysis of free archtop guitar top plates. *The Journal of the Acoustical Society of America* 150, 2 (2021) 1505–1513.
10. A. Brauchler, S. Gonzalez, M. Vierneisel, P. Ziegler, F. Antonacci, A. Sarti, P. Eberhard: Model-predicted geometry variations to compensate material variability in the design of classical guitars. *Scientific Reports* 13, 1 (2023) 12766.
11. R. Viala, V. Placet, S. Le Conte, S. Vaiedelich, S. Cogan: Model-based decision support methods applied to the conservation of musical instruments: application to an antique cello. In: *Model validation and uncertainty quantification, Volume 3: Proceedings of the 37th IMAC, A Conference and Exposition on Structural Dynamics 2019*, pp. 223–227, Springer, 2020.
12. R. Viala, V. Placet, S. Cogan, E. Foltête: Model-based effects screening of stringed instruments. In: *Model validation and uncertainty quantification, Volume 3: Proceedings of the 34th IMAC, A Conference and Exposition on Structural Dynamics 2016*, pp. 151–157, Springer, 2016.
13. V. Chatziioannou: Reconstruction of an early viola da gamba informed by physical modeling. *The Journal of the Acoustical Society of America* 145, 6 (2019) 3435–3442.
14. S. Gonzalez, D. Salvi, F. Antonacci, A. Sarti: Eigenfrequency optimisation of free violin plates. *The Journal of the Acoustical Society of America* 149, 3 (2021) 1400–1410.
15. P. Dumond, N. Baddour: Effects of using scalloped shape braces on the natural frequencies of a brace-soundboard system. *Applied Acoustics* 73, 11 (2012) 1168–1173.
16. R. Viala, V. Placet, S. Cogan: Identification of the anisotropic elastic and damping properties of complex shape composite parts using an inverse method based on finite element model updating and 3d velocity fields measurements (femu-3dvf): application to bio-based composite violin soundboards. *Composites Part A: Applied Science and Manufacturing* 106 (2018) 91–103.
17. H. Tahvanainen, J. Pölkki, H. Penttinen, V. Välimäki: Finite element model of a kantele with improved sound radiation. In: *Proceedings of the Stockholm Music Acoustic Conference, July 30–August 3, Stockholm, Sweden, 2013*, pp. 193–198.

18. A. Brauchler, D. Hose, P. Ziegler, M. Hanss, P. Eberhard: Distinguishing geometrically identical instruments: possibilistic identification of material parameters in a parametrically model order reduced finite element model of a classical guitar. *Journal of Sound and Vibration* 535 (2022) 117071.
19. A.C. Antoulas: Approximation of large-scale dynamical systems, Society for Industrial and Applied Mathematics, Philadelphia, USA, 2005.
20. Abaqus, Analysis User's Guide, Version 6.14. Simulia, 2014.
21. P. Cillo, A. Brauchler, S. Gonzalez, P. Ziegler, F. Antonacci, A. Sarti, P. Eberhard: A data-based method enhancing a parametrically model order reduced finite element model of a classical guitar. In: *Proceedings of Forum Acusticum 2023*, September 11–15, Turin, Italy, 2023.
22. X. Xie, C. Webster, T. Iliescu: Closure learning for nonlinear model reduction using deep residual neural network. *Fluids* 5, 1 (2020) 39.
23. K. Kaheman, E. Kaiser, B. Strom, J.N. Kutz, S.L. Brunton: Learning discrepancy models from experimental data. Preprint: arXiv, 2019. <https://doi.org/10.48550/arXiv.1909.08574>.
24. P.D. Arendt, D.W. Apley, W. Chen: Quantification of model uncertainty: calibration, model discrepancy, and identifiability. *Journal of Mechanical Design* 134, 10 (2012) 100908.
25. S. Gonzalez, D. Salvi, D. Baeza, F. Antonacci, A. Sarti: A data-driven approach to violin making. *Scientific Reports* 11, 1 (2021) 9455.
26. D. Salvi, S. Gonzalez, F. Antonacci, A. Sarti: Parametric optimization of violin top plates using machine learning. In: *27th International Congress on Sound and Vibration ICSV 2021*, July 11–16, 2021.
27. D.G. Badiane, S. Gonzalez, R. Malvermi, F. Antonacci, A. Sarti: A neural network-based method for spruce tonewood characterization. *The Journal of the Acoustical Society of America* 154, 2 (2023) 730–738.
28. J. Rettberg, D. Wittwar, P. Buchfink, A. Brauchler, P. Ziegler, J. Fehr, B. Haasdonk: Port-Hamiltonian fluid-structure interaction modeling and structure-preserving model order reduction of a classical guitar. *Mathematical and Computer Modelling of Dynamical Systems* 29, 1 (2023) 116–148.
29. D. Konopka, C. Gebhardt, M. Kaliske: Numerical modelling of wooden structures. *Journal of Cultural Heritage* 27 (2017) S93–S102.
30. A. Brauchler, P. Ziegler, P. Eberhard: An entirely reverse-engineered finite element model of a classical guitar in comparison with experimental data. *The Journal of the Acoustical Society of America* 149, 6 (2021) 4450–4462.
31. D.E. Kretschmann: Mechanical properties of wood, in: *Wood handbook: wood as an engineering material*, Forest Products Laboratory, Madison, USA, 2010.
32. P. Benner, S. Gugercin, K. Willcox: A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review* 57, 4 (2015) 483–531.
33. O.C. Zienkiewicz, R.L. Taylor: *The finite element method for solid and structural mechanics*, Butterworth-Heinemann, Elsevier, Oxford, UK, 2005.
34. A. Brauchler: Predictive computational models of classical guitars: modeling, order-reduction, simulation and experimentation, vol. 2023, 78 of *Schriften aus dem Institut für Technische und Numerische Mechanik der Universität Stuttgart*. Shaker Verlag, 2023.
35. U. Baur, C. Beattie, P. Benner, S. Gugercin: Interpolatory projection methods for parameterized model reduction. *SIAM Journal on Scientific Computing* 33, 5 (2011) 2489–2518.
36. I.M. Sobol': On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* 7, 4 (1967) 784–802.
37. M. Pastor, M. Binda, T. Harčarik: Modal assurance criterion. *Procedia Engineering* 48 (2012) 543–548.
38. J. Kennedy, R. Eberhart: Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks*, Vol. 4, IEEE, 1995, pp. 1942–1948.
39. T.M. Shami, A.A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M.A. Summakieh, S. Mirjalili: Particle swarm optimization: a comprehensive survey. *IEEE Access* 10 (2022) 10031–10061.
40. M.T. Hagan, M.B. Menhaj: Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks* 5, 6 (1994) 989–993.
41. J. Meyer: Quality aspects of the guitar tone. In: *Function, construction and quality of the guitar*, E.V. Jansson (Ed.), Royal Swedish Academy of Music, Stockholm, 1983, pp. 51–75.
42. H. Tahvanainen, H. Matsuda, R. Shinoda: Numerical simulation of the acoustic guitar in virtual prototyping. *Proceedings of ISMA 2019* (2019) 13–17.
43. M. Lercari, S. Gonzalez, C. Espinoza, G. Longo, F. Antonacci, A. Sarti: Using mechanical metamaterials in guitar top plates: a numerical study. *Applied Sciences* 12, 17 (2022) 8619.
44. G. Longo, S. Gonzalez, F. Antonacci, A. Sarti: Predicting the acoustics of archtop guitars using an ai-based algorithm trained on fem simulations. In: *Proceedings of Forum Acusticum 2023*, September 11–15, Turin, Italy, 2023.

Cite this article as: Cillo P. Brauchler A. Gonzalez S. Ziegler P. Antonacci F, et al. 2024. Improving accuracy in parametric reduced-order models for classical guitars through data-driven discrepancy modeling. *Acta Acustica*, 8, 59. <https://doi.org/10.1051/aacus/2024055>.