Simulation-Based Anomaly Detection and Damage Localization: an Application to Structural Health Monitoring

Caterina Bigoni[1,*], Jan S. Hesthaven[1]

Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract

We propose a simulation-based decision strategy for the proactive maintenance of complex structures with a particular application to structural health monitoring (SHM). The strategy is based on a data-driven approach which exploits an offline-online decomposition. A synthetic dataset is constructed offline by solving a parametric time-dependent partial differential equation for multiple input parameters, sampled from their probability distributions of natural variation. The collected time-signals, extracted at sensor locations, are used to train classifiers at such sensor locations, thus constructing multiple databases of healthy configurations. These datasets are then used to train one class Support Vector Machines (OC-SVMs) to detect anomalies. During the online stage, a new measurement, possibly obtained from a damaged configuration, is evaluated using the classifiers. Information on damage is provided in a hierarchical manner: first, using a binary feedback, the entire structure response is either classified as inlier (healthy) or outlier (damaged). Then, for the outliers, we exploit the outputs of multiple classifiers to retrieve information both on the severity and the spatial location of the damages. Because of the large number of signals needed to construct the datasets offline, a model order reduction strategy is implemented to reduce the computational burden. We apply this strategy to both 2D and 3D problems to mimic the vibrational behavior of complex structures under the effect of an active source and show the effectiveness of the approach for detecting and localizing cracks.

Keywords: Structural Health Monitoring, Digital Twin, Crack Detection, Reduced Order Modeling, Anomaly Detection, One-Class Classification

1. Overview

Structural Health Monitoring (SHM) refers to automated monitoring procedures that aim at assessing the state of damage of aerospace, civil or mechanical structures [16]. An early detection of faults, e.g., cracks or corrosion, has the potential to greatly reduce the maintenance cost over the life time of a structure and may help prevent catastrophic events. The combined advent of low-cost sensor technologies and digital twins, i.e., accurate virtual representations of complex heavy industry assets, have helped in the transition from classical time-based maintenance with scheduled periodic inspections to condition-based maintenance for large-scale structural systems. The combination of parametrized mathematical models with experimental data is crucial to guarantee reliable monitoring of the lifecycle phases of a structure. We focus here on applications where the physical system can be modeled by parametric partial differential equations (pPDEs), e.g., offshore wind turbines and concrete oil-rigs, or smaller components such as wind turbine blades or composite pipes.

We present a general *data-driven methodology* that, by combining physics-based models with experimental observations, allows us to make predictions on the state of damage of a structure of interest [16]. Mathematical numerical models are exploited to approximate the propagation of waves in the structure under the effect of an active source. However, a continuous source, used to mimic the effect of tides or wind, could also be considered. The goal is to compare the measurements of a network of sensors, placed on the structure,

*Corresponding author

Email addresses: caterina.bigoni@epfl.ch (Caterina Bigoni), jan.hesthaven@epfl.ch (Jan S. Hesthaven)

[1]Computational Mathematics and Simulation Science (MCSS), Institute of Mathematics (MATH), EPFL

with equivalent quantities of interests obtained from numerical simulations. By introducing suitable approximations, we recreate the geometry of the structure with its material properties and boundary conditions to emulate the time-signals recovered from sensors, e.g., local displacements, accelerations or strains in a specified time frame. Then, under the assumption that the received signals from a healthy or damaged structure will encode substantial differences, we aim to identify structural flaws. We rely on reduced order modeling techniques to accelerate the process of constructing the database and machine learning techniques to build a classifier.

This process fully exploits an *offline-online decomposition* of tasks. The offline phase consists in building a synthetic database of time-signals which represent the behavior of the structure of interest under normal operational conditions and healthy variations. These time-signals are an approximation of the real time-signals collected from the sensors placed on heathy structure. During the online phase, real experimental time-signals, either collected from sensors placed on a damaged or a healthy structure, are compared with those simulated offline using the classifier. This approach differs from a model-based methodology, where the goal is to estimate the parameters that minimize the difference between the model response and the new sensor measurements. Such inverse-problem approach is often ill-posed and requires many online PDE solves, which is therefore not suitable for real-time damage assessment [16].

1.1. A short review of existing methods for damage identification

Data-driven SHM is a very broad topic and has been studied from many different points of view in the civil engineering and aerospace communities. Non-destructive evaluation and testing (NDE/NDT) technologies are often classified in two categories: wave-based or vibration-based. We highlight the works related to diagnostics Lamb waves and wavelet transforms, which are often integrated with piezoelectric sensors/actuators (see e.g., [25, 32, 21, 50]). This line of work focuses primarily on diagnostic signal generation and signal processing and it aims at measuring the change in the received signals after sending diagnostic stress or ultrasonic waves along the structures. Alternatively, works considering the changes in natural frequencies and mode shape as a consequence of flaws in structures under ambient excitations, as for example [12, 30, 43], are worth mentioning.

Despite the numerous works related to structural damage identification, only few combine machine learning techniques with numerical simulations. In [63], the authors propose to use a neural network classifier to measure the size of cracks by using synthetic data generated with 2D finite element models of cracked rivet holes under the propagation of longitudinal wave modes. The performance is tested on experimental data of specimens containing similarly sized cracks. Similarly, in [34] simulations are used to generate waveforms, which are then used to train a neural network to either classify crack types or identify their locations. Both the training and test sets are obtained by extracting a few relevant features from the synthetic response to better distinguish salient characteristics of different flaw classes. Aerospace applications are presented in [31], where real time sensor information are compared to simulation data from precomputed damaged scenarios to update the estimates of vehicle capabilities using a Bayesian classification process. In the recent work [54], the authors propose a simulation-based procedure for classification by comparing the performance of four machine learning techniques. The dataset is generated by exploiting parametric model order reduction techniques to make the computational effort of constructing the synthetic database affordable, while an experimental apparatus is used for testing. An a priori error analysis is provided to link the nominal performance on synthetic data to experimental performance.

While novelty detection is popular in the structural damage identification community (see e.g., [36, 10, 3]), it has, to the authors knowledge, never been studied when combined with synthetic datasets.

1.2. Our contribution and outline

The main contributions of this paper are:

- By making the realistic assumption that real sensors measure time signals of a predefined quantity, e.g., displacement or accelerations, we solve the PDEs in the whole domain and create a dataset of time signals, extracted at the sensors locations. Instead of considering a time discretization, we solve the PDE in the *frequency domain* and reconstruct the time-signals by using a numerical inverse Laplace transform. The latter allows us to recover information of the transient phase, which is a key feature for the classification phase.

- Since machine learning algorithms are well-known to behave better when using a large dataset [5], collecting a synthetic database requires a *model order reduction* approach to overcome the computational burden involved in the repeated solution of pPDEs. As employed in other works of simulation-based SHM [54, 31], we use the Reduced Basis method, a projection-based method whose key idea is to reconstruct the solution for a new parameter as a linear combination of suitable basis functions generated from the high-fidelity problem. In particular, for stability reasons, we rely on a proper symplectic decomposition with a symplectic Galerkin projection.

- We propose an *anomaly detection* procedure where the database is constructed from synthetic sensor data obtained from undamaged configurations only. Features are then extracted from this baseline system. Any subsequent data, which may originate from either a healthy or a damaged configuration, can be tested to verify if it conforms with the generated dataset. This allows a binary classification: it either belongs to the cluster of previously considered healthy signals, i.e., it is an inlier, or it is an outlier. This corresponds to a *semi-supervised learning* approach, also called *one-class classification* method, where labelled data, belonging to the "normal" class, are used in the training phase and unlabelled data from both classes are used in the test phase to identify abnormal data which deviate from the normal model [46, 19]. With one-class algorithms it is possible to *locate the damage* by training a different classifier for each sensor, based on the measurements collected at this sensor (see e.g., [36]).

  This procedure is sometimes called novelty or outlier detection and is an alternative to supervised or unsupervised anomaly detection techniques. In the former case, the training set is composed of fully labelled data, obtained from both healthy and damaged structures by predefining a number of exhaustive configuration classes for the described system. The classifier then maps each new sensor data to one of the anticipated classes. The advantage of our approach over supervised learning methods is substantial as there is no need to model all possible types of damage in a structure. This represents a significant gain in terms of development cost and computational time, e.g., we can consider physical parametrizations only, without having to include complex geometrical parametrizations in the Reduced Basis model. Furthermore, it is unrealistic to anticipate all types of damage and the number of different classification labels may grow rapidly. Unsupervised learning, instead, does not require any label and it does no distinction between training and test phases. The anomaly detection algorithm is based solely on intrinsic properties of the dataset, typically using a distance- or density-based approach [19]. This alternative is not an option for our simulation-based approach, where labels of generated data are always available.

- In addition to 2D studies, we also present 3D digital twins examples, where experimental data from damaged and undamaged structures are replaced with noisy synthetic data. However, the presented methodology is general and permits the incorporation of experimental data, after providing a suitable model calibration.

The reminder of the paper is organized as follows. Section 2 presents the general data-driven approach and highlights the decomposition of tasks into two phases: expensive offline simulations to fully characterize the response of healthy structures, followed by the training of a classifier to be used for rapid online testing of new experimental sensor responses. These concepts are further developed in Sections 3 and 4. In the former, we provide the mathematical details to construct the database by emphasizing the important role of MOR and, in the latter, we illustrate the classification strategy and the choice of features which act as damage indicators. Numerical examples in 2 and 3 dimensions with quantitative and qualitative analysis are presented in Section 5. Conclusions, remarks, and future developments are offered in Section 6.

2. A data-driven offline-online decomposition

In this section, we describe the general setup for our data-driven approach. As mentioned previously, a data-based strategy comprises two phases: an offline expensive phase consisting in the collection of a dataset used to train a classifier followed by a fast online phase where the classifier is employed to monitor the structure based on new measurements. For SHM procedures, the assembly of the database can be done either by using experimental data from the structure or similar structures, or by performing *synthetic experiments* based on a parametrized model, approximating the structural dynamics under the effect of a source [16]. In this work, we rely solely on synthetic measurements to demonstrate the overall workflow. Furthermore, accurate datasets based on physical experiments are rarely available and often lack a comprehensive description of the *natural variations* of the structure of interest [16]. Here, we generate synthetic sensors measurements from *healthy structures* only, without the ambition of representing all possible system configurations. Indeed, our goal is to capture the baseline (uncertain) operational and environmental conditions, to create a robust database of signals reflecting healthy structure behaviors. The parameters that express such variations are physical and are typically related to the material properties, the boundary or initial conditions or the source term. Geometric parameterizations are not included here as we only consider one healthy structure at a time with the assumption that its geometrical properties are not uncertain. However, this is not an essential assumption.

In practice, let $\Omega \subset \mathbb{R}^d$, with $d = \{2, 3\}$, be an open bounded domain associated with the structure of interest, $[0, T]$ the time domain related to the temporal measurements and $\mathcal{P} \subset \mathbb{R}^p$ the parameter space with p being the number of parameters used to characterize the model. Given a generic parametric model with suitable boundary and initial conditions, for a given $\boldsymbol{\mu} \in \mathcal{P}$, we seek the vector-valued solution $\boldsymbol{u} := \boldsymbol{u}(\boldsymbol{x}, t; \boldsymbol{\mu}) : \Omega \times [0, T] \times \mathcal{P} \to \mathbb{R}^d$ such that

$$\frac{\partial^2 \boldsymbol{u}}{\partial t^2} + \mathcal{L}^{\text{damp}} \left[\frac{\partial \boldsymbol{u}}{\partial t}; \boldsymbol{\mu} \right] + \mathcal{L} \left[\boldsymbol{u}; \boldsymbol{\mu} \right] = h(t; \boldsymbol{\mu}) s(\boldsymbol{x}; \boldsymbol{\mu}) \tag{1}$$

and evaluate a relevant output of interest

$$g_i(t; \boldsymbol{\mu}) := \ell(\boldsymbol{u}(\boldsymbol{x}_i, t; \boldsymbol{\mu}); \boldsymbol{\mu}), \quad \text{for } i = 1, \dots, N_s, \text{ and } t \in [0, T]. \tag{2}$$

In (1), $\mathcal{L}^{\text{damp}}[\cdot, \boldsymbol{\mu}]$ and $\mathcal{L}[\cdot, \boldsymbol{\mu}]$ are linear operators, representing damping and elasticity, respectively, while $h : \mathbb{R} \times \mathcal{P} \to \mathbb{R}^d$ and $s : \Omega \times \mathcal{P} \to \mathbb{R}^d$ represent the source dependencies with respect to time and space, respectively. In particular, $h(t; \boldsymbol{\mu})$ is often called a control function and, in this study, it mimics the effect of an active source on the structure, possibly excited by piezoelectric actuators or shakers (see e.g., [54, 65]). Moreover, the parameter-dependent output functional $\ell : \mathbb{R}^d \times \mathcal{P} \to \mathbb{R}^q$ maps the time-signals, evaluated at locations $\boldsymbol{x}_i \in \Omega$, into q-dimensional vectors that emulate the real sensor measurements, e.g., local displacements, accelerations, or strains. The spatial locations $\{\boldsymbol{x}_i\}_{i=0}^{N_s - 1}$ represent an approximation to the position of each of the N_s sensors attached to the structure. In this framework, the time-dependent experimental sensor measurements $g_i^{\text{exp}}(t) : \mathbb{R} \to \mathbb{R}^q$ are given by

$$g_i^{\text{exp}}(t) = g_i(t; \boldsymbol{\mu}) + \varepsilon_i, \quad \text{for } i = 1, \dots, N_s, \text{ and } t \in [0, T],$$

where $\varepsilon_i \sim \mathcal{N}(0, \gamma_i^2)$ and $\gamma_i \in \mathbb{R}$ is a priori unknown.

The first goal of the offline phase is to generate N_s (one per sensor) synthetic time-signals by evaluating (2) for many values of the input parameters $\boldsymbol{\mu} \in \mathcal{P}$. With the aim of representing the natural variation of healthy configurations under normal behavior, we generate a set of N_{tr} parameters

$$\boldsymbol{\Xi}^{N_{tr}} := \{\boldsymbol{\mu}_m\}_{m=1}^{N_{tr}}, \tag{3}$$

obtained by either uniformly sampling from the parameter space $\mathcal{P}$ or by leveraging a Bayesian approach. Here, for model calibration, we assume the probability distribution of such model parameters to be known a priori, e.g., provided by engineering experience. For the sake of simplicity, but without loss of generality, only uniform distributions are considered. The numerical solutions, obtained by solving (1) N_{tr} times, once per each parameter in $\boldsymbol{\Xi}^{N_{tr}}$, are evaluated at the sensor locations to obtain the outputs of interest (2). Assuming the interval $[0, T]$ is partitioned into N_t equal subintervals, the discrete time-signals are obtained by evaluating the output of interest (2) at time $t_n := n \frac{T}{N_t}$ for $n = 0, \dots, N_t$, i.e.,

$$\mathbf{g}_i^m := [g_i(t_0; \boldsymbol{\mu}_m), g_i(t_1; \boldsymbol{\mu}_m), \dots, g_i(t_{N_t}; \boldsymbol{\mu}_m)] \quad \text{for } i = 1, \dots, N_s, \text{ and } m = 1, \dots, N_{tr}. \tag{4}$$

We observe that $\mathbf{g}_i^m \in \mathbb{R}^{q \times (N_t + 1)}$ and, in the following, we use the interchangeable notation $\mathbf{g}_i^m = \mathbf{g}_i(\boldsymbol{\mu}_m)$. The synthetic datasets correspond to the collection of these time signals, i.e.,

$$\mathcal{D}_i^{N_{tr}} := \{\mathbf{g}_i^m\}_{m=1}^{N_{tr}}, \quad \text{for } i = 1, \dots, N_s. \tag{5}$$

We remark that $\mathcal{D}_i^{N_{tr}} \in \mathbb{R}^{N_{tr} \times q \times (N_t+1)}$.

The second part of the offline phase consists in the training of N_s *one-class classifiers* based on the database of synthetic healthy signals (5). More specifically, from each sample we first extract $Q \in \mathbb{R}$ engineering-based features, assumed to be damage-sensitive indicators, by using an ad-hoc feature function $\mathcal{F} : \mathbb{R}^{q \times (N_t+1)} \to \mathbb{R}^Q$. In practice, let $\mathcal{F}_i^{N_{tr}} \in \mathbb{R}^{N_{tr} \times Q}$ be the feature-based database of signals at location $\boldsymbol{x}_i$, obtained by applying $\mathcal{F}$ to each sample of $\mathcal{D}_i^{N_{tr}}$, i.e.,

$$\mathcal{F}_i^{N_{tr}} := \left\{ \mathcal{F}\left(\mathbf{g}_i(\boldsymbol{\mu}_m)\right) \right\}_{m=1}^{N_{tr}}, \quad \text{for } i = 1, \ldots, N_s. \tag{6}$$

Then, each classifier $f_i^{N_{tr}} : \mathbb{R}^Q \to \mathbb{R}$ is constructed as

$$[f_i^{N_{tr}}] := \texttt{OC-ML}\left(\mathcal{F}_i^{N_{tr}}\right), \quad \text{for } i = 1, \ldots, N_s, \tag{7}$$

where `OC-ML` is a one-class Machine Learning (OC-ML) technique.

Finally, during the online phase, these classifiers are used to detect possible anomalies in new sensor data. The classifier will be able to distinguish data generated from an undamaged structure from data generated from a damaged one. Indeed, a new datum $\mathbf{g}_i^\star := [g_i^{exp}(t_0), \ldots, g_i^{exp}(t_{N_t})]$ is classified as outlier if $f_i^{N_{tr}}(\mathcal{F}(\mathbf{g}_i^\star)) < 0$ and as an inlier otherwise. More precisely, by looking at which sensor signals $\mathbf{g}_i^\star$ are classified as outliers, we can retrieve information about the position of the damage and its severity. For major damages, many sensors will be classified as outliers, while for minor, localized damages, only the signals obtained by evaluating the solution at sensors close to the damage will be classified as outliers. Moreover, the absolute value of $f_i^{N_{tr}}\left(\mathcal{F}\left(\mathbf{g}_i^\star\right)\right)$ gives information about the uncertainty of belonging to one of the two classes: higher values correspond to a higher confidence on the output. In practice, we replace real experimental sensor data with noisy simulated data using new, unseen sampled parameters, i.e., $\boldsymbol{\Xi}^{N_{test}} := \{\boldsymbol{\mu}_m^\star\}_{m=1}^{N_{test}} \in \mathcal{P}$. We expect $f_i^{N_{tr}}(\mathbf{g}_i(\boldsymbol{\mu}_m^\star) + \varepsilon_i)$ to be positive for all $m = 1, \ldots, N_{test}$ and all $i = 1, \ldots, N_s$ if the variance γ^2 of the additional noise is sufficiently small. To simulate the response of damaged structures we replace the domain, used to generate the healthy database, with different faulty domains, i.e., we modify the domain Ω to include cracks of different sizes and located at different positions. We expect $f_i^{N_{tr}}(\mathbf{g}_i(\boldsymbol{\mu}^\star) + \varepsilon_i)$ to be negative if $\mathbf{g}_i(\boldsymbol{\mu}^\star)$ is generated by solving (1) for $\boldsymbol{\mu}^\star \in \mathcal{P}$ over a damaged domain with a crack close to the i-th sensor. Signals obtained on healthy domains, but generated using an input parameter outside the baseline operational range $\mathcal{P}$, are also expected to be classified as outliers. However, in this work, only geometrical flaws are considered.

To summarize, the flow chart in Figure 1 gives an overview of the data-driven one-class classification problem with synthetic data and highlights the separation of the offline and online phases.

3. A database of time series using a parametrized mathematical model

3.1. Problem setup: the acoustic-elastic wave equation

Throughout this work, we consider (1) to be the acoustic-elastic wave PDE and Ω a d-dimensional domain approximating a healthy structure of interest. The acoustic-elastic wave equation in strong form, equipped with suitable boundary conditions on the piecewise smooth boundary $\Gamma = \partial\Omega$ and initial conditions for both the displacement field and its derivative, is expressed as:

$$\begin{cases} \rho \dfrac{\partial^2 \boldsymbol{u}}{\partial t^2} + \rho\eta \dfrac{\partial \boldsymbol{u}}{\partial t} - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}; \boldsymbol{\mu}) = h(t; \boldsymbol{\mu}) s(\boldsymbol{x}; \boldsymbol{\mu}) & \text{in } \Omega \times (0, T] \\ \boldsymbol{u} = \boldsymbol{g}_D(\boldsymbol{x}, t; \boldsymbol{\mu}) & \text{on } \Gamma_D \times (0, T] \\ \boldsymbol{\sigma}(\boldsymbol{u}; \boldsymbol{\mu}) \cdot \hat{\boldsymbol{n}} = \boldsymbol{g}_N(\boldsymbol{x}, t; \boldsymbol{\mu}) & \text{on } \Gamma_N \times (0, T] \\ \boldsymbol{u}|_{t=0} = \boldsymbol{u}_0(\boldsymbol{x}; \boldsymbol{\mu}) & \text{in } \Omega \\ \dfrac{\partial \boldsymbol{u}}{\partial t}\Big|_{t=0} = \boldsymbol{v}_0(\boldsymbol{x}; \boldsymbol{\mu}) & \text{in } \Omega \end{cases}, \tag{8}$$

where $\boldsymbol{u}$ represents the displacement field, ρ is the density coefficient, η is a non-dimensional damping coefficient, $h := h(t; \boldsymbol{\mu})$ and $\boldsymbol{s} := \boldsymbol{s}(\boldsymbol{x}; \boldsymbol{\mu})$ are the source functions, describing the time and space dependency,

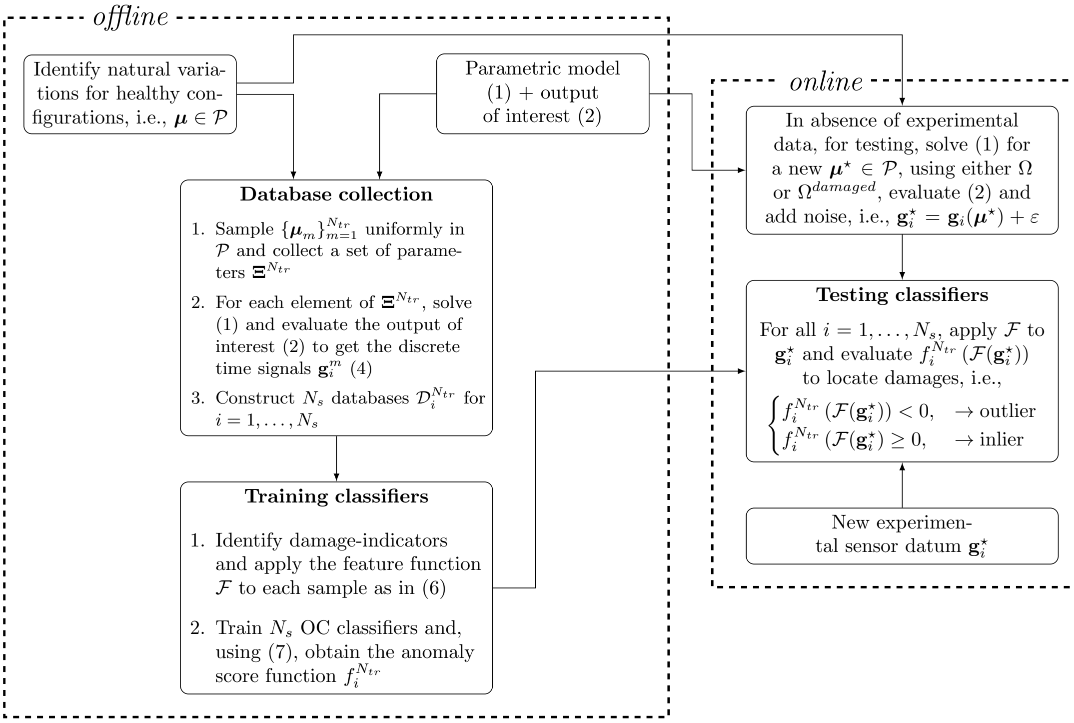

Figure 1: Workflow chart to synthesize the offline and online phases of simulation-based SHM procedure.

respectively, and $\boldsymbol{\sigma} \coloneqq \boldsymbol{\sigma}(\boldsymbol{u}; \boldsymbol{\mu})$ is the stress tensor

$$\boldsymbol{\sigma} \coloneqq 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda \mathrm{tr}\left(\boldsymbol{\varepsilon}(\boldsymbol{u})\right)\mathbb{I}, \tag{9}$$

where $\mathbb{I}$ is the d dimensional identity matrix, $\mathrm{tr}(\cdot)$ is the trace operator applied to the strain tensor

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T}{2},$$

and the Lamé constants μ and λ are immediately derived by E, the Young's modulus, and ν, the non-dimensional Poisson's ratio, as

$$\mu = \frac{E}{2(1+\nu)} \quad \text{and} \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}. \tag{10}$$

In (8), $\hat{\boldsymbol{n}}$ is the outward normal vector to Γ. Γ_D and Γ_N are such that $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$ and they represent the portions of the surface of Ω where displacement boundary conditions $\boldsymbol{g}_D \coloneqq \boldsymbol{g}_D(\boldsymbol{x}, t; \boldsymbol{\mu})$ and stress boundary conditions through the traction vector $\boldsymbol{g}_N \coloneqq \boldsymbol{g}_N(\boldsymbol{x}, t; \boldsymbol{\mu})$ are applied, respectively. We note that, alternatively, one could prescribe *free slip* boundary conditions:

$$\begin{cases} \boldsymbol{u} \cdot \hat{\boldsymbol{n}} = \boldsymbol{0} \\ (\boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}}) \cdot \boldsymbol{\tau} = \boldsymbol{g}_N \end{cases} \text{on } \partial\Omega, \tag{11}$$

where $\boldsymbol{\tau}$ is the tangential vector to Γ. For the sake of simplicity and consistent with the numerical tests, we consider zero Dirichlet and Neumann data; the non-homogeneous case can be treated similarly. Finally, $\boldsymbol{u}_0 \coloneqq \boldsymbol{u}_0(\boldsymbol{x}; \boldsymbol{\mu})$ and $\boldsymbol{v}_0 \coloneqq \boldsymbol{v}_0(\boldsymbol{x}; \boldsymbol{\mu})$ describe the initial displacement and velocity in space, respectively.

In the remaining section we consider $\boldsymbol{\mu}$ to be a generic parameter which can be related to the material properties, the boundary conditions, the initial conditions or the source functions h and s. In a real setup, the choice of these physical parameters together with their probability distribution is inferred by experimental results and prior engineering knowledge.

3.2. The discretized problem in time domain

To provide the discrete form of (8) with homogenous boundary conditions, i.e., $\boldsymbol{g}_D = \mathbf{0}$ and $\boldsymbol{g}_N = \mathbf{0}$, we introduce its weak formulation. For a fixed parameter $\boldsymbol{\mu} \in \mathcal{P}$ and a fixed $t \in (0,T]$, find $\boldsymbol{u}(t;\boldsymbol{\mu}) \in V \coloneqq \{\boldsymbol{w} \in [H^1(\Omega;\mathbb{R}^d)]^d : \boldsymbol{w}|_{\Gamma_D} = \mathbf{0}\}$ [2] such that

$$\rho m\left(\frac{\partial^2 \boldsymbol{u}(t;\boldsymbol{\mu})}{\partial t^2}, \boldsymbol{\psi}\right) + \rho\eta\, m\left(\frac{\partial \boldsymbol{u}(t;\boldsymbol{\mu})}{\partial t}, \boldsymbol{\psi}; \boldsymbol{\mu}\right) + a(\boldsymbol{u}(t;\boldsymbol{\mu}), \boldsymbol{\psi}; \boldsymbol{\mu}) = h(t;\boldsymbol{\mu}) f(\boldsymbol{\psi};\boldsymbol{\mu}), \tag{12}$$

for all $\boldsymbol{\psi} \in V$ with $\boldsymbol{u}(0) = \boldsymbol{u}_0$ and $\frac{\partial \boldsymbol{u}(t)}{\partial t}\big|_{t=0} = \boldsymbol{v}_0$. In (12), the bilinear forms $m(\cdot,\cdot)$ and $a(\cdot,\cdot;\boldsymbol{\mu})$ and the functional $f(\cdot;\boldsymbol{\mu})$ have the following expressions

$$\begin{aligned} m(\boldsymbol{u},\boldsymbol{\psi}) &\coloneqq \int_\Omega \boldsymbol{u}(t;\boldsymbol{\mu}) \cdot \boldsymbol{\psi}\, d\Omega, \\ a(\boldsymbol{u},\boldsymbol{\psi};\boldsymbol{\mu}) &\coloneqq \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}(t;\boldsymbol{\mu});\boldsymbol{\mu}) : \nabla\boldsymbol{\psi}\, d\Omega \\ &= \int_\Omega \left(2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}(t;\boldsymbol{\mu})) : \boldsymbol{\varepsilon}(\boldsymbol{\psi}) + \lambda(\nabla\cdot\boldsymbol{u}(t;\boldsymbol{\mu}))(\nabla\cdot\boldsymbol{\psi})\right) d\Omega, \\ f(\boldsymbol{\psi};\boldsymbol{\mu}) &\coloneqq \int_\Omega \boldsymbol{s}(\boldsymbol{\mu}) \cdot \boldsymbol{\psi}\, d\Omega \end{aligned} \tag{13}$$

where, in the definition of $a(\cdot,\cdot;\boldsymbol{\mu})$, we have used the definition of the stress tensor (9) and the fact that

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) : \nabla\boldsymbol{\psi} = \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{\psi})$$

and

$$\mathrm{tr}(\boldsymbol{\varepsilon}(\boldsymbol{u}))\mathbb{I} : \nabla\boldsymbol{\psi} = (\nabla\cdot\boldsymbol{u})\mathbb{I} : \nabla\boldsymbol{\psi} = (\nabla\cdot\boldsymbol{u})(\nabla\cdot\boldsymbol{\psi}).$$

The weak formulation is discretized in space by introducing an approximation for the displacement in a finite-dimensional subspace to obtain a linear system of ordinary differential equations. Let us introduce a triangulation $\mathcal{T}_h$ of the domain Ω, i.e., K non-overlapping triangles ($d = 2$) or tetrahedra ($d = 3$) and the FE space $X_h^r = \{\boldsymbol{w}_h \in C^0(\bar{\Omega}) : \boldsymbol{w}_h|_K \in \mathbb{P}_r\, \forall K \in \mathcal{T}_h\}$, where h represents the mesh size[3], i.e., $h_K \coloneqq \mathrm{diam}(K) \leq h, \forall K \in \mathcal{T}_h$. Consider $V_h \coloneqq V \cap X_h^r$ as a conforming finite-dimensional subspace of V and $\{\boldsymbol{\varphi}_j \in \mathbb{R}^d\}_{j=1}^{N_h}$ as a basis for V_h, we define

$$\boldsymbol{u}_h(\boldsymbol{x},t;\boldsymbol{\mu}) \coloneqq \sum_{j=1}^{N_h} u_j(t;\boldsymbol{\mu})\boldsymbol{\varphi}_j(\boldsymbol{x}), \tag{14}$$

where $N_h \coloneqq \dim(V_h)$ is the number of degrees of freedom (DOFs) which depends on the number of physical variables, the underlying mesh and the polynomial order r of the FE discretization. Moreover, if we denote by $\mathbf{u}_h(t;\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ the vector having as components the unknown coefficients $u_j(t;\boldsymbol{\mu})$ then, at the algebraic level, we obtain the discrete system

$$\rho\mathbf{M}\left(\frac{\partial^2 \mathbf{u}_h}{\partial t^2}(t;\boldsymbol{\mu}) + \eta\frac{\partial \mathbf{u}_h}{\partial t}(t;\boldsymbol{\mu})\right) + \mathbf{A}(\boldsymbol{\mu})\mathbf{u}_h(t;\boldsymbol{\mu}) = h(t;\boldsymbol{\mu})\mathbf{f}(\boldsymbol{\mu}), \tag{15}$$

where $\mathbf{M} \in \mathbb{R}^{N_h \times N_h}$ is the mass matrix with elements $\mathbf{M}_{ij} = m(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i)$, $\mathbf{A} \coloneqq \mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is the stiffness matrix with elements $\mathbf{A}_{ij} = a(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i;\boldsymbol{\mu})$ and $\mathbf{f} \coloneqq \mathbf{f}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is the vector with components $\mathbf{f}_i = f(\boldsymbol{\varphi}_i;\boldsymbol{\mu})$.

[2] Note that throughout this work we slightly abuse the notation by considering $\boldsymbol{u}(t) \in V$ for all $t \in (0,T]$, while it would be more precise to consider $\boldsymbol{u} \in C^2\left([0,T];[L^2(\Omega;\mathbb{R}^d)]^d\right) \cap C^0\left([0,T],V\right)$. Moreover, we note that when one seeks to solve (8) with free slip boundary conditions (11), V has to be replaced with $V_{fs} = \{\boldsymbol{w} \in [H^1(\Omega;\mathbb{R}^d)]^d : \boldsymbol{w}\cdot\hat{\boldsymbol{n}} = \mathbf{0}\}$.

[3] The mesh size h should not be confused with the time-dependent source function $h \coloneqq h(t;\boldsymbol{\mu})$.

To obtain a fully discretized system, we use the classic Newmark method, defined in [42], for the time discretization of the second order initial value problem (15). Let us first consider a partition of the interval $[0, T]$ in N_t subintervals of equal size $\Delta t = \frac{T}{N_t}$, such that $t_n = n\Delta t$, $\forall n = 0, \ldots, N_t$. Moreover, we denote by $\mathbf{u}_h^n(\boldsymbol{\mu}) \coloneqq \mathbf{u}_h(t_n; \boldsymbol{\mu})$ the displacement, $\mathbf{v}_h^n(\boldsymbol{\mu}) \coloneqq \frac{\partial \mathbf{u}_h(t;\boldsymbol{\mu})}{\partial t}\big|_{t=t_n}$ the velocity, and $\mathbf{a}_h^n(\boldsymbol{\mu}) \coloneqq \frac{\partial^2 \mathbf{u}_h(t;\boldsymbol{\mu})}{\partial t^2}\big|_{t=t_n}$ the acceleration vectors at time t_n, respectively. The Newmark method is defined as

$$\mathbf{u}_h^{n+1} \coloneqq \mathbf{u}_h^n + \Delta t \mathbf{v}_h^n + (\Delta t)^2 \left(\beta \mathbf{a}_h^{n+1} + \frac{1-2\beta}{2} \mathbf{a}_h^n \right), \tag{16a}$$

$$\mathbf{v}_h^{n+1} \coloneqq \mathbf{v}_h^n + \Delta t \left(\zeta \mathbf{a}_h^{n+1} + (1-\zeta) \mathbf{a}_h^n \right), \tag{16b}$$

where β and ζ are constant parameters. This method is implicit unless $\beta = \zeta = 0$ and it is unconditionally stable if $2\beta \geq \zeta \geq \frac{1}{2}$. In this work we fix $\zeta = 2\beta = \frac{1}{2}$, which corresponds to a popular second order method, even if spurious oscillatory solutions may arise for long time intervals (see e.g., [48, 64]).

If in (15) we replace $\mathbf{u}_h(t; \boldsymbol{\mu})$ and $\frac{\partial \mathbf{u}_h(t;\boldsymbol{\mu})}{\partial t}$ with the expressions in (16a) and (16b), respectively, and solve for $\mathbf{a}_h^{n+1}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, we obtain the fully discrete linear system:

$$\mathbf{K}(\boldsymbol{\mu}) \mathbf{a}_h^{n+1}(\boldsymbol{\mu}) = \mathbf{q}^{n+1}(\boldsymbol{\mu}), \tag{17}$$

where $\mathbf{K} \coloneqq \mathbf{K}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ and $\mathbf{q}^{n+1} \coloneqq \mathbf{q}^{n+1}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ have the following expression

$$\begin{aligned} \mathbf{K} &\coloneqq \rho \left(1 + \eta \zeta \Delta t\right) \mathbf{M} + \beta (\Delta t)^2 \mathbf{A}(\boldsymbol{\mu}), \\ \mathbf{q}^{n+1} &\coloneqq h^{n+1}(\boldsymbol{\mu}) \mathbf{f}(\boldsymbol{\mu}) - \mathbf{A}(\boldsymbol{\mu}) \mathbf{u}_h^n(\boldsymbol{\mu}) - (\rho \eta \mathbf{M} + \Delta t \mathbf{A}(\boldsymbol{\mu})) \mathbf{v}_h^n(\boldsymbol{\mu}) \\ &\quad - \left(\rho \eta (1-\zeta) \Delta t \mathbf{M} + \frac{1-2\beta}{2} (\Delta t)^2 \mathbf{A}(\boldsymbol{\mu}) \right) \mathbf{a}_h^n(\boldsymbol{\mu}), \end{aligned}$$

where $h^n(\boldsymbol{\mu}) \coloneqq h(t_n; \boldsymbol{\mu})$. Hence, the semi-discrete variational problem (15) is equivalent to the following statement: for $n = 0, \ldots, N_t - 1$, solve (17) for $\mathbf{a}_h^{n+1}(\boldsymbol{\mu})$ and update $\mathbf{u}_h^{n+1}(\boldsymbol{\mu})$ and $\mathbf{v}_h^{n+1}(\boldsymbol{\mu})$ using the Newmark method (16). We observe that both $m(\cdot, \cdot)$ and $a(\cdot, \cdot; \boldsymbol{\mu})$ are symmetric and coercive bilinear forms, where, for the coerciveness of a, we have used Korn's inequality [23]. This guarantees that $\mathbf{K}$ is invertible. Moreover, note that the initial conditions for $\mathbf{u}_h^0(\boldsymbol{\mu})$ and $\mathbf{v}_h^0(\boldsymbol{\mu})$ are given, while $\mathbf{a}_h^0(\boldsymbol{\mu})$ must be recovered by solving (17) with $\mathbf{q}^0(\boldsymbol{\mu}) = h^0(\boldsymbol{\mu}) \mathbf{f}(\boldsymbol{\mu})$.

3.3. The need for a reduced order model

As introduced in Section 2, our goal is to construct N_s synthetic databases $\mathcal{D}_i^{N_{tr}}, i = 1, \ldots, N_s$ as defined in (5). In the numerical examples, the generic output of interest (2) will be given by the local displacement, i.e., the solution of (8) at the sensors locations:

$$g_i(t_n; \boldsymbol{\mu}_m) \coloneqq \boldsymbol{u}_h(\boldsymbol{x}_i, t_n; \boldsymbol{\mu}_m) \in \mathbb{R}^d, \tag{18}$$

with $\boldsymbol{u}_h(\cdot, \cdot; \boldsymbol{\mu})$ defined in (14). In the literature, sensor measurements often correspond to displacements or accelerations, see e.g., [36]. Moreover, we highlight that the location of the i-th sensor, i.e., $\boldsymbol{x}_i \in \Omega$, may not belong to the triangularization $\mathcal{T}_h$ introduced in the previous section, i.e., $\boldsymbol{x}_i$ is not necessarily a DOF. The construction of such databases requires the solution of (8) N_{tr} times, using N_{tr} different input parameters $\boldsymbol{\mu}_m \in \mathcal{P}$. In particular, the linear system (17) with N_h DOFs has to be solved $N_{tr} N_t$ times. This suggests that, in a many-query context when either the number of DOFs or the number of time steps is large, solving the full-order model is not affordable. Indeed, in our damage-detection setting, we need many samples to build robust classifiers.

We therefore introduce a strategy that, on one hand, reduces the number of times we need to solve the linear system (17), and, on the other hand, replaces the original FE high-fidelity problem with a reduced order model without compromising the overall accuracy. The former point is achieved by replacing the time domain with the frequency domain, combined with the use of the Laplace transform of the displacement as unknown field, described in detail in Section 3.4. Since we are also interested in reconstructing the time history of the displacement, we employ a numerical inverse Laplace transform strategy, the details of which are provided in Section 3.5. The reduced order model in space is obtained using the reduced basis method, discussed in Section 3.6.

3.4. The Laplace domain

When considering the translation of a time-dependent PDE into frequency domain, we face the choice of the transform to use. Popular choices in the structural damage detection field are the Fourier transform (see e.g., [54]) or the Laplace transform as in [65], where the authors model the behavior of smart structures combined with piezoelectric actuators and sensors using the boundary element method applied to the elastodynamics equation. Here, we also choose the Laplace transform to allow the study of the *transient response* of damaged structures when using active sources to excite the structure. The Fourier transform is a suitable alternative if we study the periodic behavior of the vibrations of a structure under the effect of continuous sources, e.g., wind, waves or tides. The choice of the Laplace transform will be better motivated in Section 4.2, where we discuss the damage sensitive features extracted from raw time signals.

Given a fixed frequency $z \in \mathbb{C}$ and a fixed input parameter $\boldsymbol{\mu} \in \mathcal{P}$, by multiplying the acoustic-elastic wave equation (8) by e^{-zt} and integrating in time over the infinite interval $[0, \infty)$, the time-dependent problem reduces to the computation of the Laplace transform of $\boldsymbol{u}$ evaluated at z, i.e., find $\tilde{\boldsymbol{u}} \coloneqq \tilde{\boldsymbol{u}}(\boldsymbol{x}, z; \boldsymbol{\mu}) : \Omega \times \mathbb{C} \times \mathcal{P} \to \mathbb{C}^d$ such that

$$\begin{cases} \rho(z^2 + z\eta)\tilde{\boldsymbol{u}} - \nabla \cdot \boldsymbol{\sigma}(\tilde{\boldsymbol{u}}; \boldsymbol{\mu}) = \tilde{h}(z; \boldsymbol{\mu}) s(\boldsymbol{x}; \boldsymbol{\mu}) & \text{in } \Omega \\ \tilde{\boldsymbol{u}} = \mathbf{0} & \text{on } \Gamma_D \\ \boldsymbol{\sigma}(\tilde{\boldsymbol{u}}; \boldsymbol{\mu}) \cdot \hat{\boldsymbol{n}} = \mathbf{0} & \text{on } \Gamma_N \end{cases}, \tag{19}$$

where, for the sake of simplicity, we have assumed homogenous boundary conditions and zero initial conditions. In (19) $\tilde{h} \coloneqq \tilde{h}(z; \boldsymbol{\mu}) : \mathbb{C} \times \mathcal{P} \to \mathbb{C}^d$ is the Laplace transform of the time-dependent part of the source function $h(t; \boldsymbol{\mu})$.

Since both $\boldsymbol{u}$ and $\tilde{\boldsymbol{u}}$ have the same dependency on the space variable $\boldsymbol{x} \in \Omega$, the space discretization derived in Section 3.2 applies here. Given $\tilde{V} \coloneqq \{\boldsymbol{w} \in [H^1(\Omega; \mathbb{C}^d)]^d : \boldsymbol{w}|_{\Gamma_D} = \mathbf{0}\}$ as the corresponding Hilbert space in frequency domain, the approximate Galerkin problem becomes: for all $z \in \mathbb{C}$ and all $\boldsymbol{\mu} \in \mathcal{P}$ find $\tilde{\boldsymbol{u}}_h(z; \boldsymbol{\mu}) \in \tilde{V}_h \coloneqq \tilde{V} \cap X_h^r$ such that

$$\rho\left(z^2 + \eta z\right) m(\tilde{\boldsymbol{u}}_h(z; \boldsymbol{\mu}), \boldsymbol{v}_h) + a(\tilde{\boldsymbol{u}}_h(z; \boldsymbol{\mu}), \boldsymbol{v}_h; \boldsymbol{\mu}) = \tilde{h}(z; \boldsymbol{\mu}) f(\boldsymbol{v}_h; \boldsymbol{\mu}), \quad \forall \boldsymbol{v}_h \in \tilde{V}_h, \tag{20}$$

where $\tilde{\boldsymbol{u}}_h$ is the Galerkin approximation of $\tilde{\boldsymbol{u}}$, while the bilinear forms $m(\cdot,\cdot)$, $a(\cdot,\cdot;\boldsymbol{\mu})$ and the functional $f(\cdot;\boldsymbol{\mu})$ are defined in (13). The discrete problem (20) is equivalent to a system of linear equations. In order to provide an algebraic formulation analogous to the time-dependent one in (15), we first introduce a complex canonical basis $\tilde{\boldsymbol{\varphi}}_j \in \mathbb{C}^d$ for $j = 1, \dots, N_h$ for the finite-dimensional space $\tilde{V}_h$. Note that each complex basis $\tilde{\boldsymbol{\varphi}}_j$ is either purely real or purely imaginary and all the mixed terms are obtained by their linear combinations. The N_h basis are therefore given by $N_h/2$ purely real basis and $N_h/2$ purely imaginary basis, i.e.,

$$\tilde{\boldsymbol{\varphi}}_j \coloneqq \boldsymbol{\psi}_j \mathbb{I}_{j \le \frac{N_h}{2}} + i \boldsymbol{\psi}_{N_h - j + 1} \mathbb{I}_{j > \frac{N_h}{2}}, \qquad \text{for } j = 1, \dots, N_h, \tag{21}$$

where i is the imaginary constant. Moreover, let

$$\tilde{\boldsymbol{u}}_h(\boldsymbol{x}, z; \boldsymbol{\mu}) \coloneqq \sum_{j=1}^{N_h} \tilde{u}_j(z; \boldsymbol{\mu}) \tilde{\boldsymbol{\varphi}}_j(\boldsymbol{x}). \tag{22}$$

If we denote by $\tilde{\mathbf{u}}_h(z; \boldsymbol{\mu})$ the vector having as components the unknown coefficients $\tilde{u}_j(z; \boldsymbol{\mu})$, solving problem (20) is equivalent to: find $\tilde{\mathbf{u}}_h(z; \boldsymbol{\mu}) \in \mathbb{C}^{N_h}$ such that

$$\left[\rho\left(z^2 + \eta z\right) \tilde{\mathbf{M}} + \tilde{\mathbf{A}}(\boldsymbol{\mu})\right] \tilde{\mathbf{u}}_h(z; \boldsymbol{\mu}) = \tilde{h}(z; \boldsymbol{\mu}) \tilde{\mathbf{f}}(\boldsymbol{\mu}), \tag{23}$$

where $\tilde{\mathbf{M}} \in \mathbb{C}^{N_h \times N_h}$ is the complex mass matrix with elements $\tilde{\mathbf{M}}_{ij} = m(\tilde{\boldsymbol{\varphi}}_j, \tilde{\boldsymbol{\varphi}}_i)$, $\tilde{\mathbf{A}} \coloneqq \tilde{\mathbf{A}}(\boldsymbol{\mu}) \in \mathbb{C}^{N_h \times N_h}$ is the stiffness matrix with elements $\tilde{\mathbf{A}}_{ij} = a(\tilde{\boldsymbol{\varphi}}_j, \tilde{\boldsymbol{\varphi}}_i; \boldsymbol{\mu})$ and $\tilde{\mathbf{f}} \coloneqq \tilde{\mathbf{f}}(\boldsymbol{\mu}) \in \mathbb{C}^{N_h}$ is the vector with components $\tilde{\mathbf{f}}_i = f(\tilde{\boldsymbol{\varphi}}_i; \boldsymbol{\mu})$. This system can be split into a set of $2N_h$ real equations such that, for a given $z \coloneqq \alpha + iy$, the solution of (23) can be rewritten as $\tilde{\mathbf{u}}_h(z; \boldsymbol{\mu}) \coloneqq \tilde{\mathbf{u}}_h^{\alpha}(\boldsymbol{\mu}) + i \tilde{\mathbf{u}}_h^{y}(\boldsymbol{\mu})$. This splitting is especially important for implementation purposes and, by simple manipulations, we obtain

$$\begin{bmatrix} \mathbf{K}^{\alpha}(\boldsymbol{\mu}) & -\mathbf{K}^{y}(\boldsymbol{\mu}) \\ \mathbf{K}^{y}(\boldsymbol{\mu}) & \mathbf{K}^{\alpha}(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_h^{\alpha}(z; \boldsymbol{\mu}) \\ \tilde{\mathbf{u}}_h^{y}(z; \boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{q}}^{\alpha}(z; \boldsymbol{\mu}) \\ \tilde{\mathbf{q}}^{y}(z; \boldsymbol{\mu}) \end{bmatrix}, \tag{24}$$

where

$$
\begin{aligned}
\mathbf{K}^{\alpha}(\boldsymbol{\mu}) &:= \Theta^{\alpha}\tilde{\mathbf{M}}^{\alpha} - \Theta^{y}\tilde{\mathbf{M}}^{y} + \tilde{\mathbf{A}}^{\alpha}(\boldsymbol{\mu}),\\
\mathbf{K}^{y}(\boldsymbol{\mu}) &:= \Theta^{\alpha}\tilde{\mathbf{M}}^{y} + \Theta^{y}\tilde{\mathbf{M}}^{\alpha} + \tilde{\mathbf{A}}^{y}(\boldsymbol{\mu}),\\
\tilde{\mathbf{q}}^{\alpha}(z;\boldsymbol{\mu}) &:= \tilde{h}^{\alpha}(z;\boldsymbol{\mu})\tilde{\mathbf{f}}^{\alpha}(\boldsymbol{\mu}) - \tilde{h}^{y}(z;\boldsymbol{\mu})\tilde{\mathbf{f}}^{y}(\boldsymbol{\mu}),\\
\tilde{\mathbf{q}}^{y}(z;\boldsymbol{\mu}) &:= \tilde{h}^{y}(z;\boldsymbol{\mu})\tilde{\mathbf{f}}^{\alpha}(\boldsymbol{\mu}) + \tilde{h}^{\alpha}(z;\boldsymbol{\mu})\tilde{\mathbf{f}}^{y}(\boldsymbol{\mu}).
\end{aligned}
\tag{25}
$$

Here, $\Theta^{\alpha} := \rho\left(\alpha^2 - y^2 + \eta\alpha\right)$, $\Theta^{y} := \rho y\,(2\alpha + \eta)$ and $\Theta^{\alpha} + i\Theta^{y} = \rho(z^2 + \eta z)$. In (24) and (25) we have used the following notation: $\tilde{\mathbf{M}}^{\alpha} \in \mathbb{R}^{N_h \times N_h}$ and $\tilde{\mathbf{M}}^{y} \in \mathbb{R}^{N_h \times N_h}$ are the real and imaginary parts of the mass matrix $\tilde{\mathbf{M}}$, respectively, with components

$$
\begin{aligned}
\tilde{\mathbf{M}}^{\alpha}_{ij} &= m(\boldsymbol{\psi}_j, \boldsymbol{\psi}_i)\mathbb{I}_{\left\{i,j \le \frac{N_h}{2}\right\}} - m(\boldsymbol{\psi}_{N_h-j+1}, \boldsymbol{\psi}_{N_h-i+1})\mathbb{I}_{\left\{i,j > \frac{N_h}{2}\right\}},\\
\tilde{\mathbf{M}}^{y}_{ij} &= m(\boldsymbol{\psi}_j, \boldsymbol{\psi}_{N_h-i+1})\mathbb{I}_{\left\{j \le \frac{N_h}{2},\, i > \frac{N_h}{2}\right\}} + m(\boldsymbol{\psi}_{N_h-j+1}, \boldsymbol{\psi}_i)\mathbb{I}_{\left\{i \le \frac{N_h}{2},\, j > \frac{N_h}{2}\right\}},
\end{aligned}
$$

where the real basis $\boldsymbol{\psi}_j$ is introduced in (21). We observe that, given $\tilde{\mathbf{M}}^{N_h/2} \in \mathbb{R}^{N_h/2 \times N_h/2}$ as the mass matrix with half degrees of freedom and components $\tilde{\mathbf{M}}^{N_h/2}_{ij} = m(\boldsymbol{\psi}_j, \boldsymbol{\psi}_i)$, $\tilde{\mathbf{M}}^{\alpha}$ and $\tilde{\mathbf{M}}^{y}$ have a block diagonal structure:

$$
\tilde{\mathbf{M}}^{\alpha} = \begin{bmatrix} \tilde{\mathbf{M}}^{N_h/2} & \mathbf{0} \\ \mathbf{0} & -\tilde{\mathbf{M}}^{N_h/2} \end{bmatrix}, \quad \tilde{\mathbf{M}}^{y} = \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{M}}^{N_h/2} \\ \tilde{\mathbf{M}}^{N_h/2} & \mathbf{0} \end{bmatrix}.
$$

The real and imaginary parts of the stiffness matrix $\tilde{\mathbf{A}}$ have the same expressions as the mass matrix by simply replacing $m(\cdot,\cdot)$ with $a(\cdot,\cdot;\boldsymbol{\mu})$. For the right-hand-side, we define $\tilde{h}^{\alpha}(z;\boldsymbol{\mu})$ and $\tilde{h}^{y}(z;\boldsymbol{\mu})$ to be the real and imaginary parts of $\tilde{h}(z;\boldsymbol{\mu})$, respectively, and $\tilde{\mathbf{f}}^{p} := \tilde{\mathbf{f}}^{p}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ with components $\tilde{\mathbf{f}}^{p}_{i} := f(\tilde{\boldsymbol{\varphi}}^{p}_{i};\boldsymbol{\mu})$ for $p \in \{\alpha, y\}$.

3.5. Recovering the time-dependent signals using the Weeks method

To recover the time signals (18) at all sensors locations we need to compute the inverse Laplace transform on the solution of (23) or (24). This corresponds to an integration over the infinite imaginary axis in the complex plane:

$$
\boldsymbol{u}(t) = \frac{e^{\alpha t}}{2\pi i}\int_{-\infty}^{\infty} e^{ity}\tilde{\boldsymbol{u}}(\alpha + iy)dy, \quad t > 0, \quad \alpha > \alpha_0,
\tag{26}
$$

where $\alpha \in \mathbb{R}$ is a free parameter greater than α_0[4], which is the rightmost real number for which $\tilde{\boldsymbol{u}}(\cdot)$ is defined. This integral, known as the Bromwich integral, is difficult to evaluate analytically, especially since $\tilde{\boldsymbol{u}}(\cdot)$ is here replaced with

$$
\tilde{g}_i(z_j;\boldsymbol{\mu}_m) := \tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, z_j;\boldsymbol{\mu}_m) \in \mathbb{C}^d,
\tag{27}
$$

where $\tilde{\boldsymbol{u}}_h(\cdot,\cdot;\boldsymbol{\mu})$ is defined in (22) and the expansion coefficients are obtained by solving (23) at discrete points $z_j := \alpha_j + iy_j$. Therefore, we need to approximate (26) by resorting to numerical inverse Laplace transform strategies.

Among three numerical inverse Laplace transform methods, reviewed in [14], i.e., the trapezoidal rule [13], Talbot's method [55] and the expansion in the Laguerre's polynomials, also known as the Weeks method [60, 37], we choose the latter one. Indeed, the former two are unfeasible: the complex inversion integral is obtained by a numerical quadrature where the nodes depend on the independent variable t. This means that, to reconstruct the entire discrete time series $\mathbf{g}^{m}_{i}$, introduced in (18), we need to solve (23) as many times as the number of time steps. As a result, the computational cost would be greater than solving the direct problem with the Newmark method. Instead, the Weeks method is obtained as an expansion in terms of the Laguerre's polynomials. The main advantage is that, once the expansion coefficients are determined, the Laplace transform and the inverse can be obtained at any value t_n by means of a simple series summation. We mention that there exists variants of the trapezoidal rule, relying on added correction terms (see e.g.,

[4]Note that this parameter is usually denoted by σ_0 in the literature, but here we choose α_0 to avoid confusion with the stress tensor.

[9, 15]), where the Laplace transform does not depend on time. These variants have been successfully used to reconstruct time histories with a time interval of T of the order of $10^{-4}, 10^{-5}$ seconds in [65]. However, they often become oscillatory and deviate from the right solution when T is large.

We briefly recall the Weeks method to retrieve a generic time signal, beginning with the representation

$$\boldsymbol{u}(t) = e^{(\alpha-b)t} \sum_{k=0}^{\infty} \boldsymbol{a}_k L_k(2bt), \tag{28}$$

where $b \in \mathbb{R}^+$ is a free parameter and $L_k(\cdot)$ denotes the Laguerre polynomial of degree k. The expansion coefficients $\boldsymbol{a}_k$, which depend on the Laplace transform $\tilde{\boldsymbol{u}}(z)$, are defined by a Maclaurin series

$$G(\omega; \alpha, b) := \frac{2b}{1-\omega} \tilde{\boldsymbol{u}} \left(\alpha + b \frac{1+\omega}{1-\omega} \right) = \sum_{k=0}^{\infty} a_k \omega^k,$$

where $\omega = \frac{iy-b}{iy+b}$. Using the Cauchy's formula one can show that

$$\boldsymbol{a}_k := \frac{1}{2\pi i} \int_{|\omega|=1} \frac{G(\omega; \alpha, b)}{\omega^{k+1}} d\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} G(e^{i\theta}; \alpha, b) e^{-ik\theta} d\theta, \tag{29}$$

where the change of variable $\omega = e^{i\theta}$ has been used. To approximate this integral, we follow [61], where it is suggested to use the midpoint rule instead of the trapezoidal rule because both $\theta = 0$ and $\theta = 2\pi$ would map to $\omega = 1$ in (29), which would require one to evaluate $\tilde{\boldsymbol{u}}(z)$ at infinity. The coefficients $\boldsymbol{a}_k$, $k = 0, \ldots, N_z - 1$, are therefore approximated as

$$\hat{\boldsymbol{a}}_k := \frac{1}{2N_z} \sum_{j=-N_z}^{N_z-1} G(e^{i\theta_{j+1/2}}; \alpha, b) e^{-ik\theta_{j+1/2}} = \frac{b}{N_z} \sum_{j=-N_z}^{N_z-1} \frac{e^{-ik\theta_{j+1/2}}}{1-e^{i\theta_{j+1/2}}} \tilde{\boldsymbol{u}} \left(\alpha + b \frac{1+e^{i\theta_{j+1/2}}}{1-e^{i\theta_{j+1/2}}} \right), \tag{30}$$

where we have used a midpoint discretization based on $2N_z$ intervals with $\theta_j = j\pi/N_z$. It is easy to see that, by evaluating $G(\cdot; \alpha, b)$ at $e^{i\theta_{j+1/2}}$, the frequencies at which $\tilde{\boldsymbol{u}}(\cdot)$ has to be evaluated have the following simplified expression

$$z_j := \alpha + ib \cot \frac{\theta_{j+1/2}}{2} \quad \text{for } j = -N_z, \ldots, N_z - 1. \tag{31}$$

We note that only the imaginary part varies with the discretization index, while the real part α remains fixed. Finally, the time signal, based on a N_z-term truncation of the Laguerre series (28), becomes

$$\hat{\boldsymbol{u}}(t) := e^{(\alpha-b)t} \sum_{k=0}^{N_z-1} \hat{\boldsymbol{a}}_k L_k(2bt), \tag{32}$$

where $L_k(\cdot)$ can be computed recursively using, e.g., the Clenshaw's algorithm [7].

As mentioned in Section 3.3, our goal is to recover the (discrete) time signals at sensors locations, so we replace $\tilde{\boldsymbol{u}}(z_j)$ in the definition of the Weeks coefficients (30) with $\tilde{g}_i(z_j; \boldsymbol{\mu}_m)$ defined in (27), thus obtaining the expansion coefficients

$$\hat{\boldsymbol{a}}_{k,h} := \frac{b}{N_z} \sum_{j=-N_z}^{N_z-1} \frac{e^{-ik\theta_{j+1/2}}}{1-e^{i\theta_{j+1/2}}} \tilde{g}_i(z_j; \boldsymbol{\mu}_m), \quad k = 0, \ldots, N_z - 1, \tag{33}$$

where the additional subscript h indicates that the Laplace transform is the solution of a PDE using a FE discretization. Then, by replacing $\hat{\boldsymbol{a}}_k$ with $\hat{\boldsymbol{a}}_{k,h}$ in (32), we obtain the discrete displacement vectors at point $\boldsymbol{x}_i$ and at time t_n:

$$\hat{g}_i(t_n; \boldsymbol{\mu}_m) := e^{(\alpha-b)t_n} \sum_{k=0}^{N_z-1} \hat{\boldsymbol{a}}_{k,h} L_k(2bt_n), \quad \text{for all } i = 1, \ldots, N_s \text{ and all } n = 1, \ldots, N_t. \tag{34}$$

We thus obtain the full discrete time history $\hat{\mathbf{g}}_i^m := [\hat{g}_i(t_0; \boldsymbol{\mu}_m), \ldots, \hat{g}_i(t_{N_t}; \boldsymbol{\mu}_m)]$, i.e., the Weeks approximation of the discrete time signals $\mathbf{g}_i^m$, defined in (18), for all sensors locations.

Remark 1. *(Halving the number of solutions)* By observing that $\theta_{j+1/2} = -\theta_{2N_z-(j+1/2)+1}$ for all $j = -N_z, \dots, -1$ and exploiting trigonometric identities, one can show that $z_j = \bar{z}_{(2N_z-j+1)}$ in (31), where $\bar{z}$ is the complex conjugate of z. Moreover, it is easy to prove that if $\tilde{\mathbf{u}}_h^j$ is the complex solution of (23) for $z := z_j$ then its conjugate $\overline{\tilde{\mathbf{u}}_h^j}$ is the solution of (23) for $z := \bar{z}_j$. This halves the number of times we need to solve the linear system (23) to compute $\{\tilde{g}_i(z_j;\boldsymbol{\mu}_m)\}_{j=-N_z}^{N_z-1}$ and the coefficients $\hat{\boldsymbol{a}}_{k,h}$ in (33).

Remark 2. *(The free parameters α and b)* The Weeks method contains two free parameters, $\alpha \in \mathbb{R}$ and $b \in \mathbb{R}^+$, and it has been observed that the accuracy of this algorithm depends critically on the choice of these. There exists several rules of thumb in the literature (see, e.g., [60, 45, 17]), where an estimate for α and b often requires the user to know at least the real part of the rightmost singularity of the Laplace transform α_0. In these studies, larger values of b correspond to faster convergence of the series, but at the same time a smaller value is preferable for large time intervals T. A more systematic study is presented in [18], where the authors define the optimal b for a given α and a particular class of transforms. However, to apply this we would need to determine the location of the singularities (and in particular α_0) of the solution of (23), evaluated at the sensors locations, i.e., $\tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, z_j; \boldsymbol{\mu}_m)$ defined in (22). This is challenging because this quantity is expensive to compute and thus it would be available only at few frequency locations. Moreover, it would be complex to verify that this Laplace transform fulfils the properties required to belong to the class defined in [18].

Two additional strategies to find the optimal values are proposed in [61]. While the second one requires no information of the location of the singularities, both algorithms assume t to be fixed and require as input the analytical expression of the Laplace transform. While one may overcome the first issue by observing that the optimal parameters α and b are, to a large degree, independent of t for large N_z, no alternative is known for the case in which the Laplace transform is not known analytically. Indeed, α and b are obtained by performing a minimization on a truncation error which is based on the evaluation of the Laplace transform at multiple frequency locations. When the Laplace transform is the unknown solution of a PDE, the Weeks method is ideal to retrieve the entire time signal at the cost of solving N_z times the linear system (23). Unfortunately, the solutions proposed in [61] to identify optimal values of α and b are not suitable as they would require many additional solutions of (23).

Instead, we choose these hyper-parameters using a different approach: for a fixed $\boldsymbol{\mu}^* \in \mathcal{P}$ and a fixed resolution N_z, we solve (23) for few input values in the ansatz intervals $\alpha \in [\alpha_m, \alpha_M]$ and $b \in [b_m, b_M]$. Then, using a fixed number of time steps N_t, we choose as optimal the values for which the $\|\cdot\|_2$ error between the recovered time signals and the corresponding Newmark solutions at all sensors locations is minimized, i.e.,

$$\alpha^{opt}, b^{opt} := \min_{\alpha, b} \left\| \sum_i \left(\mathbf{g}_i^* - \hat{\mathbf{g}}_i^*\right) \right\|_2^2, \tag{35}$$

where $\mathbf{g}_i^*$ and $\hat{\mathbf{g}}_i^*$ are defined in (18) and (34), respectively. We remark that only $\hat{\mathbf{g}}_i^*$ depends on the parameters α and b.

Algorithm 1 summarizes the Weeks method and how it is connected to the solution of the acoustic-elastic wave equation in the frequency domain. Clearly, the Weeks method, applied to the solutions of (23), is advantageous with respect to solving the PDE in time only if the number of frequencies, needed to generate an adequate numerical inverse Laplace transform, are significantly less than the number of time steps to generate the discrete time signal, i.e., $N_z \ll N_t$.

3.6. The Reduced Basis method

We present a reduced-order approach that significantly reduces the computational burden of repeatedly solving the parametrized problem (19) by exploiting the $\boldsymbol{\mu}$-dependence of the solution. Indeed, solving the high-fidelity complex linear system (23), or its real counterpart (24), for many input parameters is essential to construct databases and robust classifiers to detect anomalies in unseen data. Even though the translation to frequency domain described in the previous sections reduces the computational effort to generate the datasets of discrete time signals, a substantial speedup can still be achieved by applying reduced order modelling (ROM) techniques. Projection-based ROM techniques, and in particular the well-known reduced

Algorithm 1 Construction of N_s synthetic databases of time signals by solving PDE in frequency domain

procedure ConstructDatabases($\{\boldsymbol{x}_i \in \Omega\}_{i=1}^{N_s}$, $\boldsymbol{\Xi}^{N_{tr}}$, α, b, N_z, N_t)
  for $m = 1$ to N_{tr} **do**
    for $j = 0$ to $N_z - 1$ **do**
      Compute $y_j = \mathrm{Im}(z_j)$ defined in (31)
      Solve the linear system (23) for $z = \alpha + iy_j$ and $\boldsymbol{\mu}_m$
      Evaluate the solution at all the N_s sensors' locations and obtain $\tilde{g}_i(z_j; \boldsymbol{\mu}_m) \in \mathbb{C}^d$ using (27)
    Obtain the remaining $\{\tilde{g}_i(z_j; \boldsymbol{\mu}_m)\}_{j=-N_z}^{-1}$ by complex conjugation $\forall i = 0, \dots, N_s$
    for $i = 1$ to N_s **do**
      Compute the coefficients $\{\hat{\boldsymbol{a}}_{k,h} \in \mathbb{R}^d\}_{k=0}^{N_z-1}$ using (33)
      Retrieve the full time series $\hat{\mathbf{g}}_i^m \in \mathbb{R}^{d \times (N_t+1)}$ by expansion in the Laguerre's polynomials (34)
  return $\hat{\mathcal{D}}_i^{N_{tr}} = \{\hat{\mathbf{g}}_i^m\}_{m=1}^{N_{tr}}, \forall i = 1, \dots, N_s$

basis (RB) method, have been applied extensively to efficiently replace large algebraic parametric systems with much smaller ones in many-query contexts for design, real-time control, optimization or uncertainty quantification, and others. We refer the interested reader to [47, 22, 49] and the references therein for an in-depth overview of RB methods.

The main idea of RB methods is to generate an approximate solution to (20) for any choice of the parameter within the given parameter set at a cost that is independent of the cost of the original high-fidelity problem. In particular, the reduced solution $\tilde{\boldsymbol{u}}_N$ belongs to a low-dimensional subspace $\tilde{V}_N \subset \tilde{V}_h$ of dimension $N \ll N_h$. The smaller N, the cheaper it will be to solve the reduced system. To restrict the trial and test space $\tilde{V}_h$ introduced in Section 3.4, to a low-dimensional subspace $\tilde{V}_N$, we construct the *reduced basis* associated to $\tilde{V}_N$, obtained by orthonormalization of a set of high-fidelity solutions, called *snapshots*, and computed for a small set of parameter values. Then, a Galerkin projection onto this subspace is performed to construct the RB problem. The generic RB method relies on an offline-online decomposition of tasks: offline we compute the snapshots for different parameter values and use them to generate the N basis functions, while online, for a new parameter, we solve an algebraic system of dimension N, whose solution is then projected onto the original high-fidelity space by a linear combination of the precomputed basis.

We use the proper orthogonal decomposition (POD) to generate the low-dimensional subspace where the RB solution is sought. Let us generate the snapshot matrix whose columns are the high-fidelity solutions of (23), obtained for $n_s < N_h$ different values of the input frequency $z \in \mathbb{C}$ and the physical parameter $\boldsymbol{\mu} \in \mathcal{P}$:

$$\tilde{\mathbf{S}} := \left[\tilde{\mathbf{u}}_h\left(z_0; \boldsymbol{\mu}_0\right) | \dots | \tilde{\mathbf{u}}_h\left(z_{n_s-1}; \boldsymbol{\mu}_{n_s-1}\right)\right] \in \mathbb{C}^{N_h \times n_s}. \tag{36}$$

For a prescribed dimension $N \le n_s$, the POD relies on the singular value decomposition (SVD) of $\mathbf{S}$ to identify the N-dimensional subspace which best approximates the snapshots among all possible N-dimensional subspaces. Let

$$\tilde{\mathbf{S}} = \tilde{\mathbf{U}}\mathbf{\Sigma}\tilde{\mathbf{Z}}^T,$$

where $\tilde{\mathbf{U}} \in \mathbb{C}^{N_h \times N_h}$ and $\tilde{\mathbf{Z}} \in \mathbb{C}^{n_s \times n_s}$ are two orthogonal matrices and $\tilde{\mathbf{\Sigma}} = \mathrm{diag}(\sigma_1, \dots, \sigma_{n_s}) \in \mathbb{C}^{N_h \times n_s}$ with $\sigma_1 \ge \sigma_2 \ge \dots \ge \sigma_{n_s}$. The POD basis $\mathbf{V} \in \mathbb{C}^{N_h \times N}$ of dimension N is defined as the set of the first N left singular vectors of $\mathbf{U}$.

These basis minimizes the 2-norm of the projection error of the snapshot vectors (see e.g., Proposition 6.1 of [47]). However, since $\tilde{\boldsymbol{u}}_h(z; \boldsymbol{\mu}) \in \tilde{V}_h \subset \tilde{V}$, it is natural to consider the SVD with respect to a scalar product induced by the $\tilde{\mathbf{X}}_h$-norm, where $\tilde{\mathbf{X}}_h \in \mathbb{C}^{N_h \times N_h}$ is the matrix associated with the scalar product defined on $\tilde{V}_h$, i.e.,

$$\|\tilde{\boldsymbol{u}}\|^2 := m(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{u}}) + a(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{u}}; \boldsymbol{\mu}),$$

where $m(\cdot,\cdot)$ and $a(\cdot,\cdot;\boldsymbol{\mu})$ are defined in (13) for unit values of the Lamé constants (10). By considering the SVD of $\tilde{\mathbf{X}}^{1/2}\tilde{\mathbf{S}}$ we obtain a basis that is $\tilde{\mathbf{X}}_h$-orthonormal. Similarly, the POD basis can conveniently be obtained by computing the first N eigenvectors of the *correlation matrix* $\tilde{\mathbf{C}} := \tilde{\mathbf{S}}^T\tilde{\mathbf{X}}_h\tilde{\mathbf{S}}$, i.e., $\tilde{\mathbf{C}}\tilde{\boldsymbol{\psi}}_i = \sigma_i^2\tilde{\boldsymbol{\psi}}_i$. Therefore, the POD basis can also be seen as the set of vectors

$$\tilde{\boldsymbol{\zeta}}_j := \frac{1}{\sigma_j}\tilde{\mathbf{S}}\tilde{\boldsymbol{\psi}}_j, \quad j = 1, \dots, N. \tag{37}$$

In practice, the number of basis N is not chosen a priori, but for a prescribed tolerance ε_{POD}, given as the smallest integer such that

$$I(N) := \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{n_s} \sigma_i^2} \geq 1 - \varepsilon_{POD}, \tag{38}$$

i.e., the energy retained by the last $n_s - N$ modes is equal or smaller than ε_{POD}. $I(N)$, called the relative information content of the POD basis, represents the percentage of energy of the snapshots captured by the first N POD modes [47].

Given the particular setting described in Section 3.5, to recover the time-dependent signals using the Weeks method for a new parameter $\boldsymbol{\mu} \in \mathcal{P}$, we have to solve N_z reduced systems of size N. Hence, we perform a reduction not only on the parameter space $\mathcal{P}$, but also on the frequency set (31). However, as these frequencies are fixed[5], the frequency z (or equivalently its imaginary part y) does not have to be considered as an additional parameter *per se* as done in (36). Instead, by choosing the number of snapshots n_s to be a multiple of the number of frequencies N_z, we fix the snapshots to be computed for those exact frequencies that will be needed online. In practice, given $k_z \in \mathbb{R}$, we sample $n_s := k_z N_z < N_h$ parameters $\boldsymbol{\mu} \in \mathcal{P}$ and pair them with the N_z frequencies so that the snapshot matrix (36) becomes

$$\tilde{\mathbf{S}} := \left[\tilde{\mathbf{u}}_h\left(z_0;\boldsymbol{\mu}_0\right)|\ldots|\tilde{\mathbf{u}}_h\left(z_{N_z-1};\boldsymbol{\mu}_{N_z-1}\right)|\ldots|\tilde{\mathbf{u}}_h\left(z_0;\boldsymbol{\mu}_{(k_z-1)N_z}\right)|\ldots|\tilde{\mathbf{u}}_h\left(z_{N_z-1};\boldsymbol{\mu}_{k_zN_z-1}\right)\right], \tag{39}$$

where z_j are defined in (31) for $j = 0, \ldots, N_z - 1$. Provided N_z is sufficiently large to ensure that the high-fidelity time signals, retrieved with the Weeks method, are a good approximation of the high-fidelity time signals that could have been obtained with the Newmark method, N_z parameters $\boldsymbol{\mu} \in \mathcal{P}$ may not be enough to provide a good representative basis of dimension N for complex problems. When the solution in $\boldsymbol{\mu}$ is non smooth and/or $\mathcal{P}$ is too large, large values of k_z should be used. Alternatively, one could consider N_z different RB problems with N_z different snapshot matrices $\tilde{\mathbf{S}}_j \in \mathbb{C}^{N_h \times n_s}$ for $j = 0, \ldots, N_z - 1$. In this case, each frequency might be associated with a different number of basis N_j. This option is more laborious, but, at the same time, it may result in more stable approximations.

From a practical perspective, we solve (24), instead of the complex (23). Hence, the snapshot matrix (39) is rewritten as

$$\mathbf{S} := \begin{bmatrix} \tilde{\mathbf{S}}^\alpha \\ \tilde{\mathbf{S}}^y \end{bmatrix} \in \mathbb{R}^{2N_h \times k_z N_z}, \tag{40}$$

where $\tilde{\mathbf{S}}^p \in \mathbb{R}^{N_h \times k_z N_z}$ for $p = \{\alpha, y\}$ is defined as

$$\tilde{\mathbf{S}}^p := \left[\tilde{\mathbf{u}}_h^p\left(z_0;\boldsymbol{\mu}_0\right)|\ldots|\tilde{\mathbf{u}}_h^p\left(z_{N_z-1};\boldsymbol{\mu}_{N_z-1}\right)|\ldots|\tilde{\mathbf{u}}_h^p\left(z_0;\boldsymbol{\mu}_{(k_z-1)N_z}\right)|\ldots|\tilde{\mathbf{u}}_h^p\left(z_{N_z-1};\boldsymbol{\mu}_{k_zN_z-1}\right)\right], \tag{41}$$

where $\tilde{\mathbf{u}}_h^p(z_j;\boldsymbol{\mu}_i)$ for $p = \{\alpha, y\}$ is the solution of (24) for a fixed parameter $\boldsymbol{\mu}_i$ and for $z_j = \alpha + y_j$ defined in (31) for $j = 0, \ldots, N_z - 1$ and $i = 0, \ldots, k_z N_z - 1$. The correlation matrix $\mathbf{C} \in \mathbb{R}^{k_z N_z \times k_z N_z}$ is then constructed as follows

$$\mathbf{C} := \mathbf{S}^T \mathbf{X}_{2h} \mathbf{S} = \begin{bmatrix} \tilde{\mathbf{S}}^\alpha \\ \tilde{\mathbf{S}}^y \end{bmatrix}^T \begin{bmatrix} \mathbf{X}_h & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_h \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{S}}^\alpha \\ \tilde{\mathbf{S}}^y \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{S}}^{\alpha,T}, & \tilde{\mathbf{S}}^{y,T} \end{bmatrix} \begin{bmatrix} \mathbf{X}_h \tilde{\mathbf{S}}^\alpha \\ \mathbf{X}_h \tilde{\mathbf{S}}^y \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{S}}^{\alpha,T} \mathbf{X}_h \tilde{\mathbf{S}}^\alpha + \tilde{\mathbf{S}}^{y,T} \mathbf{X}_h \tilde{\mathbf{S}}^y \end{bmatrix},$$

where $\mathbf{X}_{2h} \in \mathbb{R}^{2N_h \times 2N_h}$ is the symmetric positive definite matrix associated with the scalar product in the real space V_h of dimension $2N_h$. $\mathbf{X}_{2h}$ is a block diagonal matrix with two equal blocks $\mathbf{X}_h$, where $\mathbf{X}_h \in \mathbb{R}^{N_h \times N_h}$. We solve the eigenvalue problem

$$\mathbf{C}\boldsymbol{\psi}_i = \sigma_i^2 \boldsymbol{\psi}_i, \quad i = 1, \ldots, k_z N_z \tag{42}$$

and construct the POD basis as the set of $2N_h$-dimensional vectors (37) by replacing $\tilde{\mathbf{S}}$ with $\mathbf{S}$ and $\tilde{\boldsymbol{\psi}}_i$ with $\boldsymbol{\psi}_i$. Let $\mathbf{V} := \left[\mathbf{V}^{\alpha,T}, \mathbf{V}^{y,T}\right]^T \in \mathbb{R}^{2N_h \times N}$ be the so-defined POD basis with $\mathbf{V}^\alpha, \mathbf{V}^y \in \mathbb{R}^{N_h \times N}$. Then, the

[5] Indeed, the frequencies z_j only depend on the parameters α and b, which are fixed (see Remark 2) and the number of frequencies N_z, which can be chosen to be the same offline and online.

reduced algebraic problem (24) becomes

$$\mathbf{K}_N \begin{bmatrix} \tilde{\mathbf{u}}_N^{\alpha}(z;\boldsymbol{\mu}) \\ \tilde{\mathbf{u}}_N^{y}(z;\boldsymbol{\mu}) \end{bmatrix} = \mathbf{q}_N,$$

where

$$\begin{aligned} \mathbf{K}_N \coloneqq& \mathbf{V}^T\tilde{\mathbf{K}}(\boldsymbol{\mu})\mathbf{V} = \begin{bmatrix} \mathbf{V}^{\alpha} \\ \mathbf{V}^{y} \end{bmatrix}^T \begin{bmatrix} \mathbf{K}^{\alpha}(\boldsymbol{\mu}) & -\mathbf{K}^{y}(\boldsymbol{\mu}) \\ \mathbf{K}^{y}(\boldsymbol{\mu}) & \mathbf{K}^{\alpha}(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \mathbf{V}^{\alpha} \\ \mathbf{V}^{y} \end{bmatrix} \\ =& \mathbf{V}^{\alpha T}\left(\mathbf{K}^{\alpha}(\boldsymbol{\mu})\mathbf{V}^{\alpha} - \mathbf{K}^{y}(\boldsymbol{\mu})\mathbf{V}^{y}\right) + \mathbf{V}^{yT}\left(\mathbf{K}^{y}(\boldsymbol{\mu})\mathbf{V}^{\alpha} + \mathbf{K}^{\alpha}(\boldsymbol{\mu})\mathbf{V}^{y}\right), \\ \mathbf{q}_N \coloneqq& \mathbf{V}^T \begin{bmatrix} \tilde{\mathbf{q}}^{\alpha}(z;\boldsymbol{\mu}) \\ \tilde{\mathbf{q}}^{y}(z;\boldsymbol{\mu}) \end{bmatrix}, \end{aligned}$$

where $\tilde{\mathbf{K}}(\boldsymbol{\mu})$ is the matrix on the left-hand-side of (24) and $\tilde{\mathbf{q}}^{\alpha}(z;\boldsymbol{\mu})$, and $\tilde{\mathbf{q}}^{y}(z;\boldsymbol{\mu})$ are defined in (25). We notice that the reduced matrix $\mathbf{K}_N$ fails to preserve the structure of the high-fidelity matrix $\tilde{\mathbf{K}}(\boldsymbol{\mu})$, which causes the reduced solutions to be unstable. To overcome this loss of structure, we resort to a proper symplectic decomposition (PSD) with a symplectic Galerkin projection, and apply the *cotangent-lift* method introduced in [44], where the snapshot matrix (40) is considered in extended form, i.e.,

$$\mathbf{S}^{cl} \coloneqq \left[\tilde{\mathbf{S}}^{\alpha}, \tilde{\mathbf{S}}^{y}\right] \in \mathbb{R}^{N_h \times 2k_z N_z},$$

where $\tilde{\mathbf{S}}^{\alpha}$ and $\tilde{\mathbf{S}}^{y}$ are defined in (41). The corresponding correlation matrix becomes

$$\mathbf{C}^{cl} \coloneqq \left[\tilde{\mathbf{S}}^{\alpha}, \tilde{\mathbf{S}}^{y}\right]^T \mathbf{X}_h \left[\tilde{\mathbf{S}}^{\alpha}, \tilde{\mathbf{S}}^{y}\right] = \begin{bmatrix} \tilde{\mathbf{S}}^{\alpha,T}\mathbf{X}_h\tilde{\mathbf{S}}^{y} & \tilde{\mathbf{S}}^{\alpha,T}\mathbf{X}_h\tilde{\mathbf{S}}^{y} \\ \tilde{\mathbf{S}}^{y,T}\mathbf{X}_h\tilde{\mathbf{S}}^{\alpha} & \tilde{\mathbf{S}}^{y,T}\mathbf{X}_h\tilde{\mathbf{S}}^{y} \end{bmatrix}.$$

Then, as before, we solve (42) by replacing $\mathbf{C}$ with $\mathbf{C}^{cl}$ and, for any $N \le k_z N_z$, the POD basis $\boldsymbol{\Phi} = [\boldsymbol{\zeta}_1^{cl} \,|\ldots|\, \boldsymbol{\zeta}_N^{cl}] \in \mathbb{R}^{N_h \times N}$ of dimension N is defined, similarly to (37), as the set of N_h-dimensional vectors

$$\boldsymbol{\zeta}_i^{cl} \coloneqq \frac{1}{\sigma_i}\mathbf{S}^{cl}\boldsymbol{\psi}_i^{cl}, \quad i = 1,\ldots,N.$$

Finally, the symplectic basis is constructed as

$$\mathbf{V}^{cl} = \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi} \end{bmatrix} \in \mathbb{R}^{2N_h \times 2N}. \tag{43}$$

We observe that, by construction, $\boldsymbol{\Phi}^T\mathbf{X}_h\boldsymbol{\Phi} = \mathbf{I}_N$. Therefore, $\mathbf{V}^{cl}$ is $\mathbf{X}_h$-orthonormal, i.e., $\mathbf{V}^{cl,T}\mathbf{X}_{2h}\mathbf{V}^{cl} = \mathbf{I}_{2N}$. With this particular choice of basis, the structure of the system is preserved and the reduced solutions are stable. In particular, for a new parameter $\boldsymbol{\mu}$ we need to solve the following reduced system of dimension $2N$:

$$\mathbf{K}_N^{cl} \begin{bmatrix} \tilde{\mathbf{u}}_N^{\alpha}(z;\boldsymbol{\mu}) \\ \tilde{\mathbf{u}}_N^{y}(z;\boldsymbol{\mu}) \end{bmatrix} = \mathbf{q}_N^{cl}, \tag{44}$$

where

$$\begin{aligned} \mathbf{K}_N^{cl} \coloneqq& \mathbf{V}^{cl,T}\tilde{\mathbf{K}}(\boldsymbol{\mu})\mathbf{V}^{cl} = \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi} \end{bmatrix}^T \begin{bmatrix} \mathbf{K}^{\alpha}(\boldsymbol{\mu}) & -\mathbf{K}^{y}(\boldsymbol{\mu}) \\ \mathbf{K}^{y}(\boldsymbol{\mu}) & \mathbf{K}^{\alpha}(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^T\mathbf{K}^{\alpha}(\boldsymbol{\mu})\boldsymbol{\Phi} & -\boldsymbol{\Phi}^T\mathbf{K}^{y}(\boldsymbol{\mu})\boldsymbol{\Phi} \\ \boldsymbol{\Phi}^T\mathbf{K}^{y}(\boldsymbol{\mu})\boldsymbol{\Phi} & \boldsymbol{\Phi}^T\mathbf{K}^{\alpha}(\boldsymbol{\mu})\boldsymbol{\Phi} \end{bmatrix}, \\ \mathbf{q}_N^{cl} \coloneqq& \mathbf{V}^{cl,T} \begin{bmatrix} \tilde{\mathbf{q}}^{\alpha}(z;\boldsymbol{\mu}) \\ \tilde{\mathbf{q}}^{y}(z;\boldsymbol{\mu}) \end{bmatrix}. \end{aligned}$$

Algorithm 2 summarizes the cotangent lift method to construct a symplectic RB basis.
Algorithm 1 can be updated to include the RB approach by simply modifying lines 5 and 6, provided that the symplectic basis (43) is previously constructed. In line 5 we need to solve the reduced linear system (44) instead of (24) and in line 6 the output of interests $\tilde{g}_i(z_j;\boldsymbol{\mu}_m)$ are obtained by evaluating the real and imaginary part of the solution separately, i.e.,

$$\tilde{g}_i(z_j;\boldsymbol{\mu}_m) = \sum_{j=1}^{N} \tilde{u}_j^{\alpha}(z;\boldsymbol{\mu})\tilde{\boldsymbol{\zeta}}_j^{cl}(\boldsymbol{x}) + i\sum_{j=1}^{N} \tilde{u}_j^{y}(z;\boldsymbol{\mu})\tilde{\boldsymbol{\zeta}}_j^{cl}(\boldsymbol{x}), \quad j = 1\ldots,N,$$

Algorithm 2 Construct a symplectic basis using on the cotangent lift method

procedure ConstructRB($\tilde{\mathbf{S}}^\alpha, \tilde{\mathbf{S}}^y, \mathbf{X}_h, k_z, N_z, \varepsilon_{POD}$)
  Form the snapshot matrix $\mathbf{S}^{cl} := [\tilde{\mathbf{S}}^\alpha, \tilde{\mathbf{S}}^y]$
  Form the correlation matrix $\mathbf{C}^{cl} := (\mathbf{S}^{cl})^T \mathbf{X}_h \mathbf{S}^{cl}$
  Solve the eigenvalue problem $\mathbf{C}^{cl} \boldsymbol{\psi}_i^{cl} = \sigma_i^2 \boldsymbol{\psi}_i^{cl},\ i = 1 \ldots, k_z N_z$
  Set $\boldsymbol{\zeta}_i^{cl} := \frac{1}{\sigma_i} \mathbf{S}^{cl} \boldsymbol{\psi}_i^{cl}, i = 1, \ldots, N$ where N is the minimum integer that satisfies (38)
  Set $\boldsymbol{\Phi} = [\boldsymbol{\zeta}_1^{cl} \,|\, \ldots \,|\, \boldsymbol{\zeta}_N^{cl}]$
  return $\mathbf{V}^{cl}$, defined in (43)

where $\tilde{u}_j^\alpha(z; \boldsymbol{\mu})$ is the j-th entry and $\tilde{u}_j^y(z; \boldsymbol{\mu})$ is the $j + N$-th of the solution of the linear system (44), respectively. Finally, we note that both the offline and the online phases of the RB method belong to the database construction phase, which corresponds to one of the offline steps of the anomaly-detection process (see Figure 1).

4. The one-class classification problem

Anomaly (or novelty) detection indicates the task of identifying substantial differences in the test dataset when compared to the data available during training [46]. Such method is applied to contexts where there is an abundance of "normal" (or positive) examples and abnormal examples (or negative) are scarce or non-existent. Intrusions in electronic security systems, video surveillance, medical diagnostic problems, industrial or structural faults and failure detection are examples of some of the applications involving unbalanced training datasets. The scarcity of anomalous data can be explained by three principal reasons: (i) occurrence of abnormal events is not expected or difficult to model, (ii) even if such examples are available for training, it is difficult to cover every possible abnormal event, and (iii) acquisition of abnormal events is costly [11]. Since our training dataset is a simulated one, the last two reasons motivate us to opt for a one-class classification approach instead of a supervised one.

The anomaly detection problem can be treated as a one-class classification task by considering the semi-supervised counterpart of several classical supervised machine learning algorithms. These methods learn a description of the healthy training data offline and detect if a previously unseen object reflects this description by means of an online novelty score. Among many possibilities (see e.g., the reported summaries in [11, 46, 19, 2]) we highlight three well-known strategies: the Isolation Forest [33], based on the principles of the Random Forest method, the Local Outlier Factor [6], a nearest-neighbors based approach, and the One Class Support Vector Machine (OC-SVM) [52, 8], with details given in Section 4.1. Motivated by the use of the latter one in several SHM-related studies (see e.g., [36, 10, 3]), we rely on this for our approach.

We also mention autoencoders, a particular type of neural networks, trained to attempt to copy their inputs to their outputs, which have gained particular notoriety in the framework of anomaly detection (see e.g., [26, 38, 41]). By the combination of two networks, called encoder and decoder, an autoencoder learns the underling salient features, which are sufficient to describe and reconstruct the input. In doing so, the autoencoder exploits the idea that the training data (positive examples) concentrate around a low-dimensional manifold, learned by redundancy compression. Then, the reconstruction error, i.e., the norm of the difference between a new datum and its reconstruction, is used as a novelty score under the assumption that positive instances are expected to be reconstructed accurately, while negative instances, i.e., abnormal data, are not. The main advantage of using a reconstruction-based anomaly detection approach like the autoencoders lies in the fact that specific engineering-based, damage indicator features do not need to be specified, different from others one-class methods mentioned above. We refer the interested readers to Chapter 14 of [20] and references therein for an overview on autoencoders.

4.1. One Class SVM

The One Class SVM method is derived as a simple modification of the well-known supervised SVM method [8], used in several SHM applications (see e.g., [24, 54]). Binary classification SVMs are successful learning techniques that, given two-class input data, map them into a high dimensional, non-linear feature space where it is possible to construct a linear separation boundary, i.e., a hyperplane [58]. Given X, the set of the input training data, and F, the feature space of dimension greater than X, the idea behind this method is known as the *kernel trick*, i.e., the transformation function $\Phi : X \to F$ is not computed explicitly. Instead it is defined by a kernel to project the data into a higher dimensional space. The simple evaluation of this kernel gives the dot product in the feature map

$$k(x, y) := \Phi(x) \cdot \Phi(y). \tag{45}$$

A common choice is the Gaussian kernel

$$K(x, y) := \exp\left\{-\frac{\|x - y\|^2}{\hat{\sigma}^2}\right\}, \tag{46}$$

where $\hat{\sigma} \in \mathbb{R}$ is a free parameter.

OC-SVMs, introduced in [52, 51], apply the same binary technique to find the optimal hyperplane that separates all the healthy training data from the origin with maximum margin. The origin (in feature space) is used as a proxy for the unrepresented anomalous data.

Let $\mathcal{F} : \mathbb{R}^{d\times(N_t+1)} \to \mathbb{R}^Q$ be a function which extracts Q damage-indicator features from N_{tr} recovered signals (34) and let $\mathcal{F}_i^{N_{tr}} = [\mathcal{F}(\hat{\mathbf{g}}_i^1), \ldots, \mathcal{F}(\hat{\mathbf{g}}_i^{N_{tr}})] \in \mathbb{R}^{N_{tr}\times Q}$ be the so obtained training database of feature-valued signals at location $\boldsymbol{x}_i$, which will be defined in (52). The hyperplane, described by the parameters $w_i \in F$ and the bias $\hat{b}_i \in \mathbb{R}$, is obtained by the minimization problem

$$\min_{w_i,\, b_i,\, \xi_m} \frac{\|w_i\|^2}{2} + \frac{1}{\hat{\nu} N_{tr}} \sum_{m=1}^{N_{tr}} \xi_m - \hat{b}_i$$
$$\text{subject to: } w_i \cdot \phi\left(\mathcal{F}(\hat{\mathbf{g}}_i^m)\right) \geq \hat{b}_i - \xi_m, \quad \xi_m \geq 0, \quad \text{for } m = 1, \ldots, N_{tr}$$

where $\xi_m \in \mathbb{R}, m = 1, \ldots, N_{tr}$ are non-zero slack variables that allow soft margins, i.e., large values of ξ_j allow the m-th data point to lie on the wrong side of the decision boundary. The tradeoff between the number of misclassified training examples and the smoothness of the margin, identified by w_i, is controlled by the regularization parameter $\hat{\nu} \in]0, 1]$. Given the separating hyperplane

$$p_i^{N_{tr}}(\boldsymbol{x}) := w_i \cdot \Phi(\boldsymbol{x}) - \hat{b}_i, \tag{47}$$

the OC-SVM algorithm returns a function $f_i^{N_{tr}} : \mathbb{R}^Q \to \{-1, 1\}$ that, for each sensor, evaluates every new data point to determine on which side of the hyperplane it falls in features space. Hence, the decision function

$$f_i^{N_{tr}}(\boldsymbol{x}) := \operatorname{sgn}\left(p_i(\boldsymbol{x})\right) \tag{48}$$

will take values $+1$ for most of the training samples. The problem can be transformed to a dual form using Lagrangian multipliers and the kernel trick (45) as

$$\min_{\boldsymbol{\alpha}} \sum_{m,n=1}^{N_{tr}} \alpha_m \alpha_n k\left(\mathcal{F}(\hat{\mathbf{g}}_i^m), \mathcal{F}(\hat{\mathbf{g}}_i^n)\right)$$
$$\text{subject to: } 0 \leq \alpha_m \leq \frac{1}{\hat{\nu} N_{tr}}, \forall m = 1, \ldots, N_{tr} \quad \text{and} \quad \sum_{m=1}^{N_{tr}} \alpha_m = 1,$$

where the non-zero α_m are the support vectors (SVs). The latter are required to evaluate any new datum using the SV expansion of the hyperplane (47), which becomes

$$p_i^{N_{tr}}(\boldsymbol{x}) = \sum_{m=1}^{N_{tr}} \alpha_m k\left(\boldsymbol{x}, \mathcal{F}(\hat{\mathbf{g}}_i^m)\right) - \hat{b}_i. \tag{49}$$

With this expression, it can be proven that $\hat{\nu}$ is an upper bound on the fraction of outliers, i.e., misclassified training samples, and a lower bound on the fraction of SVs [52]. A smaller value of $\hat{\nu}$ implies fewer SVs and therefore a smooth, crude decision boundary, while a larger value of $\hat{\nu}$ leads to more SVs and therefore to a curvy decision boundary. The optimal value of $\hat{\nu}$ should be large enough to capture the data distribution and small enough to avoid overfitting. In our experiments we choose $\hat{\nu} := 0.65$.

As mentioned in [2, 29], a continuous outlier score reveals more information than a simple binary label as the output (48). Indeed, the absolute value of (49) gives information on the distance of the point $\boldsymbol{x}$ from the hyperplane: larger values are farther away from the hyperplane. Larger negative values are not only associated with more severe damages, but also with a greater confidence on the binary output (48). The choice of using (48), or another anomaly score based on (49), as decision strategy depends on the importance given to misclassification errors, i.e., false negative and false positive predictions. The formers, also called false alarms, arise when a healthy structure is classified as damaged and false positive predictions when damaged structures are classified as healthy. Ideally, one would like to keep both rates low, but in practice one of the two will be more frequent. This choice translates in the relative position of the hyperplane: moving the hyperplane towards the origin (in feature space) will increase the false positive rate and, vice-versa, moving the hyperplane towards the training set will increase the number of false negative test data. A relative approach is applied here to compute the anomaly score, i.e., we follow the strategy presented in [2], where, given $\hat{p}_i$ the maximum distance between the training data and the decision boundary for the i-th sensor, the score (49) is scaled as

$$f_i^{N_{tr}}(\boldsymbol{x}) := \frac{\hat{p}_i - p_i^{N_{tr}}(\boldsymbol{x})}{\hat{p}_i}. \tag{50}$$

Therefore, the points classified as outliers outliers are identified with scores greater than 1.

Finally, a large amount of experiments have demonstrated that the choice of the free parameter $\hat{\sigma}$ in (46) may severely impact the generalization performance of OC-SVMs. Indeed, an inappropriate choice of $\hat{\sigma}$ may lead to overfitting (small values) or under-fitting (large values). In semi-supervised or unsupervised frameworks, this hyper-parameter can not be estimated using classical strategies for model parameters selection, such as cross validation. Indeed, since only positive examples exist in the training set, it is impossible to estimate the misclassification error of the OC-SVM model. In the past decades, several strategies have been proposed to overcome this issue: for example a training error based approach in [57], a geometry based approach in [28], a tightness detection strategy, based on the spatial locations of the interior and edge samples [62, 3] and an approach based on the Fisher linear discrimination [59]. The first three strategies are observed to be equivalently succesful to detect various damage scenarios on a laboratory structure in [36]. The authors also report that the least computationally expensive method, which does not require repeated training, is the geometric approach where $\hat{\sigma}$ is chosen based on the the maximum distance between the two least similar training points [28]. This strategy is used also in this work, where the Kernel factor becomes

$$\hat{\sigma}_i^2 := \frac{\hat{d}_i}{\sqrt{-\ln \delta}}, \quad \text{where } \delta := \frac{1}{N_{tr}(1-\hat{\nu})+1},$$

where $\hat{d}_i$ is the Euclidean distance between the two least similar training points for the i-th dataset.

4.2. Feature Extraction

The displacement time series at each sensor location, $\hat{\mathbf{g}}_i^m = [\hat{\mathbf{g}}_i^m(t_0), \dots, \hat{\mathbf{g}}_i^m(t_{N_t})] \in \mathbb{R}^{d\times(N_t+1)}$, for $m = 1, \dots, N_{tr}$, acquired using Algorithm 1, including the appropriate modifications to leverage the RB framework described in Section 3.6, need to be pre-processed before being used to train the one-class classifiers. The ideal features for a robust structural damage detection and localization system should be sensitive to the presence of damage, but insensitive to the operational and environmental variability in a normal range [16]. Common choices for the damage-sensitive features can be found for example in [36, 34].

In this work, the raw displacement signals are processed into a Q-dimensional feature vectors with $Q := 6d$. We consider the following characteristic values: the $d-$dimensional crest factor, which indicates how extreme the peaks are in a waveform, the maximum and minimum values of the d-dimensional response, the corresponding arrival times, i.e., the onset, and the number of peaks and valleys in the signals. Indeed, it has already been observed (see e.g., [65, 34]) that, in the presence of a crack, which acts as an obstacle dissipating some of the energy carried by the transmitted waves, the signal becomes more attenuated and the time of arrival becomes longer because of the extra distance between the source and the sensor due to the discontinuity of the material.

For each sample $\hat{\mathbf{g}}_i^m$, the crest factor $\mathbf{C}_i^m \in \mathbb{R}^d$ is defined as

$$\mathbf{C}_i^m := \frac{|\hat{\mathbf{g}}_i^m|_{peak}}{(\hat{\mathbf{g}}_i^m)_{rms}}, \qquad \text{where} \quad \begin{cases} |\hat{\mathbf{g}}_i^m|_{peak} := & \max_n |\hat{\mathbf{g}}_i^m(t_n)| \\ (\hat{\mathbf{g}}_i^m)_{rms} := & \sqrt{\frac{1}{N_t+1}\sum_{n=0}^{N_t} \left(\hat{\mathbf{g}}_i^m(t_n)\right)^2} \end{cases}. \tag{51}$$

The arrival time $\mathbf{A}_i^m \in \mathbb{R}^d$, the number of peaks $\mathbf{P}_i^m \in \mathbb{R}^d$ and valleys $\mathbf{V}_i^m \in \mathbb{R}^d$ are defined using the `peakfinder` Matlab function [39]. Precisely, $\mathbf{A}_i^m \in \mathbb{R}^d$ is defined as the time step corresponding to the first peak or valley. The two hyper-parameters of the `peakfinder` function, i.e., `sel` and `thresh`, are defined as a percentage of the maximum amplitude of 30 randomly chosen healthy training signals for the first $N_t = 20'000$ steps, sensor by sensor and component by component. In particular we choose `sel`, which gives information on the peak value, relative to surrounding data, to be identified as the 3% or 7% of the maximum amplitude of the healthy signals, for the 2D and 3D problems, respectively. The threshold `thresh`, i.e., the value for which peaks must exceed to be a maxima or a minima, is fixed to 5.5% or 9% of the maximum amplitude, for the 2D and 3D problems, respectively. These values are chosen experimentally by visually inspecting the position of the onset values over a set of signals. We note that, for the 3D problem, using higher percentages of the maximum amplitude of the healthy signals leads to a choice of these hyper-parameters, which can better distinguish between the effective signal arrival and spurious oscillations. Moreover, we observe that the classification results obtained using `peakfinder` are more robust and less prone to be affected by artefacts generated by the numerical inverse Laplace transform reconstruction with respect to finding the onset based only on a sensor-dependent threshold ε_i of the signal values, i.e., $\mathbf{A}_i^m := \arg\min_n \{|\hat{\mathbf{g}}_i^m(t_n)| \geq \varepsilon_i\}$.

Therefore, for all $i = 1, \ldots, Ns$, the feature-based database becomes

$$\mathcal{F}_i^{N_{tr}} := \left[\mathcal{F}(\hat{\mathbf{g}}_i^1), \ldots, \mathcal{F}(\hat{\mathbf{g}}_i^{N_{tr}})\right], \quad \text{where } \mathcal{F}(\hat{\mathbf{g}}_i^m) := \begin{bmatrix} \mathbf{C}_i^m \\ \mathbf{A}_i^m \\ \mathbf{P}_i^m \\ \mathbf{V}_i^m \\ \max_n \hat{\mathbf{g}}_i^m(t_n) \\ \min_n \hat{\mathbf{g}}_i^m(t_n) \end{bmatrix} \quad \text{for } m = 1, \ldots, N_{tr}. \tag{52}$$

We observe that, features extracted directly from the raw signals in frequency domain, i.e., before applying the Weeks method for reconstruction, are not considered here. Nevertheless, such features (e.g., the transmissibility defined for example in [36]), could be also included either by direct extraction for simulated samples or by pre-applying a Laplace transform for experimental sensor signals, which are available only in time domain.

4.3. Dimensionality reduction

Among the Q selected features, dimensionality reduction is needed to generate robust classifiers. Indeed, we observe that the OC-SVM strategy does not capture anomalies well if applied directly to the feature-based datasets (52). It has been shown (see e.g., [56]) that using too many features may introduce too much noise in the dataset and leadi to overfitting. In general, classic feature selection strategies do not guarantee the best classification performances when applied to highly unbalanced training datasets, i.e., retaining only the high-variance directions may not provide informative results on the features that are most sensitive to damage. Even though there exist several studies (see e.g., [40, 27]) in which the information carried by low-variance directions is emphasized, in many cases removing redundant features by projecting the data on the high-variance directions remains beneficial. Principal component analysis (PCA) and random projections (RP) are two widely used compression methods. While for very large datasets RP are known to achieve best performances (see e.g., [1]), given our choice of relatively few features, i.e., $Q := 6d$, PCA transformation is more appropriate.

In practice, we first normalize the training data so that each feature has zero mean and unit standard deviation among the training samples. We remark that the scaling required to achieve this transformation is then applied to the test dataset before class prediction. Then, we apply the PCA and store the principal coefficients $P_{PCA} \in \mathbb{R}^{Q \times k_{PCA}}$. In this work, for all sensors we observe a rapid decay of the PCA eigenvalues, which motivates our choice of retaining only 1 principal component, i.e., $k_{PCA} = 1$. Finally, we apply the OC-SVM approach to $\mathcal{F}_i^{N_{tr}} P_{PCA}$ for all $i = 1, \ldots, N_s$. The same transformation is applied to the test datasets.

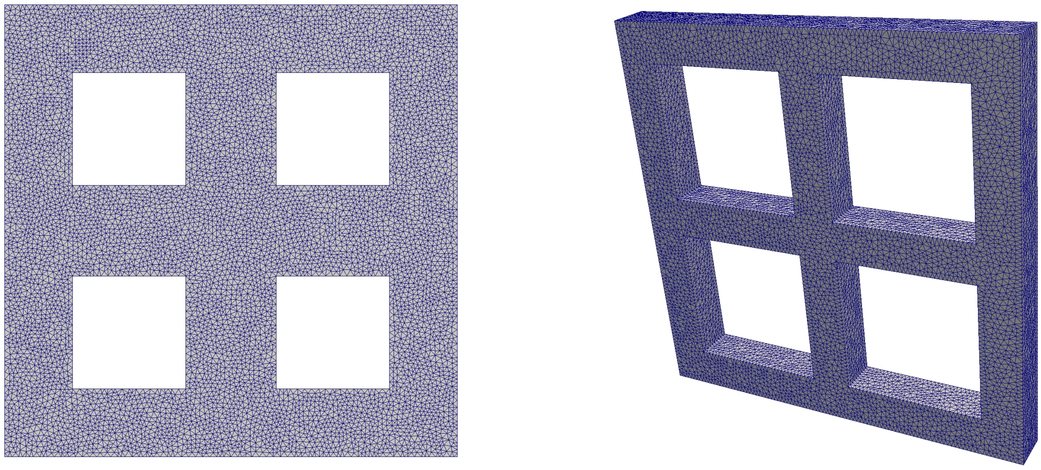

Figure 2: Healthy meshes for the 2D (*left*) and 3D (*right*) problems. The former represents the section of a simplified beam and the latter is obtained by extruding a similar 2D domain, but with larger holes, along the third direction. The 2D problem has normalized dimensions 1×1, while the 3D one has dimensions $1 \times 1 \times 0.1$.

4.4. Hierarchical classification

Training separate models for each sensor allows for both detection and localization of damages. We identify three levels of damage identification. First of all, a structure is considered damaged if at least one sensor is classified as an outlier, i.e., the anomaly score (50) is greater than 1. Secondly, as the anomaly score is a continuous value, one can additionally deduce information about the severity of the damage, distinguishing between *strong outliers*, i.e., values much bigger than 1, and *mild outliers*, i.e., values slightly above unity. Indeed, if a structure presents many strong outliers, we expect a severe damage. Finally, damage localization is achieved by observing that damage is expected to be closer to those sensors which are classified as outliers.

5. Numerical results

In this section we first present the geometrical domain with its sensors and source definition, the values and distribution of the input parameters and the parameters used for the numerical inverse Laplace transform reconstruction. Then, we describe the construction of the training and test datasets for both the 2D and 3D problems and highlight the classification results. In our experiments, FEniCS [35] is employed for the implementation of the high fidelity solver, while the open source library RBniCS [4, 22], that implements several reduced order modeling techniques, is used to implement the reduced basis solver. The numerical inverse Laplace reconstruction is implemented with ad hoc Python functions, while the feature extraction, dimensionality reduction, and classification steps are carried out in Matlab [39], employing, in particular, the built-in functions `peakfinder`, `pca`, and `fitcsvm`.

The mesh for the healthy domain $\Omega \subset \mathbb{R}^d$ is reproduced in Figure 2 for $d = 2, 3$. The domain is discretized using tetrahedral cells; a FE approximation by $\mathbb{P}_1$ elements is used, resulting in 30'912 and 217'344 DOFs for $d = 2, 3$, respectively. We remark that, since we solve (24), half of the DOFs represent the real part and the other half the imaginary part of the $d-$dimensional solution. Indeed, the number of DOFs required to solve the same problem in time domain (15) is halved, provided the same mesh is used.

5.1. The parameter space

In the following numerical experiments we use the homogenous free-slip boundary conditions (11), i.e., $\boldsymbol{g}_N = \mathbf{0}$, and we choose the density and damping coefficients as $\rho := 1$ and $\eta := 0.1$, respectively. All the other parameters are defined below and in the following subsections.

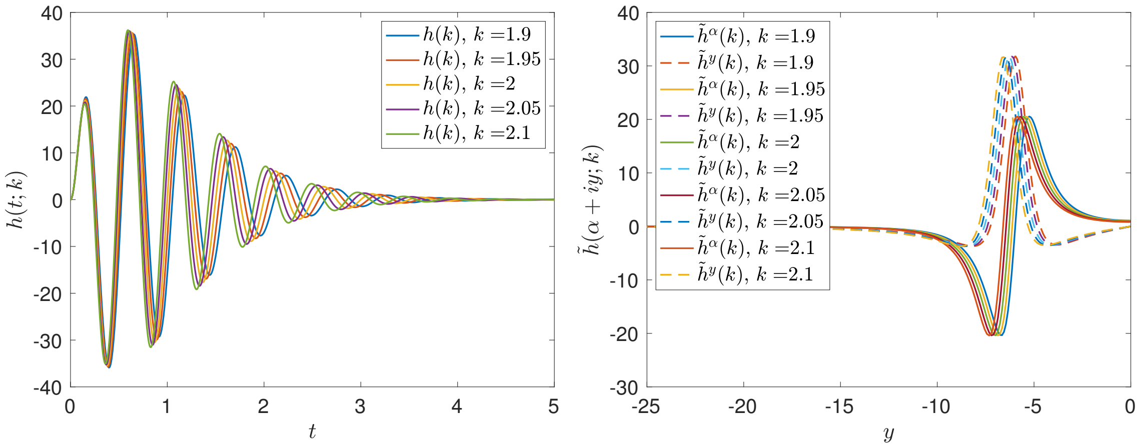

Figure 3: Source function for various values of the parameter k. The source function $h(t;k)$ is plotted as a function of time (*left*) and its corresponding Laplace transform, split in its real (*full lines*) and imaginary (*dashed lines*) components, is plotted as a function of y, i.e., the imaginary part of frequency z for a fixed α value (*right*).

Aiming at representing the different environmental and operational conditions, necessary to make reliable damage predictions, we choose three parameters of variation, i.e., $\boldsymbol{\mu} := [E, \nu, k] \in \mathcal{P} \subset \mathbb{R}^p$ with $p = 3$. E is the Young's Modulus, ν the Poisson's ratio which determines the Lamé constants (10) and k is a parameter of the source function $h(t; \boldsymbol{\mu})$ (or equivalently $\tilde{h}(z; \boldsymbol{\mu})$), whose expression is defined in the following section. In the generation of the dataset, the parameter set (3) is based on uniform random samples. We choose

$$\Xi^{N_{tr}} := \{E_m, \nu_m, k_m\}_{m=1}^{N_{tr}} \in \mathcal{P}, \quad \text{with } \mathcal{P} := [0.999, 1.001] \times [0.329, 0.331] \times [1.9, 2.1]. \tag{53}$$

A more realistic parameter space could be provided by relying on model calibration, based on the combination of experimental data with prior knowledge. However, this goes beyond the scope of this paper.

5.2. The source term and the sensors locations

The excitation of the structure is necessary to generate waveforms which propagate in the structure and are measured at sensors for signal diagnostic. In this work, we consider active sources, as an alternative to passive continuous sources such as wind or tides. In several vibration-based non-destructive evaluation tests, electromechanical shakers are used to inject pure white Gaussian noise (see e.g., [36, 43]). Alternatively, sources based on sinusoidal waves are also used (see e.g., [54, 65]), which we also focus on. Moreover, in the SHM framework, short pulse impulses are often used for non-destructive evaluation and testing (see e.g., the more sophisticated Hanning-windowed sinusoidal tone-bursts used in [65]) in combination with the damage-sensitive features described in Subsection 4.2. In particular, it is observed that damaged structures produce greater attenuation for signals with higher frequency, i.e., signals with higher frequency are more sensitive to the presence of damage sites as explained in [12, 32].

In this work, the source functions $s(\boldsymbol{x}; \boldsymbol{\mu})$ and $h(t; \boldsymbol{\mu})$ of (19) are chosen as

$$s(\boldsymbol{x}; \boldsymbol{\mu}) := \frac{\exp\left\{-\sum_{i=1}^{d} \frac{(\boldsymbol{x}_i - \bar{\mu}_i)^2}{2\bar{\sigma}_i^2}\right\}}{2\pi\bar{\sigma}^d}, \quad h(t; \boldsymbol{\mu}) := k_s \sin(k\pi t)\, te^{-t},$$

where $\bar{\sigma} := 0.01$ represents the width of a Gaussian centered at $\bar{\boldsymbol{\mu}} := [0.55, 0.125]$ and $\bar{\boldsymbol{\mu}} := [0.51, 0.06, 0]$ in 2D and 3D, respectively. Since these values are fixed for all numerical examples, the space source function is independent of the parameter $\boldsymbol{\mu}$. For the time-dependent source function, we choose the scaling factor $k_s := 100$, such that h only depends on one parameter, k, which controls the number of cycles before attenuation. Moreover, our choice guarantees $\frac{\partial h(t;\boldsymbol{\mu})}{\partial t}\big|_{t=0} = 0$, which provides a solution that is coherent with the homogenous initial conditions, i.e., $\boldsymbol{u}_0 = \boldsymbol{v}_0 = 0$. The corresponding Laplace transform of $\tilde{h}$ is

$$\tilde{h}(z; k) = k_s \frac{2\pi k(z+1)}{(\pi^2 k^2 + (z+1)^2)^2}. \tag{54}$$

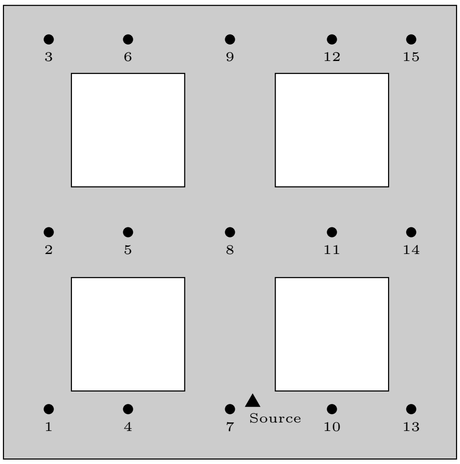

Figure 4: Sketch of sensors numbering system and source placement for the 2D problem. Numbered filled circles represent the 15 sensor locations, while the triangle represents the source position $\bar{\boldsymbol{\mu}} = [0.55, 0.125]$.

Given $z := \alpha + iy$, (54) can be split in its real and imaginary parts, i.e., $\tilde{h}^{\alpha}(\alpha + iy; k)$ and $\tilde{h}^{y}(\alpha + iy; k)$, required in (25). Figure 3 shows the source function in time and frequency domain when the real part of the frequency z is fixed, i.e., $\alpha = 0.26$, and for different values of $k \in [1.9, 2.1]$.

We consider a total of $N_s = 15$ sensors for the 2D problem and $N_s = 46$ for the 3D problem. For the 2D model, the sensor locations $\boldsymbol{x} := (x_i, y_j)$, sketched in Figure 4, are obtained by all combinations i, j, where $x_i \in [0.1, 0.275, 0.5, 0.725, 0.9], y_j \in [0.11, 0.5, 0.925]$. In 3D, for practical engineering purposes, sensors embedded in the structure are excluded and the sensors location are restricted to the model surface, i.e., $\boldsymbol{x} := (x_i, y_j, z_k)$, represented in Figure 5, is given by all combinations i, j, k, where $x_i \in [0, 0.1, 0.275, 0.5, 0.725, 0.9, 1], y_j \in [0, 0.075, 0.5, 0.925, 1]$ and $z_k \in [0, 0.5, 1]$. We observe that, in 3D, because of the homogenous free-slip boundary conditions, for each sensor on the surface, one of three displacement components (i.e., the one normal to the surface) is identically zero. This implies that 6 of the 18 features, extracted from each sensor signal and defined in 4.2, are identically zero. Hence, for the 3D problem with no embedded sensor we consider $Q = 6(d-1)$, i.e., $Q = 12$ for both the 2D and 3D case.

5.3. The free parameters in the Weeks method

As explained in Remark 2, to apply the Weeks method to reconstruct the solution in time, we need to define the free parameters α and b, which are obtained by applying (35). In particular, when $\hat{\mathbf{g}}_i^*$, $i \ldots, N_s$ in (35) are the high-fidelity signals obtained by applying Algorithm 1 to the 2D problem with $\boldsymbol{\mu}^* = [1, 0.33, 2]$, $N_z = 200$, $\Delta t = 1e{-}3$ and $N_t = 30'000$, we obtain $\alpha^{opt} = 0.26$ and $b^{opt} = 6.5$, as shown in Figure 6. For simplicity, these hyper-parameters are also used for all the other problems considered here and for all input parameters (53). Figure 7 shows that, for these optimal values, the error of the reconstructed solution in 2D decreases with second order of convergence as the number of coefficients N_z in the Laugerre's expansion increases. In all our 2D simulations, we use $N_z = 200$, which guarantees good results as shown in Figure 8, where the behavior of the time-dependent solutions (displacements in the $x-$ and $y-$ directions) recovered at the 6^{th} sensor of coordinates $\boldsymbol{x}_6 = (0.275, 0.925)$, using either the Newmark method or the Weeks method, are presented. As time increases, we observe a matching degradation between the solutions in time domain and the reconstructed solution in frequency domain, which is expected considering the expansion in the Laguerre's polynomials. For the 3D simulations, the number of frequencies N_z is increased to 500 to guarantee better alignement with the Newmark solution, considered as a reference solution, and avoid spurious oscillations before the signal arrival. Additionally, in 3D, we consider a reduced time frame of $N_t = 22'500$ time steps to discard incorrect oscillations caused by the Weeks method.

5.4. The training set

We present here the details to construct the training set for the2D and 3D problems, whose geometries, sensors and source locations are shown in Figures 4 and 5, respectively. For both problems, we primarily

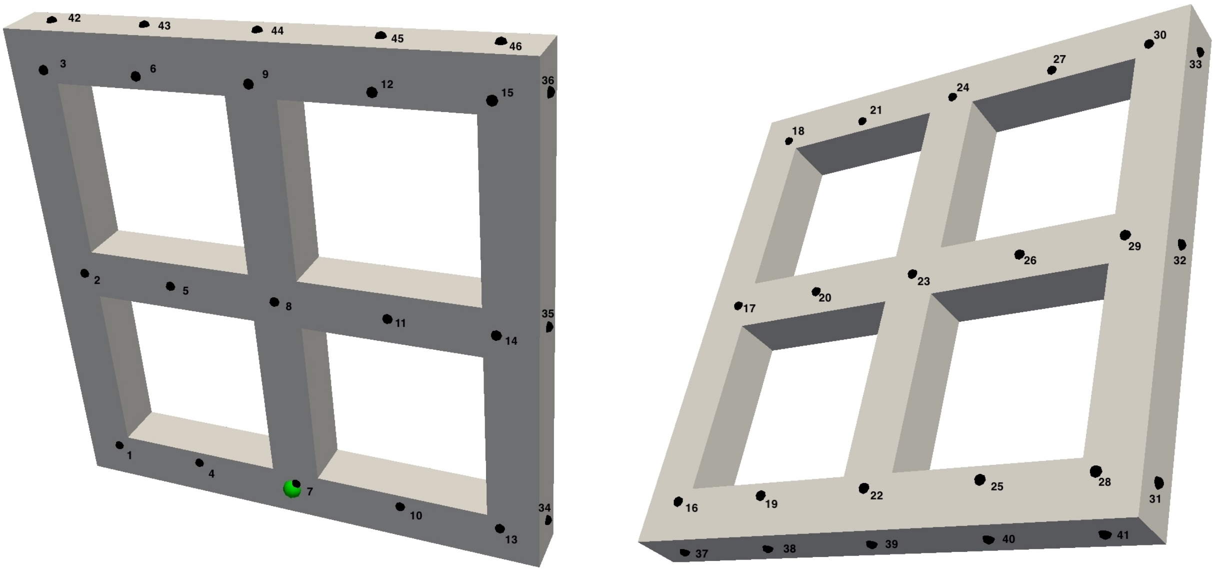

Figure 5: Sketch of sensors numbering system and source placement for the 3D problem. Numbered filled black semi-spheres represent the 46 surface sensors, while the larger green semi-sphere represents the source position, i.e., $\bar{\boldsymbol{\mu}} = [0.51, 0.06, 0]$. The face with coordinate $z = 0$ is shown on the left, while the face with coordinate $z = 0.1$ is shown on the right.

generate a dataset using the RB strategy presented in Section 3.6. For this, we set $k_{POD} = 1e - 11$ and $k_z = 3$. Having chosen $N_z = 200$ and $N_z = 500$ for the 2D and 3D problems, respectively, we consider a total of $n_s = 600$ and $n_s = 1500$ snapshots, respectively. To generate such snapshots, the input parameters $\{\boldsymbol{\mu}_m\}_{m=1}^{n_s}$ are uniformly sampled from $\mathcal{P}$ and the N_z input frequencies are defined in (31). By applying Algorithm 2, we obtain $N = 159$ basis for the 2D problem and $N = 251$ basis for the 3D case. Setting $N_{tr} = 1000$ for both problems, the training datasets $\mathcal{D}_i^{N_{tr}}$ are constructed by solving the reduced problem $N_z N_{tr}$ times and by applying Algorithm 1 for $i = 1, \dots, N_s$.

Finally, after extracting the damage-indicator features as explained in Section 4.2 and applying the PCA reduction to the normalized dataset (see Section 4.3), the OC-SVMs are trained on the reduced-feature-based datasets $\mathcal{F}_i^{N_{tr}} P_{PCA}$ for $i = 1, \dots, N_s$.

5.5. The test set

The test set is composed of both healthy and damaged synthetic sensor measurements. The discrete time signals are obtained by solving the high fidelity problem (24) for N_{test} new input parameters, sampled from the same parameter distribution used offline. As explained in Section 2, we add zero-mean random Gaussian noise to all time steps of all test signals. In particular, for each component of the reconstructed test signals $\hat{\mathbf{g}}_i^{\star}$, we add noise $\varepsilon_i \in \mathcal{N}(0, \boldsymbol{\gamma}_i^2)$, where $\boldsymbol{\gamma}_i$ corresponds to 0.01% of the maximum amplitude of 30 randomly chosen training healthy signals over the first $N_t = 20'000$ steps, component by component. Different from the training set, some of the signals are obtained by solving the PDE on faulty geometries. In particular, in 2D, we consider 9 damage scenarios, sketched in Figure 9, of which 4 are considered major damages $(a - d)$, 4 as minor damages $(e - h)$ and 1 (i) is obtained by combining two major damages. For the healthy configuration and each damaged configuration we consider 10 samples for a total of $N_{test} = 100$ test samples. In 3D, the test set is composed of 1 healthy and 3 damaged configurations (2 major damages and 1 minor damage) for a total of $N_{test} = 40$ test samples, i.e., again 10 samples for each configuration are considered. The geometries are shown in Figure 13.

We compare the high-fidelity solutions obtained in Laplace domain, before and after applying the Weeks method, for healthy and damaged structures in 2D. In particular, the signals retrieved at the 9^{th} sensor, i.e., $\boldsymbol{x}_9 = (0.5, 9.25)$, are provided in Figure 10. The graphs compare two healthy solutions obtained with two input parameters $\boldsymbol{\mu}^*, \boldsymbol{\mu}^{**} \in \mathcal{P}$ and a solution obtained when the beam located between the 8^{th} and 9^{th}

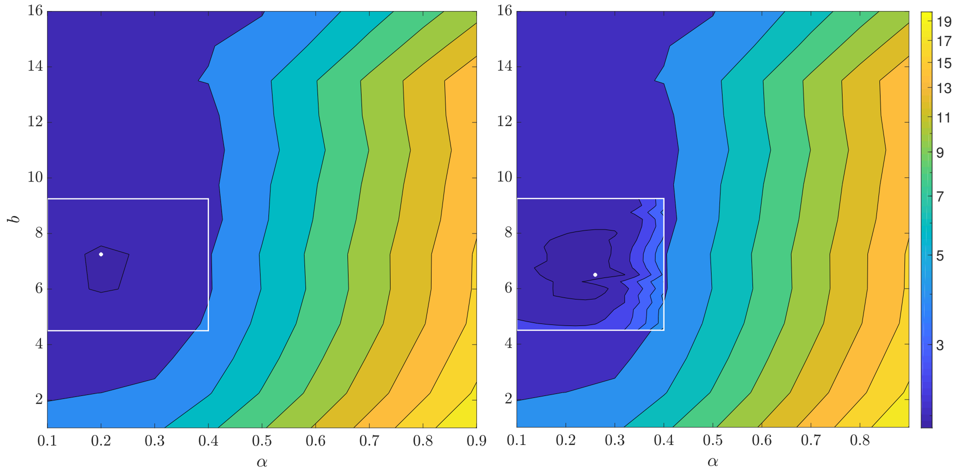

Figure 6: Contour plot of the error obtained using 9 equally spaced points for $\alpha \in [0.1, 0.9]$ and 13 equally spaced points for $b \in [1, 16]$, leading to $\alpha^{opt} = 0.2$ and $b^{opt} = 7.25$ indicated by the white dot (*left*). Additional refinement in the region $\alpha \in [0.1, 0.4]$ and $b \in [4.5, 9.25]$ for 16 and 20 equally spaced points, respectively, leading to the optimal values $\alpha^{opt} = 0.26$ and $b^{opt} = 6.5$ (*right*).

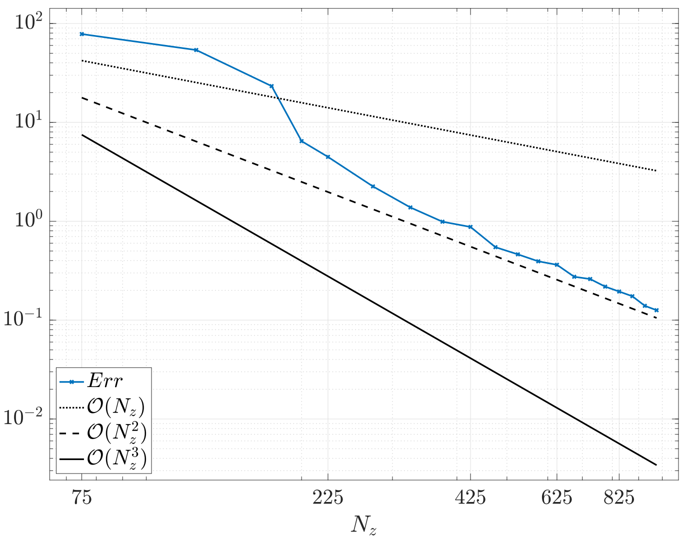

Figure 7: Loglog plot of the error $\left\|\sum_i \left(\hat{\mathbf{g}}_i^* - \mathbf{g}_i^*\right)\right\|_2^2$, where the reconstructed high-fidelity signals $\hat{\mathbf{g}}_i^*$ are obtained using $\alpha^{opt} = 0.26$ and $b^{opt} = 6.5$ for increasing values of N_z. Both $\hat{\mathbf{g}}_i^*$ and $\mathbf{g}_i^*$ are obtained using $N_t = 30'000$ time steps of size $\Delta t = 1e{-}3$ and for input parameter $\boldsymbol{\mu}^* = [1, 0.33, 2]$.

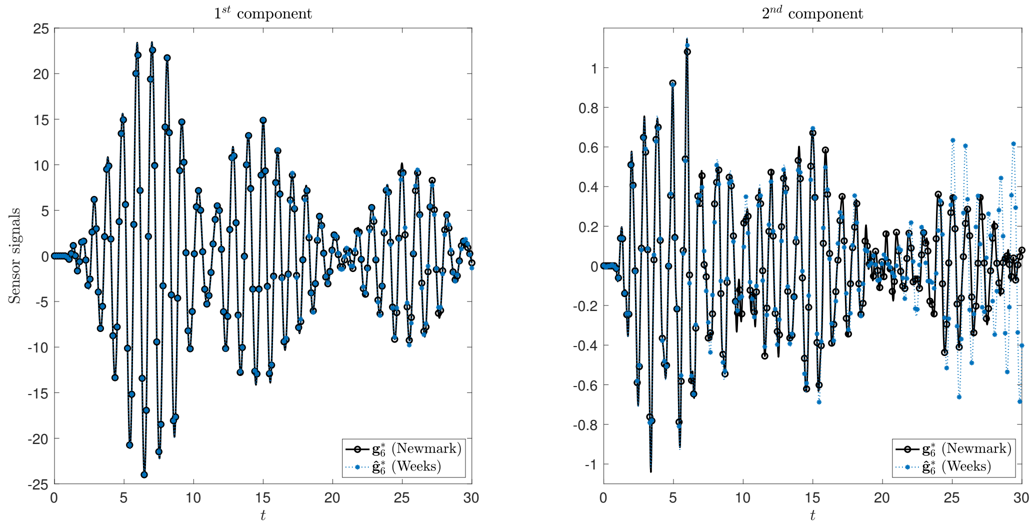

Figure 8: Comparison of the 2D high-fidelity signals retrieved at the 6$^{\text{th}}$ sensor when using the Newmark method (*black circled line*) or the Laplace method with Weeks reconstruction (*blue starred line*) using $\alpha^{opt} = 0.26$, $b^{opt} = 6.5$, $N_z = 200$. Both $\hat{\mathbf{g}}_i^*$ and $\mathbf{g}_i^*$ are obtained using $N_t = 30'000$ time steps of size $\Delta t = 1e{-}3$ and for input parameter $\boldsymbol{\mu}^* = [1, 0.33, 2]$. The first (*left*) and second (*right*) components of the displacement signals are shown.

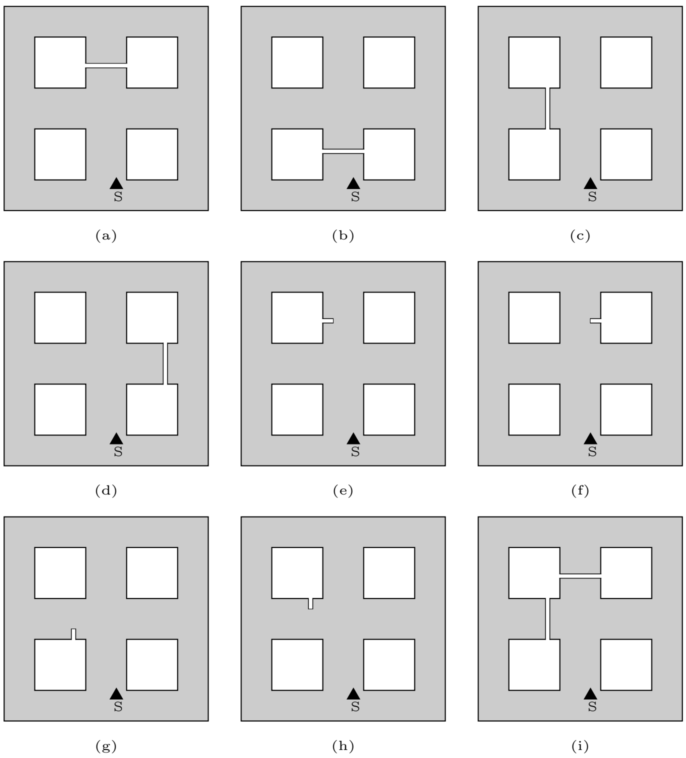

Figure 9: Sketch of 9 damage configurations. Figures (a-d) correspond to major damages, while (e-h) correspond to minor damages. Figure (i) is a superposition of two major damages, i.e., (a) and (c).

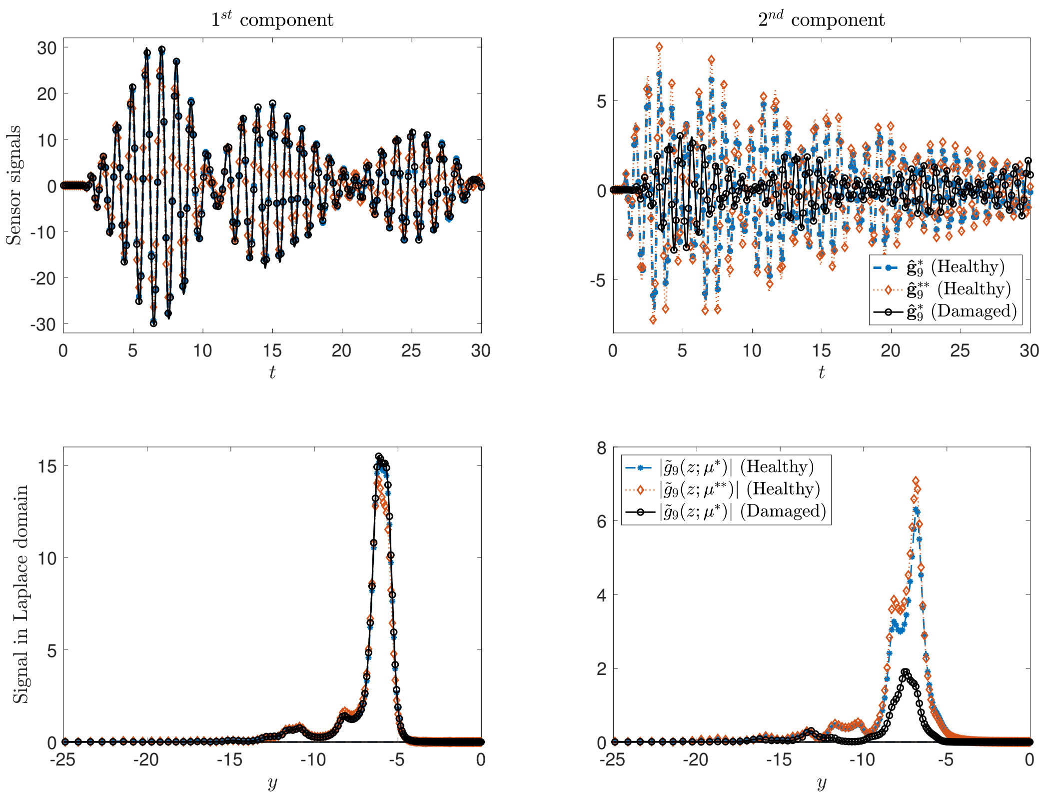

Figure 10: Comparison of 2D signals retrieved at the 9^{th} sensor, obtained from the healthy structure or form a structure with a damage between the 8^{th} and 9^{th} sensor (i.e., damage (a) in Figure 9). From left to right, the first row shows the reconstructed signals obtained using the Weeks method on the first and second component, respectively. The second row shows the absolute value of the raw solutions in Laplace domain. For the four plots, we show two healthy signals, obtained with two different parameters are shown, i.e., $\boldsymbol{\mu}^* = [1, 0.33, 2]$ (*blue dashed line with filled dots*), and $\boldsymbol{\mu}^{**} = [0.9993, 0.3307, 2.07]$ (*orange dotted line with empty diamonds*), and a damaged signal, obtained with $\boldsymbol{\mu}^*$ (*black line with empty dots*).

sensor is broken (see Fig. 9a) using $\boldsymbol{\mu}^*$ as input parameter. Especially for the second component of the solution in Laplace domain and the consequent reconstructed signals, we can observe significant differences between the two healthy signals and the damaged ones. This visual inspection confirms our assumption: signals generated from damaged structure differ from those generated from healthy structures. For this type of damage, signals retrieved at the 9−th sensor happens to be the most affected ones. This can be explained by considering the relative positions of the source, the sensor and the damage, i.e., the damage lies between the source and the receiver, which implies that the signals has to negotiate around the damage to reach the sensor, giving rise to a modified and delayed signal. The same reconstructed solutions, retrieved at sensors 6, 8 and 12, are shown in Figure 11. Qualitatively, we observe some differences between the two healthy signals and the damaged one: damaged signals at sensors 6 and 12 appear to be delayed with respect to the healthy signals, while the signals at sensor 8 are very close for few time-steps and then diverge. These observations can once again be explained by looking at the relative positions of the source, sensors, and damage. Indeed, signals retrieved at sensor 8 begin to diverge when the signals get reflected at the crack. Moreover, after computing the crest factor (51) and arrival time of these signals, we observe that these values are significantly different when looking at the damaged signals or the healthy ones (see Table 1). This observation supports our choice of using, among others, the crest factor and arrival time as damage-indicator features.

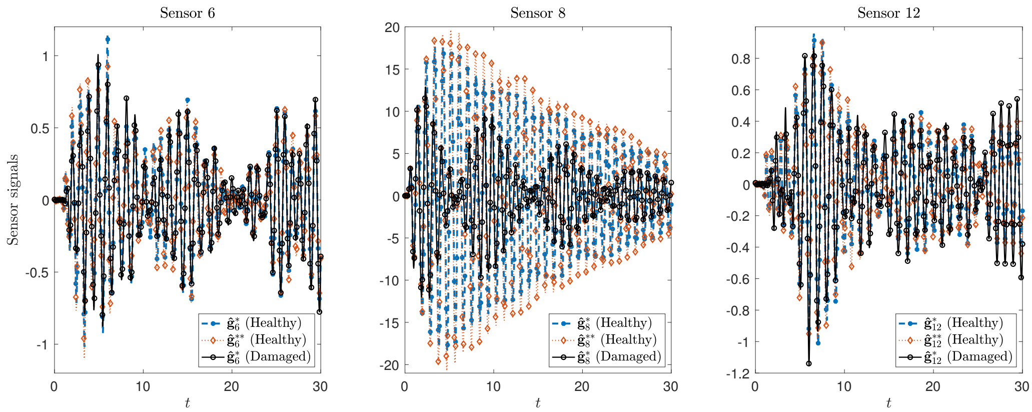

Figure 11: Comparison of the second component of 2D reconstructed signals retrieved at the 6^{th}, 8^{th} and 12^{th}, sensors, obtained from the healthy structure or form a structure with a damage between the 8^{th} and 9^{th} sensor. For the three plots, we show two healthy signals, obtained with two different parameters are shown, i.e., $\boldsymbol{\mu}^* = [1, 0.33, 2]$ (*blue dashed line with filled dots*), and $\boldsymbol{\mu}^{**} = [0.9993, 0.3307, 2.07]$ (*orange dotted line with empty diamonds*), and a damaged signal, obtained with $\boldsymbol{\mu}^*$ (*black line with empty dots*).

Sensor number	CF_1	AT_1	CF_2	AT_2	Parameter	Structure Type
6	3.04	1352	3.41	1005	$\boldsymbol{\mu}^*$	Healthy
	3.16	1354	3.26	1003	$\boldsymbol{\mu}^{**}$	Healthy
	3.74	1868	3.08	1442	$\boldsymbol{\mu}^*$	Damaged
8	3.16	649	2.35	637	$\boldsymbol{\mu}^*$	Healthy
	3.04	647	2.29	631	$\boldsymbol{\mu}^{**}$	Healthy
	3.30	651	3.31	638	$\boldsymbol{\mu}^*$	Damaged
9	3.04	2024	2.78	913	$\boldsymbol{\mu}^*$	Healthy
	3.15	2017	2.85	909	$\boldsymbol{\mu}^{**}$	Healthy
	3.04	2016	3.43	1960	$\boldsymbol{\mu}^*$	Damaged
12	3.14	1389	3.29	1016	$\boldsymbol{\mu}^*$	Healthy
	3.16	1381	3.11	1013	$\boldsymbol{\mu}^{**}$	Healthy
	3.15	2554	3.67	1750	$\boldsymbol{\mu}^*$	Damaged

Table 1: Comparison of crest factor (CF) and arrival time (AT) for high-fidelity reconstructed 2D signals at four different sensor locations for the healthy structure (see Fig. 4) and a damaged (see Fig. 9a) configuration. The retrieved signals are obtained using two input parameters, i.e. $\boldsymbol{\mu}^* = [1, 0.33, 2]$ and $\boldsymbol{\mu}^{**} = [0.9993, 0.3307, 2.07]$. The subscript indicates the signal component.

Sensor	Healthy	Damage (a)	Damage (b)	Damage (c)	Damage (d)	Minor damage (e)	Minor damage (f)	Minor damage (g)	Minor damage (h)	Combined damage (i)
1	0	0.4	0.4	0	0.1	0.1	0	0	0.1	0.1
2	0	0.5	1	1	0	0.2	0	0	0.5	1
3	0	0.7	0.1	0.7	0	0	0	0	0.1	0.3
4	0	0	0.9	0.4	0	0.1	0.5	0.1	0.1	0.1
5	0	0	1	1	1	0.2	0	1	0	1
6	0	1	1	0	0	1	0	0	0	1
7	0	0.1	1	0	0	0	0	0	0	0.1
8	0	0.3	1	1	0	0.1	0.5	0.1	0.1	1
9	0	1	1	0	0	0.1	0.1	0	0	1
10	0	0	0	0.3	0.1	0.1	0	0.2	0.1	0.1
11	0	0.1	1	0.1	1	0	0	0.2	0	0
12	0.1	1	1	0	0.1	0.5	1	0	0	1
13	0	0.1	0	0.1	0	0	0	0	0.3	0.1
14	0	0.1	1	0.1	1	0.2	0	0.1	0.1	0.8
15	0	1	1	0.1	0.1	1	0	0	0.1	1

Table 2: Fractions of test samples for the 2D problem classified as outliers (i.e., with anomaly score (50) greater than 1) for the healthy configuration (see Fig. 4) and 9 damaged configurations (see Fig. 9). A set of 10 uniformly sampled input parameters $\{\mu_m^\star\}_{m=1}^{10} \in \mathcal{P}$ is used to construct 10 test samples per configuration.

5.6. Classification results

We present here the one-class classification results on the test sets, sensor by sensor. In 2D, the test set is composed of $N_{test} \coloneqq 100$ samples, i.e., 10 samples for each one of the 10 configurations (1 healthy and 9 damaged). In 3D, $N_{test} \coloneqq 40$ samples, i.e., 10 samples for each one of the 4 configurations (1 healthy and 3 damaged), compose the test set. In both cases, each one of the 10 samples is obtained by solving the high fidelity problem with different input parameters $\boldsymbol{\mu}$. Tables 2 and 3 show, for each type of damage, the fraction of test samples classified as outliers, i.e., with an anomaly score greater than 1, while the mean values for each damaged configurations are shown in Figures 12 and 13, for the 2D and 3D problems, respectively. Sensors whose average anomaly score is greater than 1 are represented with red markers, while blue markers identify the sensors with average anomaly score smaller than 1. For visualization purposes, we introduce an arbitrary value to additionally differentiate between strong and mild outliers; i.e., strong outliers are those with mean anomaly score greater or equal than 2, while mild outliers have mean anomaly score greater than or equal to 1, but smaller 2. Strong outliers are represented with red squares, while mild outliers with red asterisks in 2D and red semi-spheres in 3D.

We observe that, both in 2D and 3D, on average, damages are always detected, i.e., at least one sensor is classified as outlier if the structure is damaged, and that, in most of the cases, damages are close to the sensors that are classified as strong outliers. Even if not reported in Figure 12, all sensors of 2D healthy configuration are, on average, classified as inliers, while the average result for the 3D healthy configuration (Figure 13 a) presents 1 misclassified sensors. In general, the 3D results present a slightly higher false alarm rate than the 2D problem, even though it is still possible to identify a macro-region where the damage is located (see Figure 13).

The relative position of source, sensors and damage is important to successfully use this approach to locate the damage. Indeed, in 2D, for the major damages (a, c, d, i), only the sensor "behind" the damage are classified as outliers, allowing for localization. Instead, with the 2D damage (b) positioned too close to the source, 11 out of 15 sensors are, on average, classified as outliers, thus preventing localization. A similar behavior is observed in the 3D results. The combination of solutions obtained with different active sources at different locations is likely to address this issue. For example, we refer to [53], where piezoelectric transducers are used as both sensors and actuators for Lamb wave propagation. In this work, once the damaged path-ways between each couple of sensor/actuator have been determined, the location of damages is identified with the regions with higher number of intersecting damaged pathways. Alternative solutions are reported in [16].

6. Conclusion

We propose a data-driven approach for SHM which leverages the physics-based representation of the structure of interest. From a mathematical standpoint, the goal of data-driven approaches is classification,

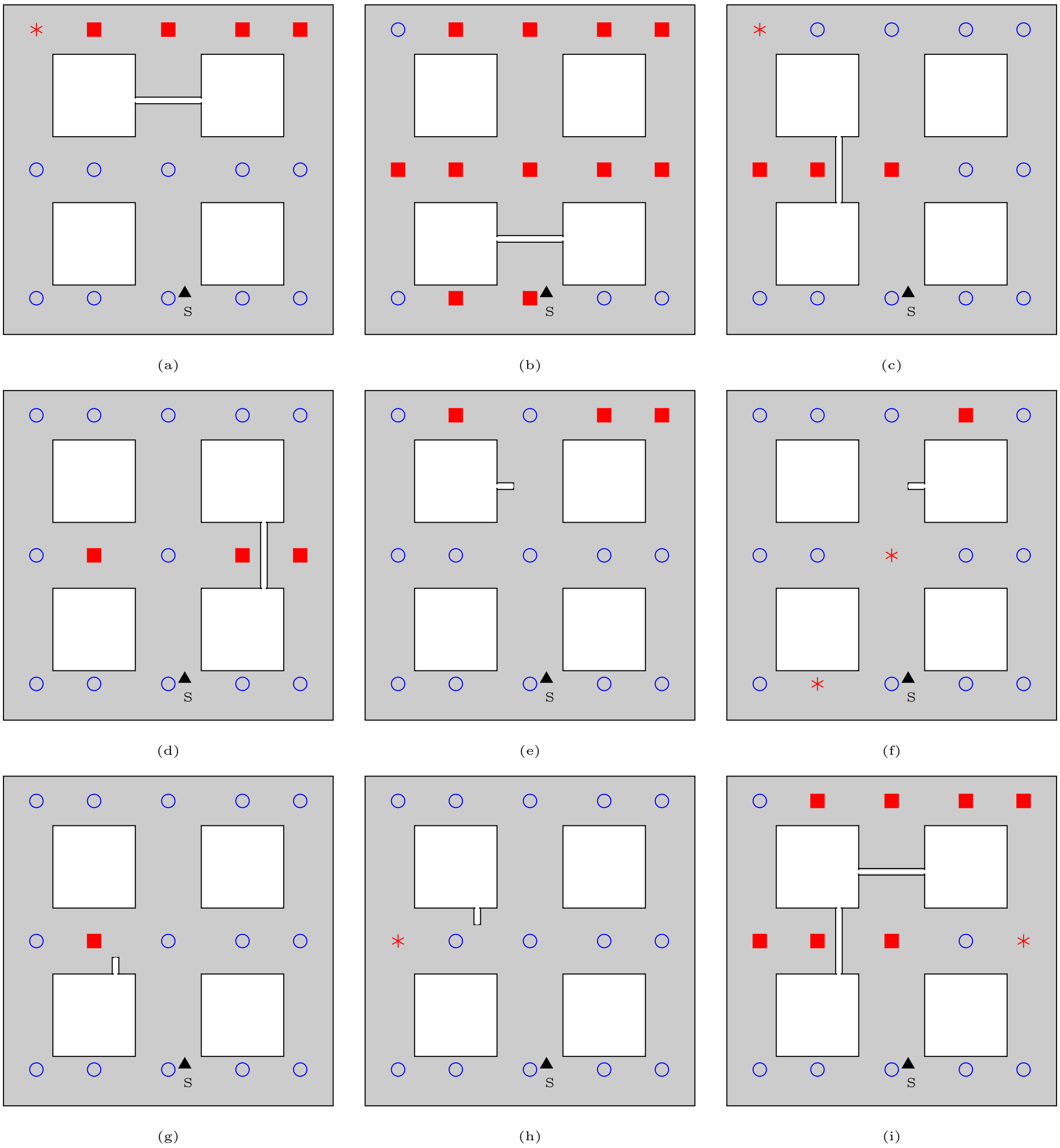

Figure 12: Sketch to summarize of the one-class classification average results on test data for 9 damaged configurations. Red filled squares correspond to sensors classified as outliers with an average score $f_i^{N_{tr}} \geq 2$ (strong outliers), red asterisks to sensors classified as outliers with an average score $f_i^{N_{tr}} \in [1, 2[$ (mild outliers), and blue empty circles to sensors classified as inliers, i.e., with an average score $f_i^{N_{tr}} < 1$. The black triangles labeled with the letter S indicate the source position. For all types of damages we can identify at least one sensor classified as a outlier. With the exception of damage (b), a clear proximity between the location of the damages and the sensors classified as outliers can be observed. The position of the source plays an important role in classification and therefore, to localize damage (b), the source should be placed differently. For major damages (a, c, d), 3 to 4 sensors are classified as strong outliers and at most 1 as as mild outlier with a maximum total of 5 sensors classified as outliers. For minor damages (e, f, g, h) from 1 to 3 sensors are classified as outliers. For the combined damage (i) 7 sensors are classified as strong outliers and 1 as mild outlier.

as opposed to model-based approaches where the goal is to solve an inverse problem and estimate the (unknown) input parameters.

Damage detection and localization is carried out on a sensor-by-sensor basis by constructing synthetic training data emulating the sensor response of the structure to active sources, i.e., we analyze the structural response to the propagation of guided waves. These training databases are constructed offline by repeatedly solving PDEs in the frequency domain for different input parameters and by exploiting MOR techniques for speedup. The reconstruction of time signals is carried out using the Weeks method, a numerical inverse Laplace transform. The set of input parameters used to generate the dataset represents the natural variations of the structure, i.e., the environmental and operational conditions, and provides the baseline variability. After extracting damage-sensitive engineering-based features from the raw discrete signals, we employ one-class classifiers, the OC-SVM algorithm, to compare the healthy training dataset with new blind test data. The latter are obtained by extracting the same features from high-fidelity signals obtained by solving the PDEs for unseen input parameters and by possibly modifying the geometry to include cracks of different sizes and at different locations. Noise is added to the test signals to emulate the unknown experimental sensor response.

This approach is successful in both detecting and localizing damages for 2D and 3D digital twins test problems. The method is highly generalizable to other examples and more realistic experiments will be carried out within a laboratory environment to validate our approach. We observe that, using active sources, localization is possible only for damages which are sufficiently far from the source. To address this limitation, we will investigate the possibility of introducing a network of sources placed at different locations. The source location could be used as additional input parameter to construct the RB model and the combination of different classification results could help gain insight on damages on the entire domain. Moreover, the offline-online decoupling of tasks and the MOR techniques allow us to compute the sensor response under different operational and environmental conditions in a fast and inexpensive manner. By exploiting this advantage, we aim to study the optimal placement of sensors needed to both retrieve maximum information about the potential structure damages and guarantee a robust network of sensors, which aims to maintain the stability of the network even when some sensors malfunction. Finally, alternative passive periodic sources, mimicking the effect of tides or wind, could be integrated in the model by replacing the Laplace transform with the Fourier transform. In this case, the features used as damage-indicators would need to be adapted or alternative anomaly detection strategies like the autoencoders should be employed to automatically identify the underling characteristics of healthy signals.

Acknowledgments

We would like to show our sincere gratitude to Dr. Francesco Ballarin for his help in the setup and further extension of our RBniCS code. Dr. Niccolò Dal Santo, Dr. Cecilia Pagliantini and Nicolò Ripamonti are thanked for the fruitful discussions and their advice on reduced basis methods.

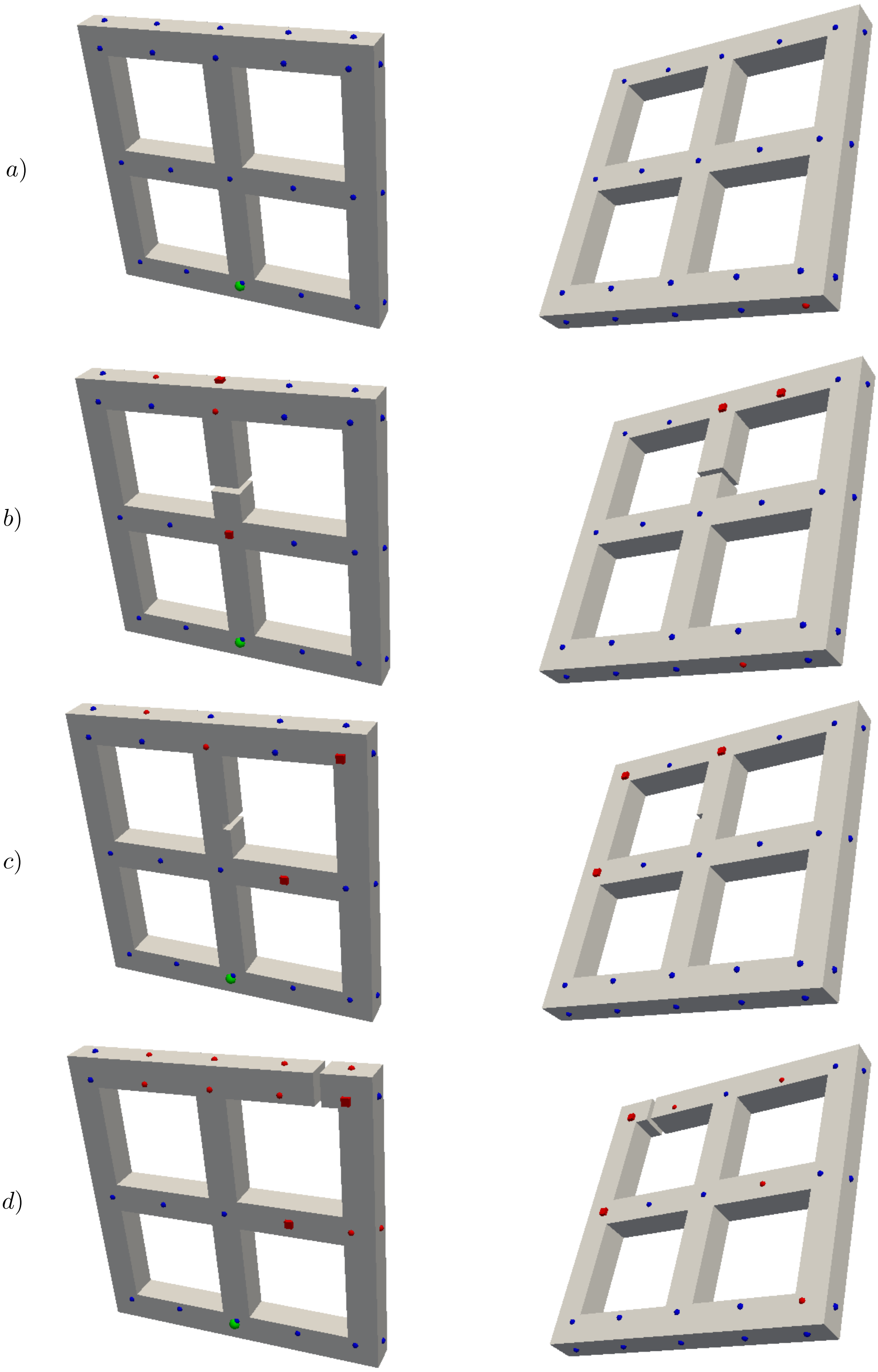

Figure 13: Sketch to summarize the geometries of the 4 configurations used in the test set for the 3D problem, together with the one-class classification average results. For each configuration, sensors represented by red squares indicate that the mean classification score is above 2 (i.e., the sensor is classified as strong outlier on average), red semi-spheres indicate sensors classified as mild outliers, , i.e. with mean anomaly score between 1 and 2, and blue semi-spheres represent sensors classified as inliers, i.e. with mean anomaly score below 1. The green larger semi-sphere indicates the source position. The left and right plots show the front ($z = 0$) and rear ($z = 0.1$) of the 3D configurations. For the damaged configurations, a correlation between sensors classified as outliers and location of damage can be identified. A low false positive error is observed for both the healthy and damaged configurations: 1 sensor is misclassified in the healthy configuration a and few sensors, far from the damages, are mistakenly classified as mild outliers, especially for the damaged configuration d.

Sensor	Healthy (a)	Damage (b)	Minor damage (c)	Damage (d)
1	0	0	0.2	0
2	0	0	0.1	1
3	0	0	0	1
4	0	0	0	0
5	0.4	0	0	1
6	0	0	0	1
7	0	0	0	0
8	0	1	0.3	0
9	0	1	1	1
10	0	0	0	0
11	0	0.1	0	0
12	0	0	0	1
13	0	0	0	0.6
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0
17	0	0	0.3	1
18	0	0	0	1
19	0	0.3	0	0
20	0	0	0	0
21	0	0.3	0	1
22	0	0	0	0
23	0	0	0	0
24	0	1	1	0
25	0	0	0	0
26	0	0	0	0.9
27	0	1	0	1
28	0	0	0	0.6
29	0	0	0	0
30	0	0	0	0
31	0	0	0	0.2
32	0	0	0.3	1
33	0	0	0	0
34	0	0	0	0.3
35	0.6	0	0	0
36	0	0	0	0
37	0	0	0	0
38	0	0.3	0	0.1
39	0	0	0	0
40	0	1	0	0
41	0.7	0	0	0
42	0	0	0	1
43	0	0.4	0	1
44	0	1	0	1
45	0	1	0.8	1
46	0.6	0	0.1	0

Table 3: Fractions of test samples for the 3D problem classified as outliers (i.e., with anomaly score (50) greater than 1) for the healthy configuration and the 3 damaged configurations (see Fig. 13). A set of 10 uniformly sampled input parameters $\{\mu_m^\star\}_{m=1}^{10} \in \mathcal{P}$ is used to construct 10 test samples per configuration.

References

[1] D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.

[2] M. Amer, M. Goldstein, and S. Abdennadher. Enhancing one-class Support Vector Machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, pages 8–15. ACM, 2013.

[3] A. Anaissi, N. L. D. Khoa, S. Mustapha, M. M. Alamdari, A. Braytee, Y. Wang, and F. Chen. Adaptive one-class Support Vector Machine for damage detection in Structural Health Monitoring. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 42–57. Springer, 2017.

[4] F. Ballarin, A. Sartori, and G. Rozza. RBniCS - reduced order modelling in FEniCS. http://mathlab.sissa.it/rbnics, 2016.

[5] C. M. Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

[6] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[7] C. Clenshaw. A note on the summation of Chebyshev series. *Mathematics of Computation*, 9(51):118–120, 1955.

[8] N. Cristianini and B. Scholkopf. Support Vector Machines and Kernel Methods: The New Generation of Learning Machines. *Ai Magazine*, 23(3):31, 2002.

[9] K. S. Crump. Numerical inversion of Laplace transforms using a Fourier series approximation. *Journal of the ACM (JACM)*, 23(1):89–96, 1976.

[10] S. Das, A. N. Srivastava, and A. Chattopadhyay. Classification of damage signatures in composite plates using one-class SVMs. In *Aerospace Conference, 2007 IEEE*, pages 1–19. IEEE, 2007.

[11] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire. An experimental evaluation of novelty detection methods. *Neurocomputing*, 135:313–327, 2014.

[12] S. Doebling, C. Farrar, M. Prime, and D. Shevitz. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review. *Technical Report No.: La-13070-MS Los Alamos National Laboratory*, 5 1996.

[13] H. Dubner and J. Abate. Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform. *Journal of the ACM (JACM)*, 15(1):115–123, 1968.

[14] D. G. Duffy. On the Numerical Inversion of Laplace Transforms: Comparison of Three New Methods on Characteristic Problems from Applications. *ACM Transactions on Mathematical Software (TOMS)*, 19(3):333–359, 1993.

[15] F. Durbin. Numerical inversion of Laplace transforms: an efficient improvement to Dubner and Abate's method. *The Computer Journal*, 17(4):371–376, 1974.

[16] C. R. Farrar and K. Worden. *Structural Health Monitoring: a Machine Learning Perspective*. John Wiley & Sons, 2012.

[17] B. S. Garbow, G. Giunta, J. N. Lyness, and A. Murli. Software for an implementation of Weeks' method for the inverse Laplace transform. *ACM Transactions on Mathematical Software (TOMS)*, 14(2):163–170, 1988.

[18] G. Giunta, G. Laccetti, and M. Rizzardi. More on the Weeks method for the numerical inversion of the Laplace transform. *Numerische Mathematik*, 54(2):193–200, 1989.

[19] M. Goldstein and S. Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[20] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[21] M. Gresil, A. Poohsai, and N. Chandarana. Guided wave propagation and damage detection in composite pipes using piezoelectric sensors. *Procedia Engineering*, 188:148–155, 2017.

[22] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. SpringerBriefs in Mathematics. Springer International Publishing, 2015.

[23] C. O. Horgan. Korn's inequalities and their applications in continuum mechanics. *SIAM review*, 37(4):491–511, 1995.

[24] J. E. Hurtado and D. A. Alvarez. Classification approach for reliability analysis with stochastic finite-element modeling. *Journal of Structural Engineering*, 129(8):1141–1149, 2003.

[25] J.-B. Ihn and F.-K. Chang. Detection and monitoring of hidden fatigue crack growth using a built-in piezoelectric sensor/actuator network: I. diagnostics. *Smart materials and structures*, 13(3):609, 2004.

[26] N. Japkowicz, C. Myers, M. Gluck, et al. A novelty detection approach to classification. In *IJCAI*, volume 1, pages 518–523, 1995.

[27] N. M. Khan, R. Ksantini, I. S. Ahmad, and L. Guan. Covariance-guided one-class Support Vector Machine. *Pattern Recognition*, 47(6):2165–2177, 2014.

[28] S. Khazai, S. Homayouni, A. Safari, and B. Mojaradi. Anomaly detection in hyperspectral images based on an adaptive Support Vector method. *IEEE Geoscience and Remote Sensing Letters*, 8(4):646–650, 2011.

[29] N. L. Khoa, B. Zhang, Y. Wang, F. Chen, and S. Mustapha. Robust dimensionality reduction and damage detection approaches in Structural Health Monitoring. *Structural Health Monitoring*, 13(4):406–417, 2014.

[30] J.-T. Kim, Y.-S. Ryu, H.-M. Cho, and N. Stubbs. Damage identification in beam-type structures: frequency-based method vs mode-shape-based method. *Engineering structures*, 25(1):57–67, 2003.

[31] M. Lecerf, D. Allaire, and K. Willcox. Methodology for Dynamic Data-Driven Online Flight Capability Estimation. *AIAA Journal*, 53(10):3073–3087, 2015.

[32] X. Lin and F. Yuan. Diagnostic Lamb waves in an integrated piezoelectric sensor/actuator plate: analytical and experimental studies. *Smart Materials and Structures*, 10(5):907, 2001.

[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[34] S.-W. Liu, J. H. Huang, J.-C. Sung, and C. Lee. Detection of cracks using neural networks and computational mechanics. *Computer methods in applied mechanics and engineering*, 191(25-26):2831–2845, 2002.

[35] A. Logg, K.-A. Mardal, G. N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.

[36] J. Long and O. Buyukozturk. Automated Structural Damage Detection Using One-Class Machine Learning. In *Dynamics of Civil Structures, Volume 4*, pages 117–128. Springer, 2014.

[37] J. N. Lyness and G. Giunta. A Modification of the Weeks Method for Numerical Inversion of the Laplace Transform. *Mathematics of Computation*, 47(175):313–322, 1986.

[38] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller. A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 1996–2000. IEEE, 2015.

[39] MATLAB. *Version 9.5 (R2018b)*. The MathWorks Inc., Natick, Massachusetts, 2018.

[40] P. Mitra, C. Murthy, and S. K. Pal. Unsupervised feature selection using feature similarity. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):301–312, 2002.

[41] P. C. S. Nadith, L. Jun, L. Ling, H. Hong, and L. Wanquan. Application of deep autoencoder model for structural condition monitoring. *Journal of Systems Engineering and Electronics*, 29(4):873–880, 2018.

[42] N. M. Newmark et al. A method of computation for structural dynamics. American Society of Civil Engineers, 1959.

[43] Y. Ou, E. N. Chatzi, V. K. Dertimanis, and M. D. Spiridonakos. Vibration-based experimental damage detection of a small-scale wind turbine blade. *Structural Health Monitoring*, 16(1):79–96, 2017.

[44] L. Peng and K. Mohseni. Symplectic model reduction of Hamiltonian systems. *SIAM Journal on Scientific Computing*, 38(1):A1–A27, 2016.

[45] R. Piessens and B. Maria. Numerical inversion of the Laplace transform using generalised Laguerre polynomials. In *Proceedings of the Institution of Electrical Engineers*, volume 118, pages 1517–1522. IET, 1971.

[46] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[47] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2015.

[48] A. Quarteroni and F. Saleri. Scientific Computing with MATLAB and Octave, 2006.

[49] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1, 2007.

[50] A. Sadri and K. Mirkhani. Wave Propagation Concrete NDT Techniques for Evaluation of Structures and Materials. 2009.

[51] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[52] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt. Support Vector Method for Novelty Detection. In *Advances in neural information processing systems*, pages 582–588, 2000.

[53] R. A. Swartz, E. Flynn, D. Backman, R. J. Hundhausen, and G. Park. Active piezoelectric sensing for damage identification in honeycomb aluminum panels. In *Proceedings of 24th Intl. Modal Analysis Conference*, 2006.

[54] T. Taddei, J. Penn, M. Yano, and A. Patera. Simulation-Based Classification; a Model-Order-Reduction Approach for Structural Health Monitoring. *Archives of Computational Methods in Engineering*, 25(1):23–45, 2018.

[55] A. Talbot. The accurate numerical inversion of Laplace transforms. *IMA Journal of Applied Mathematics*, 23(1):97–120, 1979.

[56] D. M. Tax and K.-R. Müller. Feature extraction for one-class classification. In *Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003*, pages 342–349. Springer, 2003.

[57] R. Unnthorsson. Model selection in one class ν-SVMs using RBF kernels. *Proc. of 16th Int. Cong. CAMADEM, 2003*, 2003.

[58] V. N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, 1998.

[59] W. Wang, Z. Xu, W. Lu, and X. Zhang. Determination of the spread parameter in the Gaussian kernel for classification and regression. *Neurocomputing*, 55(3-4):643–663, 2003.

[60] W. T. Weeks. Numerical Inversion of Laplace Transforms Using Laguerre Functions. *Journal of the ACM (JACM)*, 13(3):419–429, 1966.

[61] J. A. C. Weideman. Algorithms for Parameter Selection in the Weeks Method for Inverting the Laplace Transform. *SIAM Journal on Scientific Computing*, 21(1):111–128, 1999.

[62] Y. Xiao, H. Wang, and W. Xu. Parameter selection of Gaussian kernel for one-class SVM. *IEEE transactions on cybernetics*, 45(5):941–953, 2015.

[63] K. Zgonc and J. D. Achenbach. A neural network for crack sizing trained by Finite Element calculations. *NDT & E International*, 29(3):147–155, 1996.

[64] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method for solid and structural mechanics.* Elsevier, 2005.

[65] F. Zou, I. Benedetti, and M. Aliabadi. A Boundary Element Model for Sructural Health Monitoring using piezoelectric transducers. *Smart Materials and Structures*, 23(1):015022, 2013.