Thèse n° 7734

Numerical methods for structural anomaly detection using model order reduction and data-driven techniques

Présentée le 25 septembre 2020

à la Faculté des sciences de base
Chaire de mathématiques computationnelles et science de la simulation
Programme doctoral en mathématiques

pour l'obtention du grade de Docteur ès Sciences

par

Caterina BIGONI

Acceptée sur proposition du jury

Prof. K. Hess Bellwald, présidente du jury
Prof. J. S. Hesthaven, directeur de thèse
Prof. E. Chatzi, rapporteuse
Dr T. Taddei, rapporteur
Prof. F. Nobile, rapporteur

École
polytechnique
fédérale
de Lausanne

2020

ABSTRACT

The objective of this thesis is to provide a mathematical and computational framework for the proactive maintenance of complex systems with a particular application to structural health monitoring (SHM). SHM techniques rely primarily on sensor responses to assess the risk associated with a structure of interest and seek to provide a support for automated decision-making strategies. An efficient integration of experimental measurements and numerical models is needed to exhaustively describe the environmental and operational scenarios that a structure undergoes during its life time.

We propose a simulation-based approach that combines the solution of a parametric time-dependent partial differential equation (PDE) for multiple input parameters with data-driven techniques to discriminate between healthy and damaged configurations. This process exploits an offline-online decomposition of tasks. The dataset of synthetic sensor measurements is generated offline by repeatedly solving a parametric PDE for a predefined configuration under several healthy variations. A reduced order model is employed to overcome the computational bottleneck associated with the many-query context by building a projection-based reduced basis method in combination with a Laplace-domain solver. Weeks method is adopted to numerically invert the Laplace transform and transform the signals to the time domain. The datasets are used to train various one-class machine learning algorithms in a semi-supervised setting, sensor by sensor. Finally, the outputs of the classifiers are used to assess the state of damage of the structure online. Using a decision-level fusion strategy, we provide insight on the existence, location, and severity of possible damages.

The performance of SHM depends critically on the quality of the sensor measurements although their availability is often limited due to budget constraints and installation difficulties. We therefore propose a strategy to systematically place a fixed number of sensors on a structure of interest to minimize uncertainty at *unsensed* locations. The so-called inducing points, an outcome of sparse Gaussian processes, originally introduced to overcome the computational burden associated with performing a regression task with standard Gaussian processes, are here used to guide the sensor placement. A clustering approach is employed to select the sensor locations among a set of inducing points,

computed for different input parameters.
We apply this methodology to 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 (*i*) detecting damaged geometries and (*ii*) identifying the locations for a network of sensors. This framework considers the realistic case where damage types and locations are a priori unknown, thus overcoming the main limitation of existing simulation-based damage detection and sensor placement strategies for SHM.

Key words: simulation-based anomaly detection, reduced order models, one-class classification, sensor placement, structural health monitoring, digital twin.

RÉSUMÉ

L'objectif de cette thèse est de fournir un cadre mathématique et computationnel pour une maintenance proactive de systèmes complexes avec une application particulière à la surveillance de la santé structurelle. Les techniques de surveillance de la santé structurelle s'appuient principalement sur les réponses de capteurs pour évaluer le risque associé à une structure et cherchent à informer le développement d'une stratégie de prise de décision automatique. Une intégration efficace des mesures expérimentales et des modèles numériques est nécessaire afin de décrire de manière exhaustive les scénarios environnementaux et opérationnels qu'une structure subit au cours de sa vie.

Nous proposons ici une approche basée sur des simulations qui combine d'une part la résolution d'une équation aux dérivées partielles (EDP) paramétrique et dépendante du temps pour une multitude de paramètres et d'autre part des techniques guidées par les données (data-driven) ayant pour but de différencier entre des configurations saines et endommagées. Ce processus exploite une décomposition des tâches entre une phase en amont (offline) et une phase en temps réel (online). Des mesures synthétiques de capteurs sont générées en amont en résolvant l'EDP paramétrique pour plusieurs instances saines d'une configuration prédéfinie. Pour surpasser le coût computationnel lié à la résolution répétée de l'EDP, nous utilisons un modèle d'ordre réduit en construisant une base réduite associée au problème de départ dans le domaine de Laplace. Nous choisissons la méthode de Weeks pour inverser numériquement la transformée de Laplace et transformer le signal dans le domain temporel. Les mesures synthétiques sont ensuite utilisées pour entraîner plusieurs algorithms d'apprentissage automatique à classe unique dans un contexte d'apprentissage semi-supervisé capteur par capteur. Finalement, les résultats de ces classificateurs sont utilisés pour évaluer l'état d'endommagement de la structure en temps réel. Nous parvenons ainsi à fournir un aperçu de l'existence, de la position, et de la sévérité de possibles endommagements en utilisant une stratégie de décision regroupant les informations après la classification locale.

La performance de la surveillance de la santé structurelle est extrêmement influencée par la qualité des mesures des capteurs, or leur nombre est souvent limité pour des questions de budget ou de difficulté d'installation. Nous proposons alors une stratégie afin de placer

systématiquement un nombre prédéfini de capteurs sur une structure en minimisant l'incertitude aux points ne disposant pas de capteurs. Les points nommés inducteurs, résultats de processus Gaussiens sparses introduits originellement afin de surpasser le coût computationnel associé au calcul de régression dans les processus Gaussiens standards, sont ici utilisés comme guides pour le placement des capteurs. Une approche de regroupement (clustering) est utilisée pour sélectionner la position des capteurs parmi une série de points inducteurs calculés pour différentes valeurs des paramètres d'entrée.
Nous appliquons cette méthodologie à des problèmes en 2D et en 3D en simulant le comportement vibratoire de structures complexes sous l'influence d'une source active et nous montrons l'efficacité de cette approche pour (*i*) détecter des géométries endommagées et (*ii*) identifier où installer un réseau de capteurs de taille prédéfinie. Nous considérons un cas réaliste pour la surveillance de la santé structurelle où les types d'endommagements et leurs positions sont a priori inconnus, surmontant ainsi la limitation principale des stratégies dans la literature qui, au contraire, simulent les configurations saines ainsi que celles endommagées, perdant ainsi en généralité.

Mots cléfs : détection d'anomalies basée sur des simulations, models d'ordre réduit, classement automatique à classe unique, positionnement des capteurs, surveillance de la santé structurelle, jumeau numérique.

SOMMARIO

L'obiettivo di questa tesi è quello di fornire un quadro matematico e computazionale per la manutenzione preventiva di sistemi complessi con una particolare attenzione ad applicazioni nel campo del monitoraggio dell'integrità strutturale. Le tecniche per il monitoraggio dell'integrità strutturale valutano il rischio associato ad una struttura basandosi su segnali provenienti da sensori posizionati sulla stessa ed ambiscono a sviluppare una strategia basata su un processo decisionale automatico. Un'efficace unione di misure sperimentali e di modelli numerici è necessaria al fine di descrivere in maniera esauriente i diversi scenari ambientali e operativi che una struttura subisce nel corso della sua vita.

Proponiamo un approccio fondato su simulazioni che combina la risoluzione di un'equazione alle derivate parziali (EDP) parametrizzata e tempo-dipendente per un elevato numero di parametri di input con delle tecniche data-driven per poter distinguere le configurazioni sane da quelle danneggiate. Questo processo si serve di una decomposizione offline-online. I dati sensoriali sintetici sono generati offline risolvendo l'EDP parametrizzata considerando molteplici istanze sane per la stessa configurazione predefinita. Per abbattere i costi computazionali legati alla bisogno di valutare la soluzione di un grande numero di problemi, utilizziamo un modello di ordine ridotto costruendo una base ridotta associata al problema originale nel dominio di Laplace. Successivamente, facciamo affidamento al metodo di Weeks per invertire numericamente la trasformata di Laplace e trasformare il segnale nel dominio del tempo. I dati sintetici sono poi usati per allenare diversi algoritmi di apprendimento automatico a classe unica in un contesto di apprendimento semi-supervisionato, sensore per sensore. Infine, i risultati dei classificatori, utilizzando una strategia di raggruppamento delle informazioni dopo la classificazione locale, forniscono una panoramica sull'esistenza, posizione, e gravità dei possibili danni. La qualità dei dati sensoriali costituisce un fattore cruciale per la performance del monitoraggio dell'integrità strutturale, tuttavia il numero di sensori disponibili è spesso limitato da questioni legate al costo o alla difficoltà di installazione. Proponiamo quindi una strategia per collocare sistematicamente un numero predefinito di sensori su una struttura di interesse in modo da minimizzare l'incertezza ove i sensori non sono presenti.

I cosiddetti punti induttori (*inducing points* in inglese), un risultato dei processi Gaussiani sparsi, originariamente introdotti per abbattere i costi computazionali associati all'analisi della regressione nei processi Gaussiani standard, sono qui utilizzati come guida per il posizionamento dei sensori. Successivamente, adottiamo un approccio di analisi dei gruppi (*clustering* in inglese) per selezionare le posizioni finali fra una serie di punti induttori calcolati per diversi valori dei parametri di input.
Applichiamo questa metodologia a dei problemi in 2D e 3D simulando il comportamento vibratorio di strutture complesse sotto l'effetto di una sorgente attiva e ne dimostriamo l'efficacia nel (*i*) rilevare le geometrie danneggiate e (*ii*) identificare le posizioni in cui collocare la rete di sensori. Il nostro approccio per il monitoraggio dell'integrità strutturale considera il caso realistico in cui le tipologie di danni e le loro posizioni sono a priori sconosciute, superando così il limite principale delle strategie in letteratura che, contrariamente, simulano sia configurazioni sane che danneggiate, perdendo così di generalità.

Parole chiave: rilevamento di anomalie basato su simulazioni, modelli di ordine ridotto, apprendimento automatico a classe unica, posizionamento di sensori, monitoraggio dell'integrità strutturale, gemello digitale.

ACKNOWLEDGEMENTS

I foremost wish to thank Prof. Jan S. Hesthaven for giving me the invaluable opportunity of pursuing my PhD at EPFL. I am thankful for his continuous support, scientific guidance, positive attitude, and, not less importantly, for his trust in my abilities and work.

I would like to express my special appreciation to the jury members Prof. Eleni Chatzi, Prof. Fabio Nobile, and Dr. Tommaso Taddei for taking the time to evaluate my work, as well as for their useful suggestions and feedbacks that greatly improved the quality of this thesis. Many thanks to Prof. Kathryn Hess Bellwald, who committed herself as president of the jury.

My sincere gratitude goes to my co-author Dr. Zhenying Zhang for her vision and expertise in designing the sensor placement methodology and for implementing complex support examples. Prof. Francesco Ballarin is thanked for his precious help and patience in the set up and further development of the RBniCS solver. I am grateful for the fruitful scientific discussions with Dr. Niccolò Dal Santo, Prof. Cecilia Pagliantini, Nicolò Ripamonti, and Davide Pradovera, who helped me in the design of the reduced basis solver for PDEs in the frequency domain. I also wish to acknowledge the work of Fabien Rochat and Emeric Sibeude, whose master and semester project results, respectively, are partially included in this thesis.

I am very glad to have shared this journey with great colleagues, who became some of my dearest friends. Thank you Fabian, Luca, Cecilia, Nicolò, Luca, Mariella, Giacomo, and all the people in Mathicse group, past and present.

A special thank goes to the friends I can always count on. Thank you Maria, Francesco, Beatrice, Chiara, Martina, Marta, Clara, Clizia, Filippo, Sara, Laura, Bertrand, Bertrand, Marion, Kilian, Thomas, and Priscille.

I would like to thank my family, Annalisa and Claudia. I feel incredibly lucky to have had a wonderful sister by my side this whole time and a supportive and caring mum not too far.

To Benjamin goes my deepest gratitude. Thank you for always being there for me when I most need it, for you love, and for taking the time to thoroughly proofread this manuscript!

Lausanne, August 25, 2020 Caterina Bigoni

CONTENTS

Abstract (English/Français/Italiano) i

Acknowledgements vii

Contents ix

List of Figures xi

List of Tables xv

1 Introduction 1
- 1.1 Background 3
- 1.2 Objectives 6
- 1.3 Contributions and outline of the thesis 7
- 1.4 Implementation details 9

2 Setting of the study 11
- 2.1 Offline phase 12
- 2.2 Online phase 15

3 Mathematical model for sensor measurements 19
- 3.1 Acoustic-elastic wave equation in the time domain 19
- 3.2 The need for a reduced order model 24
- 3.3 Acoustic-elastic wave equation in the frequency domain 26
  - 3.3.1 Numerical inverse Laplace transforms 29
  - 3.3.2 Weeks method 31
- 3.4 Model Order Reduction 37
  - 3.4.1 Reduced basis method 38
  - 3.4.2 Cotangent lift method 41
- 3.5 Conclusions 45

4 Anomaly detection **47**
4.1 An overview of machine learning techniques for SHM 48
4.2 One-class classification methods 49
4.2.1 One-class Support Vector Machines 51
4.2.2 Isolation Forest 54
4.2.3 Local Outlier Factor 55
4.2.4 Autoencoders 58
4.3 Local semi-supervised method for damage detection and localization . . 62
4.3.1 Feature extraction 65
4.3.2 Dimensionality reduction 67
4.4 Numerical results 69
4.4.1 A training set of healthy configurations 72
4.4.2 A test set of healthy and damaged configurations 74
4.4.3 Classification results 78
4.5 Conclusions 91

5 Sensor placement **93**
5.1 A survey of sensor placement for SHM 93
5.2 Gaussian process regression 96
5.2.1 A short review of GP regression models 97
5.2.2 Sparse GPR models 98
5.2.3 Variational inference of sparse GPR 100
5.3 Variational approximation for systematic sensor placement 102
5.3.1 Constrained variational approximation 104
5.3.2 Including parameter dependence in sensor placement 106
5.3.3 A tool for sensor sensitivity 107
5.4 Numerical results 108
5.4.1 Two-dimensional examples for the guided-wave problem 108
5.4.2 A three-dimensional example for the guided-wave problem 114
5.4.3 Application to a realistic geometry of an offshore jacket 118
5.5 Conclusions 125

6 Conclusions **129**
6.1 Summary and conclusions 129
6.2 Outlook 130

A Sensitivity analysis and parameter estimation **133**

Bibliography **143**

Curriculum Vitae **159**

LIST OF FIGURES

1.1 The collapse of the Morandi bridge in Genoa, in Italy, in August 2018. . . 2

1.2 The Ambarnaya river turned red after the oil leakage from a fuel tank at a power plant in the Siberian city of Norilsk, in Russia, in May 2020. . . . 2

2.1 Flowchart for the offline and online phases of simulation-based SHM procedure for a known sensor network. 17

3.1 Contour plot of the error between the time signals obtained with Newmark's method and the time signals obtained with Weeks method to guide the choice of the values of the hyper-parameters α and b. 36

3.2 Loglog plot of the error between the reconstructed time signals with Weeks method and the Newmark signals for increasing values of N_z. 37

3.3 Comparison of the 2D high-fidelity signals retrieved at the 6^{th} sensor when using Newmark method or the Laplace method with Weeks reconstruction. 38

4.1 Flowchart to compare the feature-level and the decision-level fusion approaches for the semi-supervised damage detection strategy with multi-dimensional training data captured by multiple sensors. 63

4.2 Meshes for the healthy 2D and 3D problems. 69

4.3 Sketch of sensors numbering system and source placement for the 2D problem. 70

4.4 Sketch of sensors numbering system and source placement for the 3D problem. 71

4.5 Source function for various values of the parameter k in time and frequency domain. 73

4.6 2D geometries of nine damage configurations, distinguishing major and minor damages. 75

4.7 Comparison of 2D signals in both time and frequency domain, retrieved at the 9th sensor, obtained either from the healthy structure or from a structure with a major damage in the proximity of the 9th sensor. 77

4.8 Comparison of the second component of 2D reconstructed signals retrieved at the 6th, 8th and 12th, sensors, obtained from the healthy structure or from a structure with a major damage in the proximity of the 9th sensor. 78

4.9 Relative anomaly scores of the training samples obtained using the oc-SVM classifiers, sensor by sensor. 80

4.10 Relative anomaly scores of the test samples obtained using the oc-SVM classifiers, sensor by sensor. 81

4.11 Sketch to summarize the one-class SVMs classification average results on test data for nine 2D damage configurations. The distinction between strong and mild outliers is also shown. 83

4.12 Sketch to summarize the one-class SVMs classification average results on test data for four 3D damage configurations (one undamaged and three damaged). The distinction between strong and mild outliers is also shown. 84

4.13 Sketch to summarize the IF classification average results on test data for nine 2D damage configurations. The distinction between strong and mild outliers is also shown. 88

4.14 Sketch to summarize the LOF classification average results on test data for nine 2D damage configurations. The distinction between strong and mild outliers is also shown. 89

4.15 Sketch to summarize the VAE classification average results on test data for nine 2D damage configurations. The distinction between strong and mild outliers is also shown. 90

5.1 2D examples with different geometries: Problem $1a$ relies on 360 training inputs, corresponding to the vertices of a coarse mesh over the domain, while Problems $1b$ and $1c$ have 286 and 375 training points, respectively. 109

5.2 Example of normalized features, extracted from the solution obtained by solving the acoustic-elastic problem on the geometry $1a$ with a fixed parameter. The N_f corresponding principal components are also shown. 110

5.3 Comparison of the location of inducing points obtained by applying Algorithm 3 for $n_\mu = 100$ samples and the corresponding n_s centroids obtained with Algorithm 4. 112

5.4 Comparison of the centroids obtained with Algorithm 4 for different number of samples n_μ, namely $n_\mu = 10, 40, 70$ and 100. 113

5.5 Comparison of the first three principal components obtained for Problem $1a$ either by extracting the features from the time signals and then performing PCA or by sparse GP reconstruction using $n_s = 4, 9, 16$ or 25 inducing points. 115

5.6 Comparison of the first three principal components obtained for Problem 1b either by extracting the features from the time signals and then performing PCA or by sparse GP reconstruction using $n_s = 4, 9, 16$ or 25 inducing points. 116
5.7 Comparison of the first three principal components obtained for Problem 1c either by extracting the features from the time signals and then performing PCA or by sparse GP reconstruction using $n_s = 4, 9, 16$ or 25 inducing points. 117
5.8 Mean reconstruction errors with error-bars with respect to the number n_s of inducing points for the three principal components used to train the variational sparse GP model for Problems 1a, 1b, and 1c. The corresponding mean reconstruction error, obtained by training a new variational sparse GP model with fixed inducing inputs corresponding to the centroids, is also shown. 118
5.9 Relative variance reduction obtained using n_s centroids and averaged over n_μ samples. 119
5.10 Visualization of the number of sensors belonging to each of the n_s clusters by means of different bubble sizes. 120
5.11 Comparison of centroids obtained using Algorithm 4 and the naive clustering, referred to as equidistant points. 121
5.12 Mean reconstruction errors with error-bars with respect to the number n_s of inducing points for the three quantities of interest jointly used to train the variational sparse GP model. The corresponding mean reconstruction error, obtained by training a variational sparse GP model with fixed inducing inputs corresponding to the centroids is also shown together with the one where the fixed inducing inputs are the naive centroids 122
5.13 3D geometry of a T-beam with 4688 training points. 122
5.14 Comparison of the location of inducing points obtained by applying Algorithm 3 for $n_\mu = 10$ samples and the corresponding n_s centroids obtained with Algorithm 4 for the 3D problem. 123
5.15 Comparison of the first two principal components obtained either by extracting the features from the time signals and then performing PCA or by sparse GP reconstruction using $n_s = 4, 16$, or 36 inducing points for the 3D problem. 124
5.16 Relative variance reduction obtained using n_s centroids and averaged over n_μ samples for the 3D problem. 125
5.17 Jacket model: wind loads applied on components in the square. 126
5.18 Jacket model: components chosen for sensor placement. 127

A.1 Joint posterior density approximations of the parameters characterizing the wave speed. 135
A.2 2D geometry with $d_\Omega = 2$ subdomains. 137
A.3 2D geometry with $d_\Omega = 21$ subdomains. 138

A.4 DGSM indices with respect to $d_\mu = 5$ parameters for the first component of the sensor signals $\hat{\mathbf{g}}_i(\boldsymbol{\mu})$ for $i = 1, \dots, 6, 10, \dots, 15$. 139
A.5 Parametric dependency of the signals at sensors 5 and 11 with respect to different values of the $d_\mu = 5$ the input parameters. 140
A.6 DGSM indices with respect to $d_\mu = 43$ parameters for the first component of the sensor signals $\hat{\mathbf{g}}_i(\boldsymbol{\mu})$ for $i = 4, 5, 6, 10, 11, 12$. 141

LIST OF TABLES

4.1 Summary of the datasets used to train different one-class algorithms. . . 68
4.2 Summary of the parameters used to generate the synthetic 2D and 3D sensor measurements. 73
4.3 Comparison of crest factor **c** and arrival time **a** for high-fidelity reconstructed 2D signals at four different sensor locations for the healthy structure and a structure with a major damage 76
4.4 Summary of the parameters used to generate the synthetic train and test datasets in 2D and 3D, sensor by sensor, for four different one-class algorithms. 78
4.5 Fractions of test samples classified as outliers using oc-SVMs for the 2D problem with both healthy and damaged configurations. 82
4.6 Fractions of test samples classified as outliers using oc-SVMs for the 3D problem with both healthy and damaged configurations. 85
4.7 Fractions of test samples classified as outliers using IFs for the 2D problem with both healthy and damaged configurations. 87
4.8 Fractions of test samples classified as outliers using the LOF algorithm for the 2D problem with both healthy and damaged configurations. 87
4.9 Fractions of test samples classified as outliers using VAE for the 2D problem with both healthy and damaged configurations. 87

5.1 Sensor-wise percentages of accuracy for undamaged, minor damaged and major damaged scenarios. 127

CHAPTER 1

INTRODUCTION

Unexpected bridge collapses, offshore oil platforms disasters, or aerospace structural failures can have tremendous consequences in terms of fatalities, economic loss, and environmental pollution. Many existing private and public assets, such as civil engineering infrastructures, buildings, or aircrafts, require reliable damage detection techniques to be safely used, especially during their inevitable ageing. When monitoring a structure over time, its deterioration and damages represent a great concern and the early detection of critical decay might prevent failures that can cause sudden shutdowns or even catastrophes with severe life-safety and economic repercussions.

The collapse of the Morandi bridge in Genoa, in Italy, which killed 43 people in August 2018 (see Figure 1.1), or the oil spill into the Ambarnaya river in Siberia, caused by a damaged storage tank at the Nornickel power plant in May 2020 (see Figure 1.2), are two recent examples for which the implementation of a monitoring system could have played a crucial role in forecasting and preventing these accidents. A study of the collapse of the Morandi bridge, presented in [Mor19], identifies two principal aspects that may have affected the structural response: (*i*) the ageing and deterioration of the structural materials, and (*ii*) the increased traffic load due to the evolution of transport in general and to the development of the port of Genoa, both of which were probably underestimated at the time of the structural design in the 1960s. For the environmental disaster near the city of Norilsk, in the north of Russia, it is believed that the thawing of permafrost, as a result of climate change, caused the power plant reservoir to collapse and spill approximately 20'000 tonnes of diesel fuel, which stained vast portions of the Ambarnaya river red, threatening the Artic ecosystem[1]. The ability to monitor the development and the evolution of cracks and other deterioration indices motivates the introduction

---

[1] https://www.bbc.com/news/world-europe-52915807. BBC News: Arctic Circle oil spill prompts Putin to declare state of emergency.

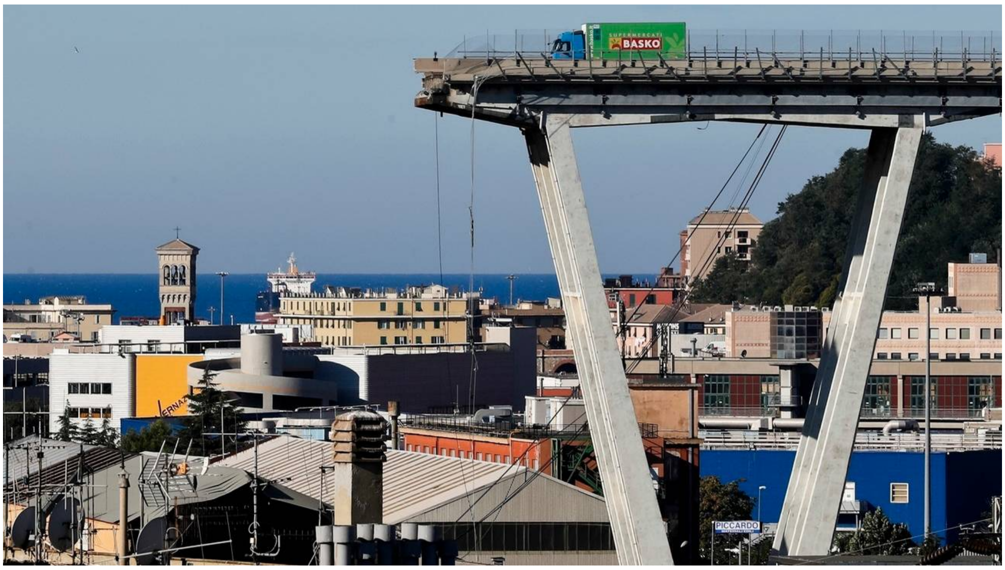


Figure 1.1 – The collapse of the Morandi bridge in Genoa, in Italy, in August 2018.

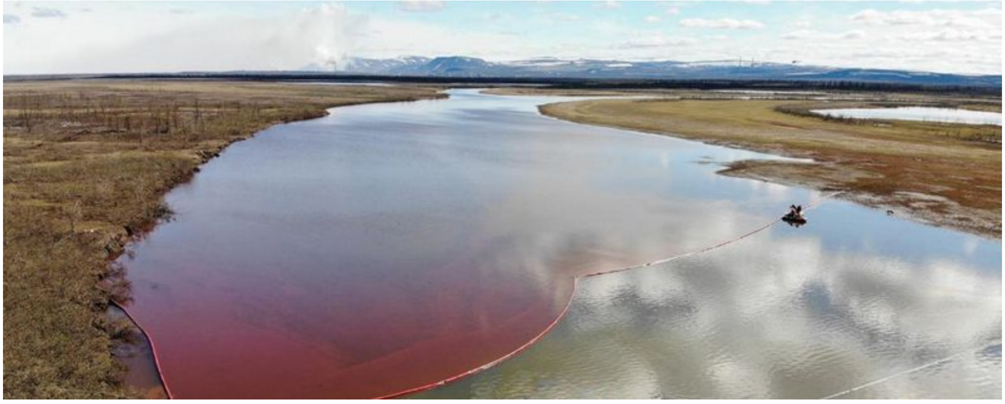


Figure 1.2 – The Ambarnaya river turned bright red after the oil leakage from a fuel tank at a power plant in the Siberian city of Norilsk, in Russia, in May 2020[2].

of structural health monitoring (SHM) systems. SHM refers to automated monitoring procedures that seek to provide reliable information on the performance and integrity of a structure in real time. It has become the subject of recent studies with applications not only in civil engineering and the aerospace industry, but also in the conservation of cultural heritage structures, see, e.g., [UCC+17; CDdMU17].

In the last decades, economic and life-safety advantages promised by SHM techniques have motivated the transition from traditional *run-to-failure* and *time-based maintenance* to *condition-based maintenance* strategies to detect unexpected failures [FW12]. In the run-to-failure approach, sometimes called breakdown-based maintenance, the system is operated until some critical component fails and is replaced. Time-based maintenance,

---

[2] ©Marine Rescue Service/AFP.

while being a more proactive approach than run-to-failure, replaces components at fixed time intervals regardless of their damage state. Often economic savings could be made by exploiting the component longer. Indeed, a structure may continue to perform its function even if small damages are present, and a trade-off between economic and safety impact should be taken into account when defining a maintenance strategy of a system. In the context of SHM, the combination of sensor measurements, numerical models simulating the underlying behavior of a structure of interest under different environmental and operational conditions, and machine learning techniques has led to the design of structural *digital twins* [WWBG20]. These accurate virtual representations of complex assets, which are equipped with cutting-edge sensor technology, replace costly and complex visual inspections. Indeed, the direct evaluation of structural damages often entails the shutdown of the installation; moreover, the location of damages may be difficult, if not impossible, to observe directly. The evolution to digital inspections allow for a significant improvement in the predictive capability of the structural state of damage. Ultimately digital twins seek to provide a unified framework to support an automated decision-making strategy in real-time.

Because the design of SHM technologies benefits from the interplay of numerous fields of science and engineering, a synergistic, multidisciplinary approach is needed to efficiently deal with this topic. We refer to [FW12], where SHM systems are presented by highlighting the combination of models of structural dynamics, signal processing, detection theory, machine learning, probability and statistics, and sensor networks. In this thesis, we present an SHM approach blending numerical modeling and data-driven techniques.

1.1 Background

Given a structure of interest, the principal objective of SHM is to provide an answer to the question *is damage present?* Additionally, the damage state of a system can be assessed using a hierarchical identification process, aiming at answering questions regarding the location, the type, the extent, and the prognosis of damage. Damage is here considered as a change to the material and/or geometrical property of a system with respect to a prior healthy state. Machine learning techniques provide a way for automated recognition of patterns and regularities in data with the possibility of assigning a label to a state observation. For this reason, pattern recognition and machine learning algorithms become an essential building block for efficient condition-based monitoring techniques.

In SHM, there are two principal approaches, namely the *inverse-problem* or *model-based* approach and the *data-based* approach. Both processes fully exploit an *offline-online decomposition* of tasks. In a model-based methodology, the goal is to estimate the parameters that minimize the difference between the model response, built and calibrated during the offline phase, and new sensor measurements, acquired online from the structure.

For every new measured data, the model is updated and the new model solution is used to assess the state of damage of the system. Such an inverse-problem approach is often ill-posed and requires many online PDE solves, which is not suitable for real-time damage assessment [FW12]. In a data-based approach, the offline phase consists in building a database of signals which represent the behavior of the structure of interest under different conditions. During the online phase, real experimental time-signals, collected from sensors placed on a structure, are compared with those simulated offline using machine learning techniques, i.e., a classifier is used to discriminate between damaged and undamaged states.

It is important to observe that the training data used in the data-based approach are not necessarily only experimental measurements collected from sensors placed on the structure of interest or similar structures. Instead, the information carried in physics-based models can be leveraged to obtain a richer and more reliable estimate of the structural behavior by building a parametric mathematical model of the monitored structure. Thanks to efficient numerical models, many more baseline scenarios, representing the natural variations of the structure, can be included in the dataset. The framework that combines data-based procedures with synthetic data is often referred to as a *simulation-based* approach and this is what we employ in this thesis. This line of research, often associated with surrogate modeling approaches, has gained popularity in the last ten years, see, e.g., [LAW15; TPYP18; QLAC+19; ABC19; KKW20; RMMC20; BWW20] and references therein.

Data-driven SHM is a very broad topic and has been studied from many different points of view, especially in the civil engineering and aerospace communities. SHM monitoring approaches are often classified in two categories: *wave propagation* approaches and *vibration-based* approaches. This separation is connected with the type of excitation system employed. Active sources refer to a system where a known input is used to excite the structure to enhance the damage detection process, while passive systems are based on ambient excitation. The former methods aim to detect damages by examining the distortions in propagating elastic waves as a result of reflections and amplitude attenuations when intersecting the damage boundary. The goal of vibration-based SHM approaches instead is to identify changes in the natural frequencies by analyzing the principal modes. Traditionally, wave-based monitoring approaches, based on piezoelectric materials, are popular in the aerospace industry or in piping systems to detect damages in complex structural components, while vibration-based SHM studies are often used in the context of large structures, where ambient excitation is often the only practical way to excite the global dynamic response of the structure, e.g., wind or wave motion for an offshore platform, and vehicular or pedestrian traffic for a bridge structure. Nevertheless, relying on a known and controlled source results in a more reliable method for failure detection compared to passive sensing systems, for which a robust data normalization procedure is needed to distinguish the dynamic response between damage and changes in the operational and environmental conditions. In this thesis we rely on the parametric acoustic-elastic wave equation to design a monitoring approach based on guided-waves and relying on a local excitation. The guided-wave approach belongs to the so-called *non-destructive evaluation* technologies in the SHM context and they are typically regarded as a more offline approach, i.e., pertaining to targeted periodic inspections. Nonetheless, the generic damage detection framework developed in this thesis could be also extended to other types of monitoring technologies, including the prominent vibration-based approach, provided that the governing equation and source of excitation are suitably changed. We highlight the works related to diagnostics Lamb waves and wavelet transforms, which are often integrated with piezoelectric sensors/actuators, see, e.g., [IC04; LY01; GPC17; SM09]. This line of work focuses primarily on diagnostic signal generation and signal processing and it aims at measuring the changes 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 [DFPS96; KRCS03; OCDS17], are worth mentioning.

Despite the numerous works related to structural damage identification, only few combine machine learning techniques with numerical simulations. In [ZA96], 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. An efficient strategy to The performance is validated on experimental data of specimens containing similarly sized cracks. Similarly, in [LHSL02] 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. The estimation of fatigue degradation of composite wind turbine blades and the condition monitoring of wind-farms, where the joint effect of multiple turbines potentially contributes in a detrimental manner to fatigue life, is addressed in [MAC17; MAC20]. Here, the authors combine surrogate modeling with variational autoencoders to quantify levels of statistical deviation on condition monitoring data. Following a common strategy in SHM, in [MAC20], the deviation in terms of probabilistic distance from the data-driven model describing the dynamics of the healthy system serves as outlier detection. Aerospace applications are presented in [LAW15], where real time sensor information is compared to simulation data from precomputed damaged scenarios to update the estimates of vehicle capabilities using a Bayesian classification process. In [TPYP18], the authors propose a simulation-based procedure to map measured data to the relevant diagnostic class by comparing the performances of different supervised 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.

1.2 Objectives

The objective of this thesis is to develop a mathematical and computational framework to monitor a structure of interest in the absence of known damaged configurations. In particular, the prediction of the damage state is here obtained by applying pattern recognition methods to a synthetic dataset obtained by simulating the propagation of acoustic-elastic waves in the structure of interest under the effect of a known excitation. The techniques developed in this thesis aim to design an all-round framework to address the following three research questions:

- **How to efficiently generate a synthetic database of sensor signals that mimic the measurements acquired from a real structure while taking into account the baseline variations.**

  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, composite plates, or pipes. To simulate *natural variations* of a structure, i.e., baseline environmental conditions and standard operation, a reduced order model is required to overcome the computational burden associated with a many query problem.

- **How to detect, localize, and quantify the severity of damages in the absence of anomalous data.**

  While novelty detection is popular in the structural damage identification community, see, e.g., [LB14; DSC07; AKM$^+$17], it has, to the authors' knowledge, never been studied when combined with synthetic datasets. The goal is to compare the classification results of different anomaly detection strategies trained on a dataset composed solely of healthy samples. Depending on the machine learning algorithm, different feature extraction and feature selection strategies can be investigated.

- **How to systematically place a network of sensors in the absence of anomalous data.**

  Even though there exists a variety of SHM techniques, mainly differing in the quantity of interest to estimate or in the type of sensors employed, they all rely on a network of sensors, which are constrained by various requirements and limitations (cost, accessibility, etc.). The correct positioning of the sensors can have a drastic impact on the monitoring efficiency [OSM19]. Existing sensor placement strategies are designed under the assumption that a certain degree of information is available on the damage type and location. As such, existing strategies are damage-dependent and do not generalize well to a new type or new location of damages. We aim at providing a method based on healthy configurations only.

These objectives offer the need to combine different techniques coming from scientific computing, numerical analysis, and machine learning to provide new approaches for the solution of simulation-based anomaly detection in the context of structural health monitoring.

1.3 Contributions and outline of the thesis

The main contributions of this thesis 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 *Laplace domain* and reconstruct the time-signals by using Weeks method, a numerical inverse Laplace transform. The latter allows us to recover information of the transient phase, which is a key feature for the classification step.
- Since machine learning algorithms are well-known to behave better when using a large dataset [Bis06], collecting a synthetic database requires a *model order reduction* (MOR) approach to overcome the computational burden involved in the repeated solution of pPDEs. As employed in other works of simulation-based SHM, see, e.g., [LAW15; TPYP18; QLAC$^+$19; KKW20], we use the Reduced Basis (RB) method, a projection-based method, based on reconstructing 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 introduce 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 evaluate an anomaly score function to see if the new measurement conforms with the generated dataset. This allows a binary classification: it either is an inlier, i.e., it belongs to the cluster of previously considered healthy signals, or it is an outlier. This corresponds to a *semi-supervised learning* approach, also called *one-class classification* method, where labeled data, belonging to the “normal” class, are used in the training phase and unlabeled data from both classes are used in the test phase to identify abnormal data which deviate from the normal model [PCCT14; GU16]. With one-class algorithms it is possible to *locate the damage* by training a different classifier for each sensor, using the measurements collected at each sensor, see, e.g., [LB14]. Additionally, by relying on a real valued anomaly score function, we can estimate

the severity of the damage. We test the performance of several one-class classifiers on 2D and 3D synthetic datasets.

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 labeled 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 RB 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 not differentiate 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 [GU16]. Since in a simulation-based approach labels of generated data are always available, the unsupervised learning technique is not suitable.

- We integrate the synthetic dataset of healthy configurations with a modified variational sparse Gaussian process (GP) method to systematically place a fixed number of sensors on a structure of interest. The variational sparse Gaussian process method offers a tool to identify a predefined number of spacial positions which minimizes the reconstruction error of an output of interest at all *unsensed* locations. The feature-based dataset, used for anomaly detection, is chosen here as output of interest, thus providing an appropriate indicator of the damage detection performance of a given network.
- We present 2D and 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.

This thesis is organized as follows:

Chapter 2 provides an overview of the general data-driven approach used in this thesis 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. Our systematic sensor placement strategy is also introduced. By setting the notation, we provide the basis and the connections to develop the details in the following chapters.

Chapter 3 develops the mathematical details to construct the synthetic database by emphasizing the important role of MOR to avoid the computational burden associated with the repeated solution of large systems of equations. In particular, we apply a projection-based RB method to the acoustic-elastic wave equation in Laplace domain and we provide its reconstruction in the time domain using Weeks method.

In **Chapter 4**, after motivating our choice of using a decision-level fusion approach to train multiple classifiers, we illustrate various semi-supervised one-class classification strategies. We present dimensionality reduction methods to avoid the curse of dimensionality by means of ad-hoc feature extraction and data compression. We conclude by comparing the classification results obtained with four one-class classification methods (one-class Support Vector Machines, Isolation Forests, Local Outlier Factor, and Variational Autoencoders) and highlight their strengths in detecting and localizing outliers on noisy synthetic test samples. We note that the numerical results obtained with one-class Support Vector Machines for 2D and 3D geometries are published in [BH20], while the results obtained with Isolation Forests and Local Outlier Factor for the 2D geometry are an extension to those presented in Emeric Sibieude's semester project [SBH19], and the preliminary results obtained with Variational Autoencoders for the 2D geometry are a joint work with Dr. Zhenying Zhang.

Chapter 5 presents a systematic sensor placement strategy which only requires healthy samples. After introducing the general setting of GP and sparse GP, we highlight how variational sparse GP can be used as tool for sensor placement. We propose different indicators to quantify the quality of the placement and provide numerical examples, including 2D and 3D examples, based on a wave-based monitoring approach, and an example of a real engineering application using a static monitoring approach.

Finally, conclusions and future developments are offered in **Chapter 6**.

This thesis contains results which have already been published, see, [BH20; BZH20].

1.4 Implementation details

The numerical simulations carried out in this thesis were performed using the open source library computing platform for solving PDEs `FEniCS` [LMW12; ABH$^+$15] with the Python interface. For the implementation of reduced order models for parametrized PDE we relied on the open source library `RBniCS` [BSR16; HRS15], which we extended by implementing the cotangent lift method [PM16], explained in Section 3.4.2. The offshore jacket example in Section 5.4.3 is obtained using the SCRBE solver from Akselos [sof20]. The numerical inverse Laplace reconstruction by means of Weeks method is implemented with ad hoc functions in Python.

Both Matlab [MAT19] and Python were used for the machine learning algorithms and data management. In particular we used Matlab built-in functions for the one-class classification using SVMs, while specific open source libraries in Python were used for the other semi-supervised strategies, i.e., Isolation Forests, Local Outlier Factor, and LSTM Variational Autoencoders. For the former two, we relied on the `scikit-learn` library [PVG$^+$11] and for the latter we used the `Keras` API [C$^+$15], which runs on the platform TensorFlow. The Gaussian process open-source framework `GPy` [GPy12] was used to implement the sensor placement strategy in Python. Related to this, we used the DEAP (Distributed Evolutionary Algorithms in Python) [FRG$^+$12] to implement the genetic algorithm for the optimization of the hyper-parameters and, to cluster the inducing points, we used the built-in Matlab function `kmedoids`.

CHAPTER 2

SETTING OF THE STUDY

The content presented in this chapter is partially based on Section 2 of [BH20].

In this introductory chapter we present the general data-driven approach for simulation-based structural anomaly detection in the absence of faulty data. The different stages of the framework presented here will be described in greater details in the following chapters. Here, we highlight the decomposition of tasks into two phases: in Section 2.1 we discuss the expensive offline phase where we simulate the response of healthy structures under different environmental and operational conditions, followed by the training of as many classifiers as the number of available sensors. The classifiers are then used for the rapid online testing of new (experimental) sensor responses, as described in Section 2.2. The training dataset corresponds to synthetic sensor measurements under the assumption that the sensors are already placed on the structure of interest and their location is therefore known. When the sensor network configuration is unknown, i.e., the sensors still have to be placed on the structure, we face the problem of how to choose their best location. For this, as part of the offline phase, we introduce a general sensor placement strategy to systematically find the sensor locations which best cover the monitoring area.

Pattern recognition approaches in SHM make use of machine learning algorithms to detect damages by comparing a new configuration with a database of previously observed healthy configurations [FW12]. Such a database can be assembled either by using experimental data from the structure or similar structures, or by performing *synthetic experiments* based on a parametrized model which approximates the structural dynamics under the effect of a source. 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. Here, we generate synthetic sensors measurements from *healthy* *structures* only, without the ambition of representing all possible system configurations including failure modes. Indeed, our goal is to capture the baseline, uncertain operational and environmental conditions to create a robust database of signals reflecting the healthy behaviors that the structure may undergo during its life time. The parameters that express such variations are physical and are typically related to the material properties, the boundary or initial conditions, or the external excitements. Geometric parameterizations are not included in this thesis 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 this thesis, we consider the relevant parameters to be characterized by a probability distribution. For the sake of simplicity, but without loss of generality, we assume that the probability distributions of these parameters are known a priori, e.g., provided by engineering experience, and, in particular, only uniform distributions are considered. However, a Bayesian inference approach may be used to update the prior knowledge on the probability distribution of the input parameters using available sensor data. Moreover, the employment of sensitivity analysis tools may help to compress the parameter space and thus retain only the parameters that influence the output of interest the most. While these topics go beyond the scope of this thesis, a brief introduction to the Bayesian inference approach and sensitivity analysis in the context of wave equations is given in Appendix A together with preliminary results.

2.1 Offline phase

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 $\mathcal{P} \subset \mathbb{R}^{d_\mu}$ the parameter space with $d_\mu \in \mathbb{N}_+$ being the number of parameters used to characterize the model. Let $\boldsymbol{x}_i \in \Omega$ for $i = 1, \ldots, n_s$ represent an approximation to the position of each of the n_s sensors attached to the structure. Given a generic parametric model with suitable boundary and initial conditions, for a given d_μ-dimensional vector $\boldsymbol{\mu} \in \mathcal{P}$, we seek the vector-valued solution $\boldsymbol{u} \coloneqq \boldsymbol{u}(\boldsymbol{x}, t; \boldsymbol{\mu}) : \Omega \times [0, T] \times \mathcal{P} \to \mathbb{R}^d$ such that

$$\frac{\partial^2 \boldsymbol{u}}{\partial t^2} + \mathcal{L}^{\text{damp}} \left[\frac{\partial \boldsymbol{u}}{\partial t}; \boldsymbol{\mu} \right] + \mathcal{L} \left[\boldsymbol{u}; \boldsymbol{\mu} \right] = h(t; \boldsymbol{\mu}) s(\boldsymbol{x}; \boldsymbol{\mu}) \tag{2.1}$$

and evaluate a relevant output of interest

$$g_i(t; \boldsymbol{\mu}) := \ell(\boldsymbol{u}(\boldsymbol{x}_i, t; \boldsymbol{\mu}); \boldsymbol{\mu}), \qquad \text{for } i = 1, \ldots, n_s, \text{ and } t \in [0, T], \tag{2.2}$$

where the possibly parameter-dependent output functional $\ell : \mathbb{R}^d \times \mathcal{P} \to \mathbb{R}^{d_\ell}$ maps the time-signals, evaluated at sensor locations $\boldsymbol{x}_i \in \Omega$ and time $t \in [0, T]$, into d_ℓ-dimensional vectors that emulate the real sensor measurements, e.g., local displacements, accelerations, or strains. In (2.1), $\mathcal{L}^{\text{damp}}[\cdot, \boldsymbol{\mu}]$ 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., [TPYP18; ZBA13]. In this framework, the time-dependent experimental sensor measurements $g_i^{\mathrm{exp}}(t) : \mathbb{R} \to \mathbb{R}^{d_\ell}$ are given by

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

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 (2.2) for multiple values of the input parameters $\boldsymbol{\mu} \in \mathcal{P}$, i.e. we consider set of N_{tr} parameters

$$\boldsymbol{\Xi}^{N_{tr}} := \{\boldsymbol{\mu}_m\}_{m=1}^{N_{tr}}, \tag{2.3}$$

which we assume to be a good representation of all the natural variations of healthy configurations under normal behavior.

The numerical solutions, obtained by solving the discrete counterpart of (2.1) N_{tr} times, once for each parameter in $\boldsymbol{\Xi}^{N_{tr}}$, are evaluated at the sensor locations to obtain the outputs of interest (2.2). Assuming the interval $[0,T]$ is partitioned into N_t equal subintervals, the discrete time-signals are obtained by evaluating the output of interest (2.2) at time $t^n := n\frac{T}{N_t}$ for $n = 0,\dots,N_t$, i.e.,

$$\mathbf{g}_i^m := [g_i(t^0;\boldsymbol{\mu}_m), g_i(t^1;\boldsymbol{\mu}_m), \dots, g_i(t^{N_t};\boldsymbol{\mu}_m)], \tag{2.4}$$

for $i = 1,\dots,n_s$ and $m = 1,\dots,N_{tr}$. We observe that $\mathbf{g}_i^m \in \mathbb{R}^{d_\ell \times (N_t+1)}$ and, in the following, we use the interchangeable notation $\mathbf{g}_i^m = \mathbf{g}_i(\boldsymbol{\mu}_m)$. The synthetic datasets of dimension N_{tr} are defined as the collection of these time signals, i.e.,

$$\mathcal{D}_i := \{\mathbf{g}_i^m\}_{m=1}^{N_{tr}}, \quad \text{for } i = 1,\dots,n_s. \tag{2.5}$$

We observe that large datasets are typically necessary to build robust classifiers in a machine learning context, see, e.g., [Mur12]. Usually, (2.1) is discretized in space by means of the Finite Element method, and in this framework the many-query problem implies the repeated solution of large linear systems. To overcome the computational burden, we introduce reduced order modeling (ROM) techniques to exploit the parametric dependency and solve for each new parameter a system of reduced dimension. The details of the numerical scheme and the ROM strategies used in this thesis are presented in Chapter 3.

The second part of the offline phase consists in training n_s separate *one-class classifiers*[1],

---

[1] As mentioned in Section 1.3 and as it will be further explained in Chapter 4, the separate training of multiple classifiers allows to gain information on the *local* state of damage.

i.e., one per database, of synthetic healthy signals (2.5). Since fitting high dimensional training data is well known to be computationally expensive and is also prone to overfitting due to high complexity, we propose to extract Q engineering-based features from the discrete time signals (2.4). More specifically, we apply Q ad-hoc feature functions $\mathcal{F}_j : \mathbb{R}^{d_\ell \times (N_t+1)} \to \mathbb{R}^{d_\ell}$ for $j = 1, \dots, Q$, which are assumed to be damage-sensitive indicators, to each output of interests (2.4). The N_{tr}-dimensional feature-based databases of synthetic signals at location $\boldsymbol{x}_i$ are defined as

$$\mathcal{D}_i^{\mathcal{F}_Q} := \{[\mathcal{F}_1(\mathbf{g}_i^m), \dots, \mathcal{F}_Q(\mathbf{g}_i^m)]\}_{m=1}^{N_{tr}}, \quad \text{for } i = 1, \dots, n_s. \tag{2.6}$$

Since the features may be correlated and this can lead to overfitting issues, we propose to further compress the dimension, i.e., we consider a feature-based database of reduced dimension

$$\mathcal{D}_i^{\mathcal{F}} := \{\mathbf{y}_i^m\}_{m=1}^{N_{tr}}, \quad \text{for } i = 1, \dots, n_s,$$

where $\mathbf{y}_i^m \in \mathbb{R}^{d_y}$, with $d_y < d_\ell Q$, may be obtained by feature selection, principal component analysis, or other strategies that capture the variability of the original data. Then, we rely on a semi-supervised learning algorithm for anomaly detection, which takes as input a dataset $\mathcal{D}_i^{\mathcal{F}}$ and returns an anomaly score function $s_i : \mathbb{R}^{d_y} \to \mathbb{R}$ and a threshold value $\theta_i \in \mathbb{R}$, i.e.,

$$[s_i, \theta_i] := \texttt{oc-ML}\left(\mathcal{D}_i^{\mathcal{F}}\right), \quad \text{for } i = 1, \dots, n_s, \tag{2.7}$$

where `oc-ML` represents the generic steps of a one-class machine learning technique. As the training set is a collection of healthy samples, we expect $s_i(\mathbf{y}_i^m) < \theta_i$ for most of the training samples $\mathbf{y}_i^m$ for $i = 1, \dots, n_s$. Various anomaly detection strategies including feature extraction and data compression are described in Chapter 4.

So far we have discussed the situation in which the sensor locations are known a priori. However, a central problem related to SHM is the identification of the *optimal* sensor locations. Sensor placement strategies are dictated by dual antagonist objectives, i.e., detect the damage and avoid false diagnosis on one hand and keep the sensor budget low on the other hand. Optimal sensor placement represents a challenging task, especially when we make no assumptions on the type or location of damages. In this thesis, we assume that a collection of $n_{\text{dof}} \gg n_s$ feature-based output of interest $\mathbf{Y}(\boldsymbol{\mu}_m) := [\mathbf{y}_i^m, \dots, \mathbf{y}_{n_{\text{dof}}}^m]$, computed at input locations $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{dof}}}]$, can be approximated by a Gaussian process. Here we choose $\mathbf{x}_i \in \Omega$ to be the vertices of a sufficiently coarse mesh defined over Ω for $i = 1, \dots, n_{\text{dof}}$. We then rely on an adaptation of the variational sparse Gaussian process method [Tit09] to define a systematic placement method for a fixed budget of n_s sensors. The procedure, indicated here by $\texttt{vsGP}(\cdot, \cdot; n_s)$, is summarized as

$$\mathbf{Z}_{\text{opt}}^m := \texttt{vsGP}\left(\mathbf{X}, \mathbf{Y}(\boldsymbol{\mu}_m); n_s\right), \tag{2.8}$$

where $\mathbf{Z}_{\text{opt}}(\boldsymbol{\mu}_m)$ is the set of n_s optimal locations the so-called *inducing points*, associated with $\mathbf{Y}(\boldsymbol{\mu}_m)$. We note that the points $\mathbf{z}_j(\boldsymbol{\mu}_m)$ of $\mathbf{Z}_{\text{opt}}(\boldsymbol{\mu}_m)$ for $j = 1, \ldots, n_s$ are obtained by constrained optimization over an admissible domain of interest Ω_s, usually assumed to be a portion of the initial domain Ω, i.e., $\Omega_s \subset \Omega$.

To take into account the parametric variations, we compute the inducing points for n_μ different input parameters and thus obtain n_μ collections of n_s points $\mathbf{Z}_{\text{opt}}^m$ for $m = 1, \ldots, n_\mu$. Finally, unsupervised clustering techniques are used to identify the desired n_s sensor locations with the centroids of the n_s clusters. This method is described in detail in Chapter 5.

2.2 Online phase

During the online phase, the classifiers are used to detect possible anomalies in new sensor data, which are referred to as test data. The classifier will be able to distinguish data generated from an undamaged structure from the data generated from a damaged one. Given a new datum

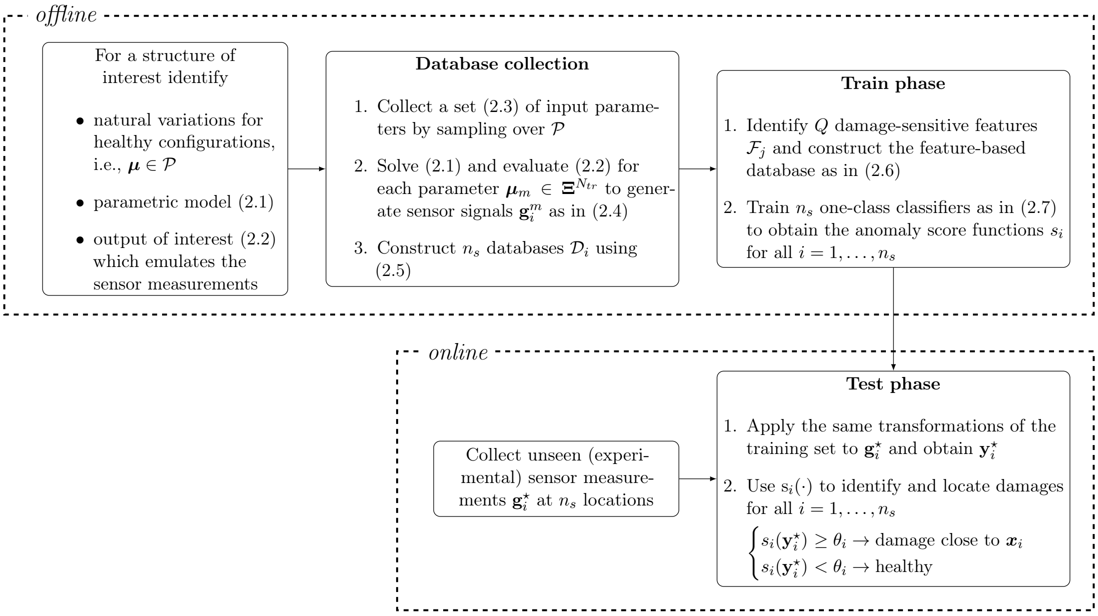
$$\mathbf{g}_i^\star := [g_i^{exp}(t_0), \ldots, g_i^{exp}(t_{N_t})],$$

let $\mathbf{y}_i^\star \in \mathbb{R}^{d_y}$ be its corresponding feature-based vector, i.e., the transformed datum after feature extraction and data compression. Then $\mathbf{g}_i^\star$ will be classified as an outlier if $s_i(\mathbf{y}_i^\star) \geq \theta_i$ and as an inlier otherwise. More precisely, by looking at which sensor signal $\mathbf{g}_i^\star$ among the n_s measurements is classified as an outlier, 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 $s_i(\mathbf{y}_i^\star)$ provides information about the uncertainty of belonging to one of the two classes: higher values correspond to a higher confidence on the output.

In this thesis, experimental sensor measurements are replaced with noisy simulated data. An healthy test sample is generated similarly to the samples of the training phase, i.e., it corresponds to the solution obtained using a new, unseen parameter belonging to the initial parameter space $\boldsymbol{\mu}^\star \in \mathcal{P}$. To avoid the so-called inverse crime, associated to using the very same model for both the training and the test simulated data, we propose to generate the training data with a low-fidelity solver, while the test data are generated using a high-fidelity solver, as it will be further explained in Chapters 3 and 4. We enforce an additional Gaussian noise to the high-fidelity noisy signals to further differentiate healthy training data from healthy test data, aiming to make the test data even more resemblant to experimental data. The response of damaged structures is simulated by replacing the domain used to generate the healthy database with a variety of faulty domains, i.e., we modify the domain Ω to include cracks of different sizes and located at different positions. Ultimately, a damaged test sample is generated by solving the PDE on the modified domain for a new parameter $\boldsymbol{\mu}^\star \in \mathcal{P}$ and by further evaluating the output of interest. Under the assumption that $\mathcal{P}$ identifies all baseline healthy conditions, damaged scenarios may be also obtained by simulating the sensor response using an input parameter outside $\mathcal{P}$, without the need of modifying the initial domain. For example, a reduction in the Young's modulus is often associated with a damaged condition, as described in [DFPS96] and references therein. Therefore two different parameter ranges could be used to simulate healthy and damaged scenarios, without modifying the geometrical domain. However, in this thesis, only geometrical flaws are considered.

Let $\mathbf{y}_i^\star$ be the vector representing features of reduced dimension, extracted from the noisy signal $\mathbf{g}_i(\boldsymbol{\mu}^\star) + \varepsilon_i$, with $\boldsymbol{\mu}^\star \in \mathcal{P}$. If $\mathbf{g}_i(\boldsymbol{\mu}^\star)$ is obtained by simulating the sensor response over the initial baseline domain Ω, we expect $s_i(\mathbf{y}_i^\star) < \theta_i$ for all $i = 1, \ldots, n_s$, provided that the variance γ_i^2 of the additional noise is sufficiently small. If instead $\mathbf{g}_i(\boldsymbol{\mu}^\star)$ is obtained by simulating the sensor response over a damaged domain, we expect $s_i(\mathbf{y}_i^\star)$ to be greater than or equal to θ_i, provided that the damaged domain presents a crack close to the i-th sensor. Otherwise, we expect $s_i(\mathbf{y}_i^\star)$ to be smaller than θ_i.

The flowchart in Figure 2.1 gives an overview of the data-driven one-class classification problem with synthetic data for a fixed sensor network and highlights the separation of the offline and online phases. To include the systematic sensor placement strategy, we should split point 2 of the database collection in two phases: a preliminary step in which we solve (2.1) and evaluate (2.2) for $n_{\text{dof}} \gg n_s$ input locations and use (2.8) to find the n_s sensor locations and a second step which computes the desired discrete time signals (2.4) only for $i = 1, \ldots, n_s$.



For a structure of interest identify

- natural variations for healthy configurations, i.e., $\boldsymbol{\mu} \in \mathcal{P}$
- parametric model (2.1)
- output of interest (2.2) which emulates the sensor measurements

1. Collect a set (2.3) of input parameters by sampling over $\mathcal{P}$
2. Solve (2.1) and evaluate (2.2) for each parameter $\boldsymbol{\mu}_m \in \boldsymbol{\Xi}^{N_{tr}}$ to generate sensor signals $\mathbf{g}_i^m$ as in (2.4)
3. Construct n_s databases $\mathcal{D}_i$ using (2.5)

1. Identify Q damage-sensitive features $\mathcal{F}_j$ and construct the feature-based database as in (2.6)
2. Train n_s one-class classifiers as in (2.7) to obtain the anomaly score functions s_i for all $i = 1, \ldots, n_s$

Collect unseen (experimental) sensor measurements $\mathbf{g}_i^\star$ at n_s locations

1. Apply the same transformations of the training set to $\mathbf{g}_i^\star$ and obtain $\mathbf{y}_i^\star$
2. Use $s_i(\cdot)$ to identify and locate damages for all $i = 1, \ldots, n_s$

$$\begin{cases} s_i(\mathbf{y}_i^\star) \geq \theta_i \rightarrow \text{damage close to } \boldsymbol{x}_i \\ s_i(\mathbf{y}_i^\star) < \theta_i \rightarrow \text{healthy} \end{cases}$$

Figure 2.1 – Flowchart for the offline and online phases of simulation-based SHM procedure for a known sensor network.

CHAPTER 3

MATHEMATICAL MODEL FOR SENSOR MEASUREMENTS

The content presented in this chapter is partially based on Sections 3 and 5 of [BH20].

In this chapter, we present the mathematical model used to approximate the physical phenomena describing the structural vibrations measured at a specific sensor location under the effect of an active source. In particular, the details of the numerical approximation of the parametric acoustic-elastic PDE with a known input excitation in time domain are given in Section 3.1. Considering the many-query scenario associated with the need to construct a database of synthetic sensor measurements, which are representative of the structure of interest under various environmental and operation conditions, we propose a reduced order modelling strategy in Section 3.2. To reduce the computational burden, the problem is first transformed from the time domain to the Laplace domain, as explained in Section 3.3. Then, in Section 3.4, a projection-based reduced order modeling technique is introduced, leveraging the similarities among healthy configurations, which are described by the parametric PDE.

3.1 Acoustic-elastic wave equation in the time domain

Throughout this thesis, the generic model (2.1) is described by the parametric acoustic-elastic wave equation. In this section, we present the parametric PDE in its strong form, followed by its variational formulation, its finite element approximation, and algebraic structure in time domain. We employ Newmark method for discretization in time.

Given $\Omega \subset \mathbb{R}^d$, a d-dimensional domain approximating a healthy structure of interest with $d = 2, 3$, and $\partial\Omega$ its piecewise smooth boundary, the acoustic-elastic wave equation in strong form, equipped with suitable boundary and initial conditions for both the displacement field and its derivative, is expressed as

$$\begin{cases} \rho \dfrac{\partial^2 \boldsymbol{u}}{\partial t^2} + \rho\eta \dfrac{\partial \boldsymbol{u}}{\partial t} - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u};\boldsymbol{\mu}) = h(t;\boldsymbol{\mu}) s(\boldsymbol{x};\boldsymbol{\mu}) & \text{in } \Omega \times (0,T] \\ \boldsymbol{u} = \boldsymbol{g}_D(\boldsymbol{x},t;\boldsymbol{\mu}) & \text{on } \Gamma_D \times (0,T] \\ \boldsymbol{\sigma}(\boldsymbol{u};\boldsymbol{\mu}) \cdot \boldsymbol{n} = \boldsymbol{g}_N(\boldsymbol{x},t;\boldsymbol{\mu}) & \text{on } \Gamma_N \times (0,T] \;, \\ \boldsymbol{u}|_{t=0} = \boldsymbol{u}_0(\boldsymbol{x};\boldsymbol{\mu}) & \text{in } \Omega \\ \left.\dfrac{\partial \boldsymbol{u}}{\partial t}\right|_{t=0} = \boldsymbol{v}_0(\boldsymbol{x};\boldsymbol{\mu}) & \text{in } \Omega \end{cases} \tag{3.1}$$

where $\boldsymbol{u}$ represents the displacement field, $\rho \in \mathbb{N}_+$ is the density coefficient, $\eta \in \mathbb{N}$ 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, respectively. $\boldsymbol{\sigma} := \boldsymbol{\sigma}(\boldsymbol{u};\boldsymbol{\mu})$ is the stress tensor

$$\boldsymbol{\sigma} := 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda \mathrm{Tr}\left(\boldsymbol{\varepsilon}(\boldsymbol{u})\right) \mathbf{I}_d,$$

where $\mathbf{I}_d$ is the d-dimensional identity matrix, $\mathrm{Tr}(\cdot)$ is the trace operator applied to the strain tensor $\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right)$ and the Lamé constants μ and λ are expressed by E, the Young's modulus, and ν, the non-dimensional Poisson's ratio, as

$$\mu = \frac{E}{2(1+\nu)} \quad \text{and} \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}. \tag{3.2}$$

In (3.1), $\boldsymbol{n}$ is the outward normal vector to $\partial\Omega$. Γ_D and Γ_N are such that $\partial\Omega = \Gamma_D \cup \Gamma_N$ with $\mathring{\Gamma}_D \cap \mathring{\Gamma}_N = \emptyset$ and represent the portions of the surface of Ω where displacement boundary conditions $\boldsymbol{g}_D := \boldsymbol{g}_D(\boldsymbol{x},t;\boldsymbol{\mu})$ and stress boundary conditions through the traction vector $\boldsymbol{g}_N := \boldsymbol{g}_N(\boldsymbol{x},t;\boldsymbol{\mu})$ are applied, respectively. We note that, alternatively, one could prescribe *free slip* boundary conditions:

$$\begin{cases} \boldsymbol{u} \cdot \boldsymbol{n} = \mathbf{0} \\ (\boldsymbol{\sigma} \cdot \boldsymbol{n}) \cdot \boldsymbol{\tau} = \boldsymbol{g}_N \end{cases} \text{on } \partial\Omega, \tag{3.3}$$

where $\boldsymbol{\tau}$ is the tangential vector to $\partial\Omega$. Finally, $\boldsymbol{u}_0(\boldsymbol{x};\boldsymbol{\mu})$ and $\boldsymbol{v}_0(\boldsymbol{x};\boldsymbol{\mu})$ describe the initial displacement and velocity in space, respectively.

We consider $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^{d_\mu}$ 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. The generic parameter space $\mathcal{P} \subset \mathbb{R}^{d_\mu}$ will be defined by d_μ intervals

$$\mathcal{P} := [\mu_1^1, \mu_2^1] \times [\mu_1^2, \mu_2^2] \times \cdots \times [\mu_1^{d_\mu}, \mu_2^{d_\mu}]. \tag{3.4}$$

In this thesis, we arbitrarily choose the input parameters and the corresponding intervals they belong to, as it will be described in Section 4.4. However, in a real engineering setup, this choice may be inferred using sensitivity analysis combined with a Bayesian approach to better include prior engineering knowledge and available experimental measurements.

While this goes beyond the scope of this thesis, we provide a brief introduction to these topics together with preliminary results in Appendix A.

For the sake of simplicity and consistent with the numerical tests, let us consider homogeneous Dirichlet and Neumann boundary conditions, i.e., $\boldsymbol{g}_D = \mathbf{0}$ and $\boldsymbol{g}_N = \mathbf{0}$, although the non-homogeneous case can be treated similarly. To provide the discrete form of (3.1) we first introduce the parametrized variational problem: for a fixed parameter $\boldsymbol{\mu} \in \mathcal{P}$ and a fixed $t \in (0, T]$, find $\boldsymbol{u} := \boldsymbol{u}(t; \boldsymbol{\mu}) \in V := \{\boldsymbol{w} \in H^1(\Omega; \mathbb{R}^d) : \boldsymbol{w}|_{\Gamma_D} = \mathbf{0}\}$ [1] such that

$$\begin{cases} \rho m\left(\dfrac{\partial^2 \boldsymbol{u}}{\partial t^2}, \boldsymbol{\psi}\right) + \rho\eta\, m\left(\dfrac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{\psi}; \boldsymbol{\mu}\right) + a(\boldsymbol{u}, \boldsymbol{\psi}; \boldsymbol{\mu}) = h(t; \boldsymbol{\mu}) f(\boldsymbol{\psi}; \boldsymbol{\mu}) \quad \forall \boldsymbol{\psi} \in V \\ \boldsymbol{u}(0; \boldsymbol{\mu}) = \boldsymbol{u}_0(\boldsymbol{\mu}) \\ \dfrac{\partial \boldsymbol{u}}{\partial t}(0; \boldsymbol{\mu}) = \boldsymbol{v}_0(\boldsymbol{\mu}) \end{cases}, \tag{3.5}$$

with $\boldsymbol{u}_0(\boldsymbol{\mu}) \in \mathrm{H}^1(\Omega; \mathbb{R}^d)$ and $\boldsymbol{v}_0(\boldsymbol{\mu}) \in \mathrm{L}^2(\Omega; \mathbb{R}^d)$ as the assigned initial conditions.

In the weak formulation (3.5), the bilinear forms $m(\cdot, \cdot)$ and $a(\cdot, \cdot; \boldsymbol{\mu})$, and the functional $f(\cdot; \boldsymbol{\mu})$ have the following expressions

$$\begin{aligned} m(\boldsymbol{u}, \boldsymbol{\psi}) &:= \int_\Omega \boldsymbol{u} \cdot \boldsymbol{\psi} \, d\Omega, \\ a(\boldsymbol{u}, \boldsymbol{\psi}; \boldsymbol{\mu}) &:= \int_\Omega \left(2\mu \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{\psi}) + \lambda (\nabla \cdot \boldsymbol{u})(\nabla \cdot \boldsymbol{\psi})\right) d\Omega, \\ f(\boldsymbol{\psi}; \boldsymbol{\mu}) &:= \int_\Omega \boldsymbol{s}(\boldsymbol{\mu}) \cdot \boldsymbol{\psi} \, d\Omega. \end{aligned} \tag{3.6}$$

Solving problem (3.5) requires suitable numerical approximation techniques, here referred to as *high fidelity* (or *full order*) approximations, to recover a discrete representation of the exact solution. Several classes of numerical methods have been introduced for the discretization of hyperbolic PDEs, among them the finite difference (FD) [Qua09], finite volume (FV) [LeV92], finite element (FE) and discontinuous Galerkin [SF73; EG13; BMS04], and spectral methods [GH01]. In this work, to approximate the solution of (3.5) in space we apply the Galerkin FE method by introducing a finite-dimensional subspace $V_h \subset V$, with $\dim(V_h) = N_h$.

For the sake of simplicity, we consider a domain Ω with polygonal shape and introduce a triangulation $\mathcal{T}_h$, i.e., K non-overlapping triangles ($d = 2$) or tetrahedra ($d = 3$) whose union perfectly coincides with Ω, where $h \in \mathbb{N}_+$ represents the mesh size[2], i.e., $h_K := \mathrm{diam}(K) \leq h$, $\forall K \in \mathcal{T}_h$. Having denoted by $\mathbb{P}_r$ the space of polynomials

---

[1] Throughout 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 $\boldsymbol{u} \in C^1\left([0, T]; L^2(\Omega; \mathbb{R}^d)\right) \cap C^0\left([0, T], V\right)$. Moreover, we note that when one seeks to solve (3.1) with free slip boundary conditions (3.3), V has to be replaced with $V_{FS} = \{\boldsymbol{w} \in H^1(\Omega; \mathbb{R}^d) : \boldsymbol{w} \cdot \boldsymbol{n} = \mathbf{0}\}$.

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

with degree lower or equal to r, the FE space corresponds to $X_h^r = \{\boldsymbol{w}_h \in C^0(\bar{\Omega}) : \boldsymbol{w}_h|_K \in \mathbb{P}_r, \forall K \in \mathcal{T}_h\}$. Consider $V_h := V \cap X_h^r$ 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

$$\boldsymbol{u}_h(\boldsymbol{x}, t; \boldsymbol{\mu}) := \sum_{j=1}^{N_h} u_j(t; \boldsymbol{\mu}) \boldsymbol{\varphi}_j(\boldsymbol{x}), \tag{3.7}$$

where N_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. Then, the Galerkin high-fidelity approximation of (3.5) reads: find $\boldsymbol{u}_h := \boldsymbol{u}_h(\boldsymbol{x}, t; \boldsymbol{\mu}) \in V_h$ such that

$$\begin{cases} \rho m\left(\dfrac{\partial^2 \boldsymbol{u}_h}{\partial t^2}, \boldsymbol{w}_h\right) + \rho\eta\, m\left(\dfrac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{w}_h; \boldsymbol{\mu}\right) + a(\boldsymbol{u}_h, \boldsymbol{w}_h; \boldsymbol{\mu}) = h(t; \boldsymbol{\mu}) f(\boldsymbol{w}_h; \boldsymbol{\mu}) \\ \boldsymbol{u}_h(0; \boldsymbol{\mu}) = \boldsymbol{u}_{h0}(\boldsymbol{\mu}) \\ \dfrac{\partial \boldsymbol{u}_h}{\partial t}(0; \boldsymbol{\mu}) = \boldsymbol{v}_{h0}(\boldsymbol{\mu}) \end{cases}, \tag{3.8}$$

for all $\boldsymbol{w}_h \in V_h$, where $\boldsymbol{u}_{h0}$ and $\boldsymbol{v}_{h0}$ are obtaiend as L^2-projection of $\boldsymbol{u}_0(\boldsymbol{\mu})$ and $\boldsymbol{v}_0(\boldsymbol{\mu})$ over V_h, respectively. Moreover, if we denote by $\mathbf{u}_h(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ the vector having as components the unknown coefficients $u_j(t; \boldsymbol{\mu})$ for $j = 1, \ldots, N_h$, then, at the algebraic level, we obtain the following discrete dynamical system

$$\begin{cases} \rho \mathbf{M}\left(\dfrac{\partial^2 \mathbf{u}_h}{\partial t^2}(t; \boldsymbol{\mu}) + \eta \dfrac{\partial \mathbf{u}_h}{\partial t}(t; \boldsymbol{\mu})\right) + \mathbf{A}(\boldsymbol{\mu}) \mathbf{u}_h(t; \boldsymbol{\mu}) = h(t; \boldsymbol{\mu}) \mathbf{f}(\boldsymbol{\mu}) & \text{in } \Omega \times (0, T] \\ \mathbf{u}_h(0; \boldsymbol{\mu}) = \mathbf{u}_0(\boldsymbol{\mu}) & \text{in } \Omega \\ \dfrac{\partial \mathbf{u}_h}{\partial t}(0; \boldsymbol{\mu}) = \mathbf{v}_0(\boldsymbol{\mu}) & \text{in } \Omega \end{cases}, \tag{3.9}$$

where $\mathbf{M} \in \mathbb{R}^{N_h \times N_h}$ is the mass matrix, $\mathbf{A} := \mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is the stiffness matrix, and $\mathbf{f} := \mathbf{f}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is the right hand side vector with elements

$$\mathbf{M}_{ij} = m(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i) \quad \mathbf{A}_{ij} = a(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i; \boldsymbol{\mu}) \quad \mathbf{f}_i = f(\boldsymbol{\varphi}_i; \boldsymbol{\mu}) \quad \forall i, j = 1, \ldots, N_h. \tag{3.10}$$

Moreover, $\mathbf{u}_0(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ and $\mathbf{v}_0(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ are the FE vector representations of $\boldsymbol{u}_{h0}(\boldsymbol{\mu})$ and $\boldsymbol{v}_{h0}(\boldsymbol{\mu})$, respectively.

To obtain a fully discretized system, we introduce a time-discretization scheme to obtain a sequence (in time) of linear systems to be solved for each parameter $\boldsymbol{\mu}$. In particular, we use the classic Newmark method, defined in [New59], for the time discretization of the second order initial value problem (3.9).

Let us first consider a partition of the time interval $[0, T]$ in $N_t \in \mathbb{N}_+$ 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 the displacement, the velocity, and the acceleration vectors at time t^n as $\mathbf{u}_h^n(\boldsymbol{\mu}) := \mathbf{u}_h(t;\boldsymbol{\mu})\Big|_{t=t^n}$, $\mathbf{v}_h^n(\boldsymbol{\mu}) := \frac{\partial \mathbf{u}_h(t;\boldsymbol{\mu})}{\partial t}\Big|_{t=t^n}$, and $\mathbf{a}_h^n(\boldsymbol{\mu}) := \frac{\partial^2 \mathbf{u}_h(t;\boldsymbol{\mu})}{\partial t^2}\Big|_{t=t^n}$, respectively. Newmark method is defined as

$$\begin{aligned} \mathbf{u}_h^{n+1} :=& \mathbf{u}_h^n + \Delta t \mathbf{v}_h^n + (\Delta t)^2 \left(\beta \mathbf{a}_h^{n+1} + \frac{1-2\beta}{2} \mathbf{a}_h^n \right), \\ \mathbf{v}_h^{n+1} :=& \mathbf{v}_h^n + \Delta t \left(\zeta \mathbf{a}_h^{n+1} + (1-\zeta) \mathbf{a}_h^n \right), \end{aligned} \tag{3.11}$$

where β and ζ are two 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 [QS06; ZT05].

If we replace $\mathbf{u}_h(t;\boldsymbol{\mu})$ and $\frac{\partial \mathbf{u}_h}{\partial t}(t;\boldsymbol{\mu})$ in (3.9) with the first and second expressions of (3.11), respectively, and solve for $\mathbf{a}_h^{n+1}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, we obtain the fully discrete linear system:

$$\mathbf{K}(\boldsymbol{\mu}) \mathbf{a}_h^{n+1}(\boldsymbol{\mu}) = \mathbf{q}^{n+1}(\boldsymbol{\mu}), \tag{3.12}$$

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

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

where $h^n(\boldsymbol{\mu}) := h(t^n;\boldsymbol{\mu})$ and

$$\begin{aligned} \mathbf{q}_{\mathbf{u}}(\boldsymbol{\mu}) :=& \mathbf{A}(\boldsymbol{\mu}), \\ \mathbf{q}_{\mathbf{v}}(\boldsymbol{\mu}) :=& \rho \eta \mathbf{M} + \Delta t \mathbf{A}(\boldsymbol{\mu}), \\ \mathbf{q}_{\mathbf{a}}(\boldsymbol{\mu}) :=& \rho \eta (1-\zeta) \Delta t \mathbf{M} + \frac{1-2\beta}{2} (\Delta t)^2 \mathbf{A}(\boldsymbol{\mu}). \end{aligned}$$

Hence, the semi-discrete variational problem (3.9) is equivalent to the following statement: for $n = 0, \ldots, N_t - 1$, solve (3.12) 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 Newmark method (3.11). 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 [Hor95]. This guarantees that $\mathbf{K}$ is invertible. We finally note that $\mathbf{a}_h^0(\boldsymbol{\mu})$ must be recovered by solving (3.12) with $\mathbf{q}^0(\boldsymbol{\mu}) = h^0(\boldsymbol{\mu}) \mathbf{f}(\boldsymbol{\mu})$.

Remark 1 *Explicit and implicit time discretization methods.* Alternative explicit methods can also be employed to approximate the derivatives in time, e.g., popular methods are the forward/centered Euler, the Lax-Friedrichs, the Lax-Wendroff, or the forward/decentered Euler, also called the upwind method. A necessary condition for the stability of an explicit advancing numerical scheme is the CFL condition (from Courant, Friedrichs, and Levy [CFL28]), which relates the size of the spacial mesh, the time step, and the wave velocity by requiring that the domain of dependence of the numerical scheme contains the domain of dependence of the physical problem. From a practical perspective, the CFL condition implies that, to improve the accuracy of the solution, the underlying mesh has to be sufficiently refined, which may result in a high number of iterations, especially if one is interested in simulating long-term displacements. Instead, implicit unconditionally stable methods, such as the Newmark method, provide some flexibility on the choice of the time step. However, we note that the time step influences the accuracy of the solution especially after many iterations, i.e., when Δt is too big, spurious oscillatory behavior, e.g., amplitude attenuations and phase discrepancies, may be observed at large t^n values. For this reason, it is advisable to follow the rule of thumb of fixing at least 10 points per wavelength. For more details on the finite difference methods for hyperbolic equations we refer the reader to Chapter 13 of [Qua09] and for more details on the numerical approximation of conservation laws to [Hes17].

3.2 The need for a reduced order model

From a computational cost point of view, solving the time-dependent problem (3.12) with the method described above requires the solution of N_t linear systems of size N_h. In the SHM setting, dealing with complex large structures implies we face several challenges. On one hand, N_h can be of order $\mathcal{O}(10^q)$ with $q = 5, 6, 7$, and, on the other hand, sensor measurements may be signals recorded over several seconds. Considering in addition the issues highlighted in Remark 1, the number of time steps may also be large, e.g., of order $\mathcal{O}(10^q)$ with $q = 4, 5$. Keeping in mind the original goal of constructing a robust database of synthetic sensor measurements comprising many possible combinations of environmental and operational conditions, as discussed in Chapter 2, the problem of solving (3.12) for a high number of parameters becomes a critical bottleneck.

The repeated solution of parametric PDEs with a varying input parameter is a frequent issue when dealing with many-query contexts such as sensitivity analysis, parameter estimation, design, or PDE-constrained optimization to name a few. In particular, when either the number of DOFs or the number of time steps is large, solving the full-order time-dependent model is not affordable. A *reduced order model* (ROM) provides an approximation of a parametric full-order model (FOM), which is fast to evaluate without compromising the overall accuracy. The construction of the ROM leverages the assumption that the manifold spanned by all possible solutions of the parametrized problem has a considerably smaller dimension than the space spanned by the basis functions of the full-order model. By exploiting an offline–online decoupling, the ROM solution is obtained as a linear combination of problem-specific basis functions, computed during an expensive offline phase. During the offline phase, a reduced basis is extracted from a collection of high-fidelity solutions, which are obtained by solving the high fidelity model for specific parameter values (snapshots). Projection-based methods, e.g., the *proper orthogonal decomposition* (POD), are popular approaches to construct the reduced basis and generate the lower dimensional subspace in which the ROM solution is sought. The so-constructed reduced-space, spanned by this set of basis functions, represents the essential features of the map $\boldsymbol{\mu} \to \boldsymbol{u}_h(\boldsymbol{\mu})$, where $\boldsymbol{u}_h(\boldsymbol{\mu})$ is the high-fidelity solution computed for the parameter $\boldsymbol{\mu}$. Then, in the online phase, for each new value of the parameters, the ROM solutions are inexpensively obtained by solving a smaller linear system, i.e., the reduced problem, and the solution is finally projected back to the original space. When the Galerkin projection is employed to generate reduced-order vectors this approach is typically referred to as the *POD-Galerkin method.*

For unsteady problems, due to the travelling-wave behavior of the solution, projection-based ROM strategies pose several challenges, e.g., the manifold of all possible solutions can often not be compressed to a small reduced basis. Furthermore, the sampling strategy is more complicated since it has to combine the solution at different time instants and for different values of the parameter. In the literature, different strategies have been proposed to construct a suitable reduced space, e.g., the POD-greedy sampling strategy, see, e.g., [HO08], combines POD for the time-trajectories with the greedy procedure in the parameter space, or the double POD approach, see, e.g., [NMA15; DS18], which relies on two nested POD, where the POD of N_t basis-in-time, each one built by combining different parameter values, is performed. Sampling issues are emhasized when the time-dependent problem has many time steps and hence the set of snapshots become large, resulting in an expensive POD strategy. To overcome this difficulty, randomized SVD algorithms [HMT11] are employed to efficiently generate a reduced basis of the problem, see, e.g., [WRH20]. Alternative local reduced basis spaces, based on different snapshot clustering techniques, have been proposed to approximate problems in cardiac electrophysiology in, e.g., [PMQ18]. Finally, we also observe that, recent efforts have been made in the direction of space-time approaches, where projection in space and time is performed simultaneosly, see, e.g., [CC19]. While an in depth study of the problem is needed to fully evaluate the performance of this approach applied to the acoustic-elastic problem, we speculate that the memory storage requirement, corresponding several matrices of size $N_h \times N_t$, associated with space-time methods, is not compatible with high dimensional problems.

For time dependent linear problems, an effective strategy is to replace the time domain formulation with a frequency domain formulation and to apply a ROM method to replace the so-obtained full-order FE problem in frequency domain with a low-dimensional model. On one hand, by replacing the time domain with the frequency domain, the number of times one expects to solve a linear system equivalent to (3.12) is reduced to N_z with $N_z \ll N_t$. On the other hand, recurring to a ROM strategy reduces the number of degrees of freedom of each linear system, i.e., from N_h to N with $N \ll N_h$. Here N_z is the number of principal frequencies, characterising the full-order model, and N is the number of reduced basis. In particular, motivated by the interest in studying the *transient response* of damaged structures using active sources, we replace the time domain formulation with the Laplace domain formulation, where the Laplace transform of the displacement becomes the unknown field, as described in detail in Section 3.3. To compare the time-dependent experimental sensor measurements in the online phase, we transform the synthetic signals, computed in frequency domain, back to time domain. We briefly review numerical inverse Laplace transform strategies in 3.3.1 and provide the details of Weeks method [Wee66; LG86], in Section 3.3.2, which is chosen to reconstruct the time history of the displacements in this thesis. The specific framework of Weeks method to reconstruct the time signals requires computing the solutions only for few fixed complex frequencies, which makes the use of the ROM method very efficient. The details on ROM are provided in Section 3.4, where we describe how to combine the frequencies of Weeks method with the parameters to construct a reduced model. Finally, in Section 3.4.2, we present the cotangent lift method, a symplectic Galerkin projection method based on a proper symplectic decomposition, to address stability issues which arise when solving the frequency-based problem decoupled in its real and imaginary parts.

In the simulation-based structural damage detection field, the Laplace transform is used for example in [ZBA13; KM17]. In the former, the authors model the behavior of smart structures combined with piezoelectric actuators and sensors using the boundary element method applied to the elastodynamics equation. In the latter, the response of a cantilever beam to arbitrary time-dependent thermal actuations is studied in the Laplace domain and a quadrature method, known as Gaver–Stehfest method, is used to numerically invert the Laplace transform. The Fourier transform may be a suitable alternative, see, e.g., [TPYP18], if one wishes to study the periodic behavior of the vibrations of a structure under the effect of continuous sources, e.g., wind, waves, or tides, or if the classification step relies on features extracted directly from the solution in frequency domain. The choice of the Laplace transform will be further motivated in Section 4.3.1, where we discuss the damage sensitive features extracted from raw time signals and in particular the importance of the *arrival time* in wave-based monitoring approaches.

3.3 Acoustic-elastic wave equation in the frequency domain

In this section we present the acoustic-elastic wave equation in Laplace domain and provide its finite element approximation and algebraic structure. To reconstruct the solution in time, numerical inverse Laplace transform methods are described in Section 3.3.1 and Section 3.3.2 focuses on Weeks method. From a notation point of view, we observe that throughout this thesis we employ the tilde superscript to identify complex quantities and $\imath$ is the imaginary unit[3].

---

[3] The greek letter $\imath$ should not be confused with the subscript i, used to indicate the location of the n_s sensor locations, i.e., $\boldsymbol{x}_i \in \Omega$ for $i = 1, \dots, n_s$.

The Laplace transform $\tilde{f}(\tilde{z})$, $\tilde{z} \in \mathbb{C}$, of a real function $f(t)$

$$\tilde{f}(\tilde{z}) = \int_0^{\infty} e^{-\tilde{z}t} f(t) dt \tag{3.13}$$

is often used in the solution of PDEs to eliminate the temporal derivatives. Indeed, it is a powerful approach to capture the initial value responses in time dependent systems and it has been extensively used in the literature, e.g., to study the transient in electrical circuits [Gol66] or to address the problem of mass transport in groundwater [Sud89]. The resulting systems become easier to solve and the solution can be inverted to the time domain, usually by relying on approximate inversion techniques.

Given a fixed frequency $\tilde{z} \in \mathbb{C}$ and a fixed input parameter $\boldsymbol{\mu} \in \mathcal{P}$, by multiplying the acoustic-elastic wave equation (3.1) by $e^{-\tilde{z}t}$ and integrating in time over the infinite interval $[0, \infty)$, the time-dependent problem reduces to the computation of the Laplace transform of $\boldsymbol{u}(\boldsymbol{x}, t; \boldsymbol{\mu})$ evaluated at $\tilde{z}$, i.e., find $\tilde{\boldsymbol{u}} := \tilde{\boldsymbol{u}}(\boldsymbol{x}, \tilde{z}; \boldsymbol{\mu}) : \Omega \times \mathbb{C} \times \mathcal{P} \to \mathbb{C}^d$ such that

$$\begin{cases} \rho(\tilde{z}^2 + \eta\tilde{z})\tilde{\boldsymbol{u}} - \nabla \cdot \boldsymbol{\sigma}(\tilde{\boldsymbol{u}}; \boldsymbol{\mu}) = \tilde{h}(\tilde{z}; \boldsymbol{\mu}) s(\boldsymbol{x}; \boldsymbol{\mu}) & \text{in } \Omega \\ \tilde{\boldsymbol{u}} = \boldsymbol{0} & \text{on } \Gamma_D \ , \\ \boldsymbol{\sigma}(\tilde{\boldsymbol{u}}; \boldsymbol{\mu}) \cdot \boldsymbol{n} = \boldsymbol{0} & \text{on } \Gamma_N \end{cases} \tag{3.14}$$

where, for the sake of simplicity, but without loss of generality, we have assumed homogeneous boundary conditions and zero initial conditions. In (3.14), $\tilde{h} := \tilde{h}(\tilde{z}; \boldsymbol{\mu}) : \mathbb{C} \times \mathcal{P} \to \mathbb{C}^d$ 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 3.1 applies here. Given $\tilde{V} := \{\boldsymbol{w} \in H^1(\Omega; \mathbb{C}^d) : \boldsymbol{w}|_{\Gamma_D} = \boldsymbol{0}\}$, the corresponding Hilbert space in frequency domain and $\tilde{V}_h := \tilde{V} \cap X_h^r$ its finite-dimensional counterpart, the approximate Galerkin problem becomes: for all $\tilde{z} \in \mathbb{C}$ and all $\boldsymbol{\mu} \in \mathcal{P}$ find $\tilde{\boldsymbol{u}}_h(\tilde{z}; \boldsymbol{\mu}) \in \tilde{V}_h$ such that

$$\rho \left(\tilde{z}^2 + \eta\tilde{z}\right) m(\tilde{\boldsymbol{u}}_h(\tilde{z}; \boldsymbol{\mu}), \tilde{\boldsymbol{v}}_h^*) + a(\tilde{\boldsymbol{u}}_h(\tilde{z}; \boldsymbol{\mu}), \tilde{\boldsymbol{v}}_h^*; \boldsymbol{\mu}) = \tilde{h}(\tilde{z}; \boldsymbol{\mu}) f(\tilde{\boldsymbol{v}}_h; \boldsymbol{\mu}), \quad \forall \tilde{\boldsymbol{v}}_h \in \tilde{V}_h, \tag{3.15}$$

where the bilinear forms $m(\cdot, \cdot)$, $a(\cdot, \cdot; \boldsymbol{\mu})$ and the functional $f(\cdot; \boldsymbol{\mu})$ are defined in (3.6), and $\tilde{\boldsymbol{v}}_h^*$ refers to the complex conjugate of the test function $\tilde{\boldsymbol{v}}_h$. Let $\{\boldsymbol{\varphi}_j\}_{j=1}^{N_h}$ be a Lagrangian basis with $\boldsymbol{\varphi}_j : \mathbb{R}^d \to \mathbb{R}^d$ for all $j = 1, \ldots, N_h$, then $\tilde{\boldsymbol{u}}_h \in \tilde{V}_h$ is the Galerkin approximation of $\tilde{\boldsymbol{u}} \in \tilde{V}$, i.e.,

$$\tilde{\boldsymbol{u}}_h(\boldsymbol{x}, \tilde{z}; \boldsymbol{\mu}) := \sum_{j=1}^{N_h} \tilde{u}_j(\tilde{z}; \boldsymbol{\mu}) \boldsymbol{\varphi}_j(\boldsymbol{x}). \tag{3.16}$$

If we denote by $\tilde{\mathbf{u}}_h(\tilde{z}; \boldsymbol{\mu})$ the vector having as components the complex unknown coefficients $\tilde{u}_j(\tilde{z}; \boldsymbol{\mu})$, solving problem (3.15) is equivalent to: find $\tilde{\mathbf{u}}_h(\tilde{z}; \boldsymbol{\mu}) \in \mathbb{C}^{N_h}$ such that

$$\left[\rho \left(\tilde{z}^2 + \eta\tilde{z}\right) \mathbf{M} + \mathbf{A}(\boldsymbol{\mu})\right] \tilde{\mathbf{u}}_h(\tilde{z}; \boldsymbol{\mu}) = \tilde{h}(\tilde{z}; \boldsymbol{\mu}) \mathbf{f}(\boldsymbol{\mu}), \tag{3.17}$$

where the mass matrix $\mathbf{M} \in \mathbb{R}^{N_h \times N_h}$, the stiffness matrix $\mathbf{A} := \mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$, and the right hand side vector $\mathbf{f} := \mathbf{f}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ are defined in (3.10).

This system can be decomposed into a set of $2N_h$ real equations such that, for a given $\tilde{z} := \alpha + \imath y$, with $\alpha \in \mathbb{R}$ and $y \in \mathbb{R}$, the solution of (3.17) can be rewritten as

$$\tilde{\mathbf{u}}_h(\tilde{z};\boldsymbol{\mu}) := \mathbf{u}_h^{\alpha}(\tilde{z};\boldsymbol{\mu}) + \imath \mathbf{u}_h^{y}(\tilde{z};\boldsymbol{\mu}),$$

where $\mathbf{u}_h^{\alpha}(\tilde{z};\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ and $\mathbf{u}_h^{y}(\tilde{z};\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ are the real and imaginary components of $\tilde{\mathbf{u}}_h(\tilde{z};\boldsymbol{\mu}) \in \mathbb{C}^{N_h}$, respectively. This splitting is especially important for implementation purposes as explained in Remark 2 and, by simple manipulations, we obtain

$$\begin{bmatrix} \mathbf{K}^{\alpha}(\tilde{z};\boldsymbol{\mu}) & -\mathbf{K}^{y}(\tilde{z};\boldsymbol{\mu}) \\ \mathbf{K}^{y}(\tilde{z};\boldsymbol{\mu}) & \mathbf{K}^{\alpha}(\tilde{z};\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \mathbf{u}_h^{\alpha}(\tilde{z};\boldsymbol{\mu}) \\ \mathbf{u}_h^{y}(\tilde{z};\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \mathbf{q}^{\alpha}(\tilde{z};\boldsymbol{\mu}) \\ \mathbf{q}^{y}(\tilde{z};\boldsymbol{\mu}) \end{bmatrix}, \tag{3.18}$$

where

$$\begin{aligned} \mathbf{K}^{\alpha}(\tilde{z};\boldsymbol{\mu}) :=& \Theta^{\alpha}(\tilde{z};\boldsymbol{\mu})\mathbf{M} + \mathbf{A}(\boldsymbol{\mu}), \\ \mathbf{K}^{y}(\tilde{z};\boldsymbol{\mu}) :=& \Theta^{y}(\tilde{z};\boldsymbol{\mu})\mathbf{M}, \\ \mathbf{q}^{p}(\tilde{z};\boldsymbol{\mu}) :=& h^{p}(\tilde{z};\boldsymbol{\mu})\mathbf{f}(\boldsymbol{\mu}), \quad \text{for } p \in \{\alpha, y\}. \end{aligned} \tag{3.19}$$

Here, $\Theta^{\alpha} := \Theta^{\alpha}(\tilde{z};\boldsymbol{\mu}) = \rho\left(\alpha^2 - y^2 + \eta\alpha\right)$ and $\Theta^{y} := \Theta^{y}(\tilde{z};\boldsymbol{\mu}) = \rho y\left(2\alpha + \eta\right)$ so that $\Theta^{\alpha} + \imath\Theta^{y} = \rho(\tilde{z}^2 + \eta\tilde{z})$. Moreover, $h^{\alpha} := h^{\alpha}(\tilde{z};\boldsymbol{\mu})$ and $h^{y} := h^{y}(\tilde{z};\boldsymbol{\mu})$ are the real and imaginary parts of $\tilde{h}(\tilde{z};\boldsymbol{\mu})$, respectively.

We note that the coefficients of (3.16) can be written as $\tilde{u}_j(\tilde{z};\boldsymbol{\mu}) = u_j^{\alpha}(\tilde{z};\boldsymbol{\mu}) + \imath u_j^{y}(\tilde{z};\boldsymbol{\mu})$, where $u_j^{\alpha}(\tilde{z};\boldsymbol{\mu})$ and $u_j^{y}(\tilde{z};\boldsymbol{\mu})$ are the j^{th} and $(N_h + j)^{\text{th}}$ entries of the solution vector obtained by solving (3.18) for a fixed frequency $\tilde{z}$ and parameter $\boldsymbol{\mu}$, respectively.

Remark 2 *The implementation of complex systems in `FEniCS`.* In this thesis, the implementation of the acoustic-elastic PDE in both time and frequency domain is developed within the open source finite element library `FEniCS` [LMW12; ABH+15]. Currently, `FEniCS` does not support complex numbers, however, it does support mixed spaces and coupled PDEs. Thus, the weak form is computed using two real trial functions $\boldsymbol{u}^{\alpha} \in V$ and $\boldsymbol{u}^{y} \in V$ and two real test functions $\boldsymbol{v}^{\alpha} \in V$ and $\boldsymbol{v}^{y} \in V$:

$$\begin{aligned} &\Theta^{\alpha}\left(m(\boldsymbol{u}^{\alpha},\boldsymbol{v}^{\alpha}) + m(\boldsymbol{u}^{y},\boldsymbol{v}^{y})\right) + \Theta^{y}\left(m(\boldsymbol{u}^{\alpha},\boldsymbol{v}^{y}) - m(\boldsymbol{u}^{y},\boldsymbol{v}^{\alpha})\right) \\ &\quad + a(\boldsymbol{u}^{\alpha},\boldsymbol{v}^{\alpha};\boldsymbol{\mu}) + a(\boldsymbol{u}^{y},\boldsymbol{v}^{y};\boldsymbol{\mu}) = h^{a} f(\boldsymbol{v}^{\alpha};\boldsymbol{\mu}) + h^{y} f(\boldsymbol{v}^{y};\boldsymbol{\mu}), \\ &\Theta^{\alpha}\left(m(\boldsymbol{u}^{y},\boldsymbol{v}^{\alpha}) - m(\boldsymbol{u}^{\alpha},\boldsymbol{v}^{y})\right) + \Theta^{y}\left(m(\boldsymbol{u}^{\alpha},\boldsymbol{v}^{\alpha}) + m(\boldsymbol{u}^{y},\boldsymbol{v}^{y})\right) \\ &\quad - a(\boldsymbol{u}^{\alpha},\boldsymbol{v}^{y};\boldsymbol{\mu}) + a(\boldsymbol{u}^{y},\boldsymbol{v}^{\alpha};\boldsymbol{\mu}) = h^{y} f(\boldsymbol{v}^{\alpha};\boldsymbol{\mu}) - h^{\alpha} f(\boldsymbol{v}^{y};\boldsymbol{\mu}). \end{aligned}$$

In the implementation, the two variational forms are summed and the solution $\boldsymbol{u}$ is sought in the mixed space $V_2 = V \times V$. In practice, V and V_2 are defined using the built-in functions `VectorElement` and `MixedElement`, respectively. To access the real and imaginary components of the solution the function `split` is used.

3.3.1 Numerical inverse Laplace transforms

The Laplace transform introduced in the previous section offers a powerful method for solving linear systems. To recover the inverse f of a Laplace transform $\tilde{f}$ in (3.13), one has to compute the inverse Laplace transform, which corresponds to an integration in the complex plane:

$$f(t) = \frac{1}{2\pi\imath} \int_{\alpha-\imath\infty}^{\alpha+\imath\infty} e^{t\tilde{z}} \tilde{f}(\tilde{z}) d\tilde{z}, \quad t > 0, \quad \alpha > \alpha_0, \tag{3.20}$$

where $\alpha \in \mathbb{R}$ is a free parameter, whose value must be greater than the real part of any singularity in $\tilde{f}(\tilde{z})$, the so-called “Laplace convergence abscissa” α_0[4]. This integral, known as the Bromwich integral, is difficult to evaluate analytically and one needs to resort to numerical approximations.

As introduced in Chapter 2 and as will be further explained in Section 4.3, our goal is to construct a database of displacement time signals, collected at few specific points, i.e., the sensor locations $\boldsymbol{x}_i \in \Omega$ for $i = 1, \ldots, n_s$. 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 Section 3.1, i.e., $\boldsymbol{x}_i$ is not necessarily a DOF. The goal of this section is to reconstruct the displacements signals $\boldsymbol{u}_h(\boldsymbol{x}_i, t; \boldsymbol{\mu})$, defined in (3.7), from their Laplace transforms $\tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \tilde{z}; \boldsymbol{\mu})$ at different locations $\boldsymbol{x}_i \in \Omega$ for $i = 1, \ldots, n_s$. This implies that the generic Laplace transform $\tilde{f}(\tilde{z})$ in (3.20) is here represented by the complex displacement $\tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \tilde{z}; \boldsymbol{\mu})$, and $f(t)$ is replaced by $\boldsymbol{u}_h(\boldsymbol{x}_i, t; \boldsymbol{\mu})$. We remind that $\tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \tilde{z}; \boldsymbol{\mu})$, defined in (3.16), is obtained by solving (3.15) for a fixed parameter $\boldsymbol{\mu} \in \mathcal{P}$ and a fixed frequency $\tilde{z} \in \mathbb{C}$, and then evaluated at $\boldsymbol{x}_i \in \Omega$ for $i = 1, \ldots, n_s$. Analogously, $\boldsymbol{u}_h(\boldsymbol{x}_i, t; \boldsymbol{\mu})$, defined in (3.7), is the solution of (3.8) for a fixed $\boldsymbol{\mu} \in \mathcal{P}$ and a fixed $t \in [0, T]$, and evaluated at $\boldsymbol{x}_i \in \Omega$. By means of a parametrization of the Bromwich integral (3.20) as $\tilde{z} = \alpha + \imath y$, we can recover $\boldsymbol{u}_h(\boldsymbol{x}_i, t; \boldsymbol{\mu})$ as an integral over the infinite imaginary axis, i.e.,

$$\boldsymbol{u}_h(\boldsymbol{x}_i, t; \boldsymbol{\mu}) = \frac{e^{\alpha t}}{2\pi\imath} \int_{-\infty}^{\infty} e^{\imath t y} \tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \alpha + \imath y; \boldsymbol{\mu}) dy, \tag{3.21}$$

for $i = 1, \ldots, n_s$ and $t \in [0, T]$. More specifically, since we seek to reconstruct the discrete time signal, the continuous variable t is here replaced by t^n for $n = 0, \ldots, N_t$. Ultimately, the outputs of interest $g_i(t^n; \boldsymbol{\mu})$, introduced in (2.2), will be approximated by

$$\hat{g}_i(t^n; \boldsymbol{\mu}) := \ell\left(\hat{\boldsymbol{u}}_h(\boldsymbol{x}_i, t^n; \boldsymbol{\mu})\right) \quad \text{for } n = 1, \ldots, N_t, \tag{3.22}$$

where $\hat{\boldsymbol{u}}_h(\boldsymbol{x}_i, t^n; \boldsymbol{\mu})$ is an approximation to $\boldsymbol{u}_h(\boldsymbol{x}_i, t; \boldsymbol{\mu})$, obtained by using a numerical inverse Laplace method, and $\ell : \mathbb{R}^d \to \mathbb{R}^{d_\ell}$ is the output of interest function.

The literature on the computation of (3.20) is extensive and we refer the interested reader to [Coh07] and the survey in [Duf93]. The review presents three methods: the trapezoidal

---

[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.

rule, Talbot method and the expansion in Laguerre polynomials, also known as Weeks method. The results presented in this thesis rely on the third method, whose details are described in Section 3.3.2. Before that, we briefly present an overview of the other two methods, highlighting the reasons of our choice.

The first scheme is a direct application of the trapezoidal rule to the truncated Bromwich integral, i.e, $|y| < \theta$ in (3.21). This method may be inefficient due to the highly oscillatory nature of the integrand, which contributes to discretization and truncation errors, as pointed out in [Cru76]. Nevertheless, the inclusion of correction terms or sequence accelerator strategies to extrapolate a slowly convergent sequence can provide effective results, as demonstrated in [HH84]. The second method is also based on the trapezoidal rule, but along a special contour line devised by Talbot in [Tal79], which suppresses the oscillations. This scheme contains a number of free parameters which can be estimated theoretically if the locations of the singularities are known or using a commercial software presented in [MR90] if only some ansatz is available. A unified framework for these types of methods is proposed in [AW06], where the following inversion formula summarizes the quadrature-based schemes as

$$f(t) \simeq f_n(t) = \frac{1}{t} \sum_{k=0}^{n} \tilde{\omega}_k \tilde{f}\left(\frac{\tilde{a}_k}{t}\right), \quad 0 < t < \infty.$$

Here the weights $\tilde{\omega}_k$ and nodes $\tilde{a}_k$ are complex numbers, which depend on n, but do not depend on the transform $\tilde{f}$ or the time argument t. Nevertheless, we observe that these schemes are impractical in our context because $\tilde{f}$ depends on the independent variable t. This means that, to reconstruct an approximation of the entire discrete time series $[g_i(t^n; \boldsymbol{\mu}), \ldots, g_i(t^{N_t}; \boldsymbol{\mu})]$, we would need to solve the linear system (3.18) as many times as the number of time steps N_t. As a result, the computational cost would be greater than solving the direct problem with Newmark method.

Instead, Weeks method, which was first proposed in [Wee66] and then improved in [LG86; GGLM88], is based on an expansion in terms of orthonormal Laguerre functions. This method has one particular advantage over the other two since it returns an explicit form of the approximate solution $\hat{\boldsymbol{u}}_h(\boldsymbol{x}_i, t; \boldsymbol{\mu})$. The expensive phase of the method corresponds to the calculation of the expansion coefficients. Once these are determined, the inverse Laplace transform is obtained at any new value t by means of a series summation, which requires minimal additional cost. In the trapezoidal and Talbot methods, one has to restart for any new value of t. For this reason, Weeks method is chosen in this thesis as the most suitable method to numerically invert the Laplace transform (3.21). For completeness, we mention that there exist variants of the trapezoidal rule, relying on added correction terms, see, e.g., [Cru76; Dur74], where the Laplace transform does not depend on time. These variants have been used to reconstruct time histories with a time interval of T of the order of $\mathcal{O}(10^{-q})$ seconds with $q = 4$ or $q = 5$ in [ZBA13]. However, they often become oscillatory and deviate from the right solution when T is large.

3.3.2 Weeks method

In this section we describe Weeks method to compute an approximation of the inverse Laplace transform (3.20) and refer to [LG86; GGLM88; Wei99] and references therein for more details. The discussion is put into the perspective of recovering sensor signals, i.e., finding an approximation of the output of interest for the acoustic-elastic wave equation as in (3.21). We also provide some computational guidance in the choice of the free parameters and conclude with a qualitative comparison of the solution of the acoustic-elastic wave equation in time domain with Newmark method and the reconstructed solution obtained by applying Weeks method to the solution of the acoustic-elastic wave equation in frequency domain.

The numerical inversion of (3.20) using Weeks method has the following representation:

$$f(t) = e^{(\alpha-b)t}\sum_{k=0}^{\infty} a_k L_k(2bt),$$

where $\alpha \in \mathbb{R}$ with $\alpha > \alpha_0$ and $b \in \mathbb{R}_+$ are two free parameters and $L_k(\cdot)$ denotes the Laguerre polynomial of degree k which can be computed recursively using, e.g., Clenshaw's algorithm [Cle55]. The expansion coefficients a_k, which depend on the Laplace transform $\tilde{f}(\alpha + \imath y)$, are defined by a Maclaurin series

$$G(\omega;\alpha,b) = \frac{2b}{1-\omega}\tilde{f}\left(\alpha + b\frac{1+\omega}{1-\omega}\right) = \sum_{k=0}^{\infty} a_k \omega^k, \tag{3.23}$$

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

$$a_k = \frac{1}{2\pi\imath}\int_{|\omega|=1} \frac{G(\omega;\alpha,b)}{\omega^{k+1}} d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} G(e^{\imath\theta};\alpha,b)e^{-\imath k\theta} d\theta, \tag{3.24}$$

where the change of variable $\omega = e^{\imath\theta}$ has been used. To approximate this integral, we follow [Wei99], 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 (3.24), which would require one to evaluate $\tilde{f}(\tilde{z})$ at infinity. Using the midpoint discretization based on $2N_z$ intervals, the coefficients a_k are therefore approximated as

$$a_k \simeq \hat{a}_k = \frac{1}{2N_z}\sum_{j=-N_z}^{N_z-1} G(e^{\imath\theta_{j+1/2}};\alpha,b)e^{-\imath k\theta_{j+1/2}} \quad \text{for } k = 0,\dots,N_z-1, \tag{3.25}$$

where $\theta_j = j\pi/N_z$. By evaluating $G(\cdot;\alpha,b)$, defined in (3.23), at $e^{\imath\theta_{j+1/2}}$ and inserting it into (3.25), we obtain

$$\hat{a}_k = \frac{b}{N_z} \sum_{j=-N_z}^{N_z-1} \frac{e^{-\imath k\theta_{j+1/2}}}{1-e^{\imath\theta_{j+1/2}}} \tilde{f}(\tilde{z}_j), \tag{3.26}$$

where $\tilde{z}_j$ has the following expression

$$\tilde{z}_j = \alpha + b\frac{1+e^{\imath\theta_{j+1/2}}}{1-e^{\imath\theta_{j+1/2}}} = \alpha + \imath b \cot\frac{\theta_{j+1/2}}{2} \quad \text{for } j=-N_z,\ldots,N_z-1. \tag{3.27}$$

Finally, the approximated inverse Laplace transform, based on a N_z-term truncation of the Laguerre series, becomes

$$\hat{f}(t) := e^{(\alpha-b)t} \sum_{k=0}^{N_z-1} \hat{a}_k L_k(2bt).$$

As mentioned in the previous section, we need a numerical inverse Laplace transform strategy such as Weeks method to approximate the discrete time signals at sensors locations. Hence, we apply Weeks method to (3.21) and obtain

$$\hat{\boldsymbol{u}}_h(\boldsymbol{x}_i, t^n; \boldsymbol{\mu}) := e^{(\alpha-b)t^n} \sum_{k=0}^{N_z-1} \hat{\boldsymbol{a}}_{k,h} L_k(2bt^n), \tag{3.28}$$

for $i=1,\ldots,n_s$ and $n=1,\ldots,N_t$, where the expansion coefficients $\hat{\boldsymbol{a}}_{k,h}$ are derived in (3.26) by replacing $\tilde{f}(\tilde{z}_j)$ with $\tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \tilde{z}_j; \boldsymbol{\mu})$, i.e., the solution of (3.15) for a fixed parameter $\boldsymbol{\mu}$ and a fixed frequency $\tilde{z}_j$, and evaluated at $\boldsymbol{x}_i$:

$$\hat{\boldsymbol{a}}_{k,h} := \frac{b}{N_z} \sum_{j=-N_z}^{N_z-1} \frac{e^{-\imath k\theta_{j+1/2}}}{1-e^{\imath\theta_{j+1/2}}} \tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \tilde{z}_j; \boldsymbol{\mu}), \quad k=0,\ldots,N_z-1. \tag{3.29}$$

Here, the frequencies $\tilde{z}_j$ are defined in (3.27) for $j=-N_z,\ldots N_z-1$ and the additional subscript h indicates that the Laplace transform is the solution of a PDE using a FE discretization. Then, the full discrete time history $\hat{\mathbf{g}}_i^m \in \mathbb{R}^{d_\ell \times (N_t+1)}$ is defined as

$$\hat{\mathbf{g}}_i^m := [\hat{g}_i(t^0; \boldsymbol{\mu}_m), \ldots, \hat{g}_i(t^{N_t}; \boldsymbol{\mu}_m)], \quad \text{for } i=1,\ldots,n_s, \tag{3.30}$$

and it corresponds to the Weeks approximation of the discrete time signals $\mathbf{g}_i^m$, defined in (2.4), for all sensors locations. The single time steps $\hat{g}_i(t^n; \boldsymbol{\mu}_m)$ in (3.30) are defined in (3.22) for $n=1,\ldots,N_t$, i.e. they are obtained by applying the output of interest function $\ell : \mathbb{R}^d \to \mathbb{R}^{d_\ell}$ to the Weeks approximations (3.28).

Remark 3 *Halving the number of computations.* We note that only the imaginary part of the frequencies (3.27) varies with the discretization index, while the real part α remains fixed. By exploiting trigonometric identities, one can show that the frequencies of (3.27) satisfy the following property

$$\tilde{z}_j = \tilde{z}^*_{(-j-1)}, \quad \text{for } j = 0, \dots, N_z - 1$$

where $\tilde{z}^*$ is the complex conjugate of $\tilde{z}$. Moreover, it is easy to prove that if $(\boldsymbol{u}_h^\alpha + \imath \boldsymbol{u}_h^y) \in \tilde{V}$ is the solution of (3.15) for $\tilde{z} = \tilde{z}_j$, then its conjugate $(\boldsymbol{u}_h^\alpha - \imath \boldsymbol{u}_h^y) \in \tilde{V}$ is the solution of (3.15) for $\tilde{z} = \tilde{z}_j^*$. This halves the number of times we need to solve the corresponding linear system (3.17), or equivalently its split version (3.18). From a practical perspective, we only need to compute $\tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \tilde{z}_j; \boldsymbol{\mu})$ for $j = 0, \dots, N_z - 1$ to compute the coefficients (3.29) and the remaining functions, i.e., those corresponding to $\tilde{z}_{-N_z}, \dots, \tilde{z}_{-1}$, are obtained as

$$\tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \tilde{z}_{(-j-1)}; \boldsymbol{\mu}) = \tilde{\boldsymbol{u}}_h^*(\boldsymbol{x}_i, \tilde{z}_j; \boldsymbol{\mu}) \quad \text{for } j = 0, \dots, N_z - 1. \tag{3.31}$$

Algorithm 1 summarizes Weeks method and how it is connected to the solution of the acoustic-elastic wave equation in Laplace domain. As mentioned in Section 3.2, we highlight that generating one synthetic sample for the database of time signals by using frequency domain solutions comes at a lower computational cost than generating the same sample in time. In time domain we need to solve (3.12) N_t times, while in frequency domain we need to solve (3.18) N_z times. Since the two linear systems are of the same size up to a multiplicative factor 2, it is clear that the frequency approach is advantageous if the number of frequencies, needed to generate an adequate numerical inverse Laplace transform, are significantly less than the number of time steps needed to generate the discrete time signal, i.e., if $N_z \ll N_t$. Although the number of frequencies N_z, needed to reconstruct the time signals with a prescribed accuracy, is problem dependent, in most cases, N_t is expected to be much larger than N_z mainly because we are interested in observing a long time window of the sensor signal to fully capture its damage-sensitive features. Moreover, depending on the chosen time integration method, N_t may be large to guarantee a small time step, needed either to satisfy stability requirements (explicit methods) or to avoid loss of accuracy (implicit methods), as explained in Remark 1. The additional computational cost to construct the expansions coefficients in Leguerre polynomials for the reconstructions of the time signal (lines 7-9 of Algorithm 1) is negligible with respect to the time needed to solve (3.18). We also note that the solution of (3.18) for each frequency is independent of all other frequencies, leading to a fully parallelizable method, different from Newmark's method.

Free parameters in Weeks method

Weeks method contains two free parameters[5], $\alpha \in \mathbb{R}$ and $b \in \mathbb{R}_+$, and the accuracy of this algorithm depends critically on their values. There exist several rules of thumb in the literature, see, e.g., [Wee66; PM71; GGLM88], where an estimate for α and b often

---

[5] In this thesis, we also use the term hyper-parameters to describe these variables.

Algorithm 1 Construction of an approximate time signal by Weeks method for high-fidelity solutions in Laplace domain

```
Inputs: $\boldsymbol{x}_i \in \Omega, \boldsymbol{\mu}_m \in \mathcal{P}, \alpha \in \mathbb{R}, b \in \mathbb{R}_+, N_z \in \mathbb{N}_+, N_t \in \mathbb{N}_+$
for $j = 0$ to $N_z - 1$ do
    Compute $y_j = \mathrm{Im}(\tilde{z}_j)$, defined in (3.27)
    Solve linear system (3.18) for $\tilde{z}_j = \alpha + \imath y_j$ and $\boldsymbol{\mu}_m$
    Evaluate function $\tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \tilde{z}_j; \boldsymbol{\mu}_m)$ at sensor location $\boldsymbol{x}_i$ using (3.16)
    Obtain $\tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \tilde{z}_{(2N_z-j+1)}; \boldsymbol{\mu}_m)$ by complex conjugation as in (3.31)
Compute expansion coefficients $\hat{\boldsymbol{a}}_{k,h}$, defined in (3.29), for $k = 0, \dots, N_z - 1$
for $n = 1$ to $N_t$ do
    Compute $\hat{\boldsymbol{u}}_h(\boldsymbol{x}_i, t^n; \boldsymbol{\mu}_m)$ by expansion in the Laguerre polynomials with (3.28)
Retrieve full time series $\hat{\mathbf{g}}_i^m$, defined in (3.30)
return $\hat{\mathbf{g}}_i^m$
```

requires the user to know at least the real part α_0 of the leftmost singularity of the Laplace transform. 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 [GLR89], 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 (3.18), evaluated at the sensors locations, i.e., $\tilde{\boldsymbol{u}}_h(\boldsymbol{x}_i, \tilde{z}_j; \boldsymbol{\mu}_m)$ defined in (3.16). This is challenging because this quantity is expensive to compute and thus it would be available only at few frequency locations $\tilde{z}_j$. Moreover, it would be complex to verify that this Laplace transform fulfils the properties required to belong to the class defined in [GLR89].

Two additional strategies to find the optimal values are proposed in [Wei99]. Whereas 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. 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. However, no alternative is known for the case in which the Laplace transform is not known analytically. Indeed, in in [Wei99], 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, Weeks method is ideal to retrieve the entire time signal at the cost of solving N_z times the linear system (3.18). Unfortunately, the solutions proposed in [Wei99] to identify optimal values of α and b are not suitable as they would require many additional solutions of (3.18). In this thesis we choose these hyper-parameters using a different approach, based on the comparison of the signals obtained with Newmark method and the reconstructed signals with Weeks method for various values of α and b.

Let $\boldsymbol{\mu}^* \in \mathcal{P}$ be the mid point value of the intervals defined in (3.4), i.e.,

$$\boldsymbol{\mu}^* = \left[\frac{\mu_1^1 + \mu_2^1}{2}, \frac{\mu_1^2 + \mu_2^2}{2}, \dots, \frac{\mu_1^{n_\mu} + \mu_2^{n_\mu}}{2}\right]. \tag{3.32}$$

Then, for a fixed resolution N_z and a fixed number of time steps N_t, we repeatedly apply Algorithm 1 for few input values in the ansatz intervals $\alpha \in [\alpha_m, \alpha_M]$ and $b \in [b_m, b_M]$ to recover the time signals $\hat{\mathbf{g}}_i^*$ for $i = 1, \dots, n_s$. Then, we choose as optimal the values for which the L^2-error between the recovered time signals and the corresponding Newmark solutions at all sensors locations is minimized, i.e.,

$$\alpha^{opt}, b^{opt} := \min_{\alpha, b} \left\| \sum_{i=1}^{n_s} (\mathbf{g}_i^* - \hat{\mathbf{g}}_i^*) \right\|_2^2, \tag{3.33}$$

where $\mathbf{g}_i^*$ and $\hat{\mathbf{g}}_i^*$ are the discrete time histories defined in (2.4) and (3.30), respectively. We note that the entries $g_i(t^n; \boldsymbol{\mu})$ of $\mathbf{g}_i^*$ are obtained by applying the output of interest function ℓ to the solution $\boldsymbol{u}_h(\boldsymbol{x}_i, t^n; \boldsymbol{\mu}^*)$, defined in (3.7). Clearly, only $\hat{\mathbf{g}}_i^*$ depends on the parameters α and b. For the sake of simplicity, and because we consider a relatively small parameter space in the simulations, in this thesis the optimal values α^{opt}, b^{opt} will be used for all other simulations with input parameters laying within the parameter space $\mathcal{P}$. The proposed strategy is problem-specific and alternative methods to identify the optimal hyper-parameters α and b for each new parameter $\boldsymbol{\mu}$ could be further explored, possibly by identifying non-linear maps using an artificial neural network.

In the remainder of this section, we provide numerical results to guide our choice in selecting the values of the hyper-parameters α and b. These results are based on the 2D numerical example with $n_s = 15$ fixed sensor locations, which will be introduced in Section 4.4. We note that the number of frequencies N_z, used to reconstruct the time signals, can also be considered as an additional hyper-parameter and will also be discussed.

In particular, we fix $\boldsymbol{\mu}^* = [1, 0.33, 2] \in \mathcal{P}$ by applying the definition in (3.32) to the parameter space chosen as

$$\mathcal{P} := [0.999, 1.001] \times [0.329, 0.331] \times [1.9, 2.1].$$

We additionally choose $N_z = 200$, $\Delta t = 1e-3$, and $N_t = 30'000$. Then, following the aforementioned strategy on the ansatz intervals $[0.1, 0.9]$ and $[1, 16]$ for α and b, respectively, we obtain $\alpha^{opt} = 0.2$ and $b^{opt} = 7.25$ as shown in the left plot of Figure 3.1. By performing a further refinement on the intervals $[0.1, 0.4]$ and $[4.5, 9.25]$ for α and b, respectively, we finally obtain $\alpha^{opt} = 0.26$ and $b^{opt} = 6.5$, as shown in right plot of Figure 3.1. Motivated by the fact that the parameter space $\mathcal{P}$ is rather small and by further observing in Figure 3.1 that, qualitatively, the error depends smoothly on α and b, these two values will be used for all the numerical simulations in this thesis.

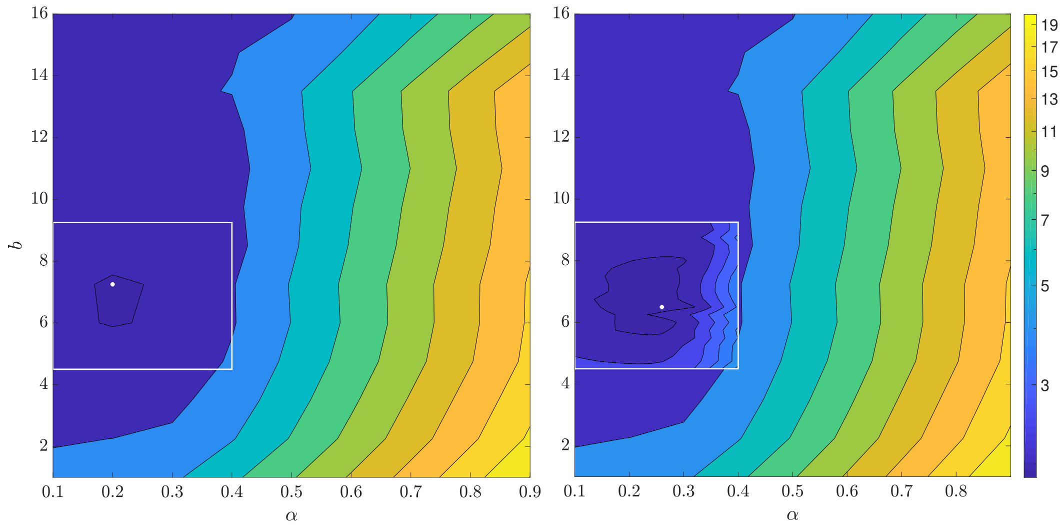


Figure 3.1 – Contour plot of the error between the time signals obtained with Newmark's method and Weeks method to guide the choice of the values of the hyper-parameters α and b. The L^2-errors, defined in (3.33), are obtained using 9 equally spaced points for $\alpha \in [0.1, 0.9]$ and 13 equally spaced points for $b \in [1, 16]$, leading to $\alpha^{opt} = 0.2$ and $b^{opt} = 7.25$ indicated by the white dot (*left*). The additional refinement in the region $\alpha \in [0.1, 0.4]$ and $b \in [4.5, 9.25]$ if performed for 16 and 20 equally spaced points, respectively, leading to the ultimate optimal values $\alpha^{opt} = 0.26$ and $b^{opt} = 6.5$ (*right*).

Figure 3.2 shows that, for these optimal values, the error of the reconstructed solution in 2D decreases with second order as the number of coefficients N_z in the Laugerre expansion increases. This graph qualitatively justifies our choice of using $N_z = 200$ frequencies for the 2D examples. In general, Weeks method yields exponential convergence in the number of frequencies, see, e.g., [Wei99], and the second order convergence shown in Figure 3.2 reflects the error with respect to the approximated solution obtained by using the Newmark method, which is second order accurate.

Additionally, Figure 3.3 shows the behavior of the time-dependent solutions recovered at the 6th sensor of coordinates $\boldsymbol{x}_6 = (0.275, 0.925)$, using either Newmark method or Weeks method. These plots, further confirm a qualitatively good reconstrcution using $N_z = 200$ for the 2D simulations. In Figure 3.3, as time increases, we observe a degradation between the solutions in time domain and the reconstructed solution in frequency domain, which is expected considering the expansion in the Laguerre polynomials.

To guarantee better alignment with the Newmark solution, considered here as a reference solution, and avoid spurious oscillations before the signal arrival, for more complex problems, e.g., for the 3D simulations, the number of frequencies N_z is increased to 500. Additionally, for these problems, we will consider a reduced time frame of $N_t = 22'500$ time steps to discard incorrect oscillations at long term caused by Weeks method.

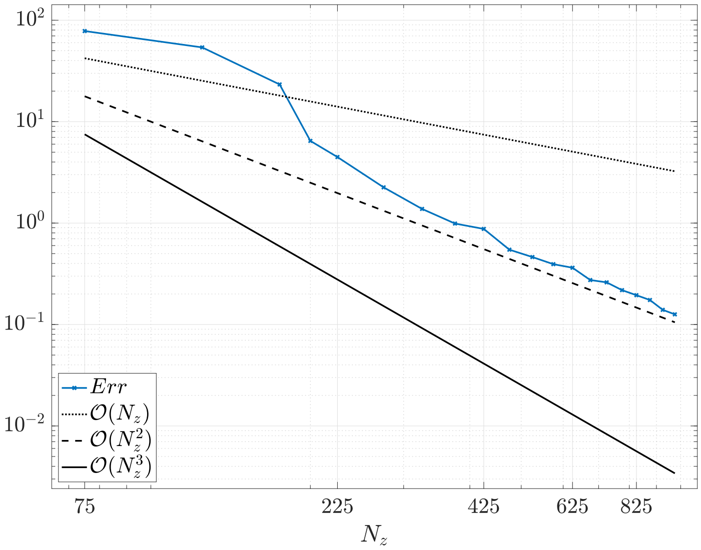


Figure 3.2 – Loglog plot of the error between the reconstructed time signals with Weeks method and the Newmark signals, i.e., $\|\sum_{i=1}^{n_s}(\hat{\mathbf{g}}_i^* - \mathbf{g}_i^*)\|_2^2$ for increasing values of N_z. The reconstructed high-fidelity signals $\hat{\mathbf{g}}_i^*$ are obtained by applying Algorithm 1 with $\alpha^{opt} = 0.26$ and $b^{opt} = 6.5$ for different number of frequencies 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 $\boldsymbol{\mu}^* = [1, 0.33, 2]$.

3.4 Model Order Reduction

In this section we present a reduced order modeling (ROM) approach that significantly reduces the computational burden of repeatedly solving the parametrized problem (3.15) by exploiting the $\boldsymbol{\mu}$-dependence of the solution. Indeed, solving the high-fidelity complex linear system (3.17), or its real counterpart (3.18), for many input parameters is essential to construct a synthetic database to train a robust classifier to detect anomalies in unseen data. Even though the translation to frequency domain, described in Section 3.3, reduces the computational effort to generate the datasets of discrete time signals, a substantial speedup can be achieved by applying ROM techniques. As discussed in Section 3.2, projection-based ROM techniques, and in particular the well-known Reduced 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, uncertainty quantification, and others. We refer the interested reader to [QMN15; HRS15; RHP07] and the references therein for an in-depth overview of RB methods.

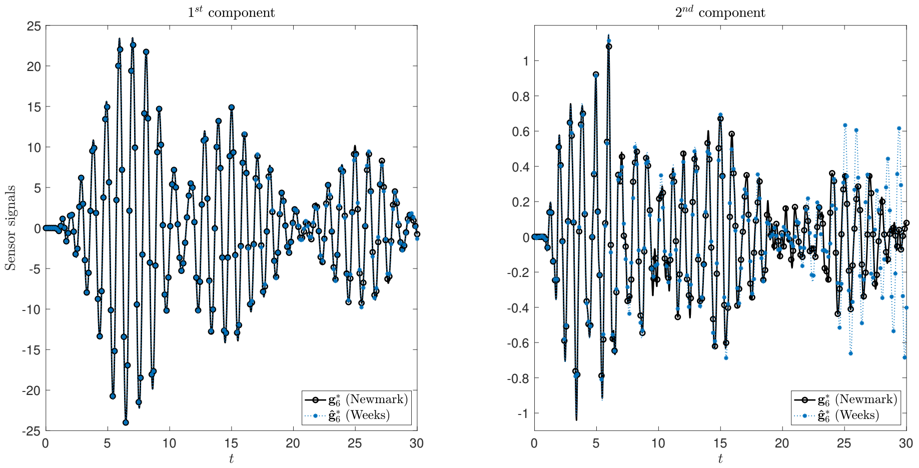


Figure 3.3 – Comparison of the 2D high-fidelity signals retrieved at the 6th sensor when using Newmark method (*black circled line*) or the Laplace method with Weeks reconstruction (*blue starred line*) using $\alpha^{opt} = 0.26$, $b^{opt} = 6.5$, $N_z = 200$. Both $\hat{\mathbf{g}}_i^*$ and $\mathbf{g}_i^*$ 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, i.e., the displacements in the x- and y- directions. For clarity the graphs only show a subset of the N_t steps.

3.4.1 Reduced basis method

The main idea of RB methods is to generate an approximate solution to (3.15) 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 3.3, 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 values of the parameter 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.

In this thesis, motivated by the implementation constraints discussed in Remark 2, we present the details of the RB method only with respect to the real-valued linear system (3.18), but an analogous discussion can be carried out for its complex counterpart. We use the proper orthogonal decomposition (POD) to generate the low-dimensional subspace where the RB solution is sought. The greedy algorithm [GP05; VPRP03; RHP07], an alternative method to construct the basis functions, iteratively chooses the next basis by evaluating an error estimate of the error between the high-fidelity and the reduced model. However, for the acoustic-elastic equation in Laplace domain we do not have an effective error bound and we therefore choose the POD approach.

First, we construct the snapshot matrix whose columns are the high-fidelity solutions of (3.18), decomposed in their real and imaginary parts, obtained for $N_s < N_h$ different values of the input frequency $\tilde{z} \in \mathbb{C}$ and the physical parameter $\boldsymbol{\mu} \in \mathcal{P}$:

$$\mathbf{S} := \begin{bmatrix} \mathbf{S}^{\alpha} \\ \mathbf{S}^{y} \end{bmatrix} \in \mathbb{R}^{2N_h \times N_s}, \tag{3.34}$$

where $\mathbf{S}^p \in \mathbb{R}^{N_h \times N_s}$ is defined as

$$\mathbf{S}^p := \left[\mathbf{u}_h^p \left(\tilde{z}_0; \boldsymbol{\mu}_0 \right) | \dots | \mathbf{u}_h^p \left(\tilde{z}_{n_s-1}; \boldsymbol{\mu}_{n_s-1} \right) \right], \quad \text{for } p \in \{\alpha, y\}. \tag{3.35}$$

For a prescribed dimension $N \leq N_s$, 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

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

where

$$\mathbf{U} := [\boldsymbol{\zeta}_1 | \dots | \boldsymbol{\zeta}_{2N_h}] \quad \text{and} \quad \mathbf{Z} := [\boldsymbol{\Psi}_1 | \dots | \boldsymbol{\Psi}_{N_s}]$$

are two orthogonal matrices of dimension $2N_h \times 2N_h$ and $N_s \times N_s$, respectively, and $\boldsymbol{\Sigma} := \operatorname{diag}(\sigma_1, \dots, \sigma_{N_s}) \in \mathbb{R}^{2N_h \times N_s}$ with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{N_s}$. The POD basis of dimension N is defined as the set of the first N left singular vectors of $\mathbf{U}$, i.e.,

$$\mathbf{V} := [\boldsymbol{\zeta}_1 | \dots | \boldsymbol{\zeta}_N] \in \mathbb{R}^{2N_h \times N}. \tag{3.36}$$

This basis minimizes the 2-norm of the projection error of the snapshot vectors (see e.g., Proposition 6.1 of [QMN15]). To guarantee that the projection is controlled by a suitable tolerance ε_{POD}, the number of basis N is chosen as the smallest integer such that

$$I(N) := \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{N_s} \sigma_i^2} \geq 1 - \varepsilon_{POD}, \tag{3.37}$$

i.e., the energy retained by the last $N_s - N$ modes is equal to or smaller than a prescribed tolerance ε_{POD} [QMN15].

Since the real and imaginary functions $\boldsymbol{u}_h^p(\tilde{z}; \boldsymbol{\mu}) \in V_h \subset V$ for $p \in \{\alpha, y\}$, it is natural to consider the SVD with respect to a scalar product induced by the $\mathbf{X}_{2h}$-norm, where $\mathbf{X}_{2h} \in \mathbb{R}^{2N_h \times 2N_h}$ is the symmetric positive definite matrix associated with a scalar product on the real V_h of dimension $2N_h$. Specifically, $\mathbf{X}_{2h}$ 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 choose

$$(\mathbf{X}_h)_{ij} = m(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i) + a(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i; \boldsymbol{\mu}),$$

where $m(\cdot,\cdot)$ and $a(\cdot,\cdot;\boldsymbol{\mu})$ are defined in (3.6) for an arbitrary choice of $\boldsymbol{\mu}$. We choose the parameters that guarantee unitary Lamé constants (3.2), i.e., $\lambda = \mu = 1$. By considering the SVD of $\mathbf{X}_{2h}^{1/2}\mathbf{S}$ we obtain a basis that is $\mathbf{X}_{2h}$-orthonormal.

When N_h is large, the POD basis can conveniently be obtained by computing the first N eigenvectors of the *correlation matrix* $\mathbf{C} \in \mathbb{R}^{N_s \times N_s}$, which is constructed as

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

Hence, the POD basis (3.36) can be rewritten as

$$\mathbf{V} := \begin{bmatrix} \mathbf{V}^\alpha \\ \mathbf{V}^y \end{bmatrix}, \quad \text{with } \mathbf{V}^p := [\boldsymbol{\zeta}_1^p | \dots | \boldsymbol{\zeta}_N^p] \in \mathbb{R}^{N_h \times N} \quad \text{for } p \in \{\alpha, y\},$$

where

$$\begin{bmatrix} \boldsymbol{\zeta}_j^\alpha \\ \boldsymbol{\zeta}_j^y \end{bmatrix} := \frac{1}{\sigma_j}\mathbf{S}\boldsymbol{\psi}_j, \quad \text{for } j = 1, \dots, N,$$

and $\boldsymbol{\psi}_j$ for $j = 1, \dots, N$ are the first N eigenvectors, solution of the following eigenvalue problem

$$\mathbf{C}\boldsymbol{\psi}_j = \sigma_j^2\boldsymbol{\psi}_j \quad j = 1, \dots, N.$$

We remark that computing the POD basis by solving the above eigenvalue problem may yield inaccurate results for the eigenvectors associated with small singular values. Since, roundoff errors may be introduced in the construction of $\mathbf{C}$, in this situations, it is preferable to compute the SVD by means of stable algorithms, see [CD06].

Then, the high-fidelity problem (3.18) is replaced by the following reduced algebraic problem for any instance of frequency $\tilde{z}$ and the parameter $\boldsymbol{\mu}$:

$$\mathbf{K}_N(\tilde{z}; \boldsymbol{\mu})\, \mathbf{u}_N(\tilde{z}; \boldsymbol{\mu}) = \mathbf{q}_N(\tilde{z}; \boldsymbol{\mu}), \tag{3.38}$$

where the low-dimensional unknown $\mathbf{u}_N(\tilde{z}; \boldsymbol{\mu}) \in \mathbb{R}^N$ is the RB solution, $\mathbf{K}_N(\tilde{z}; \boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$

is the RB matrix, and $\mathbf{q}_N(\tilde{z};\boldsymbol{\mu}) \in \mathbb{R}^N$ is the right hand side vector, defined as

$$\begin{aligned}
\mathbf{K}_N(\tilde{z};\boldsymbol{\mu}) :=& \mathbf{V}^T\mathbf{K}_h(\tilde{z};\boldsymbol{\mu})\mathbf{V} = \begin{bmatrix}\mathbf{V}^\alpha \\ \mathbf{V}^y\end{bmatrix}^T \begin{bmatrix}\mathbf{K}^\alpha(\tilde{z};\boldsymbol{\mu}) & -\mathbf{K}^y(\tilde{z};\boldsymbol{\mu}) \\ \mathbf{K}^y(\tilde{z};\boldsymbol{\mu}) & \mathbf{K}^\alpha(\tilde{z};\boldsymbol{\mu})\end{bmatrix} \begin{bmatrix}\mathbf{V}^\alpha \\ \mathbf{V}^y\end{bmatrix} \\
=& \mathbf{V}^{\alpha T}\left(\mathbf{K}^\alpha(\tilde{z};\boldsymbol{\mu})\mathbf{V}^\alpha - \mathbf{K}^y(\tilde{z};\boldsymbol{\mu})\mathbf{V}^y\right) + \mathbf{V}^{yT}\left(\mathbf{K}^y(\tilde{z};\boldsymbol{\mu})\mathbf{V}^\alpha + \mathbf{K}^\alpha(\tilde{z};\boldsymbol{\mu})\mathbf{V}^y\right), \\
\mathbf{q}_N(\tilde{z};\boldsymbol{\mu}) :=& \mathbf{V}^T\mathbf{q}_h(\tilde{z};\boldsymbol{\mu}) = \begin{bmatrix}\mathbf{q}^\alpha(\tilde{z};\boldsymbol{\mu}) \\ \mathbf{q}^y(\tilde{z};\boldsymbol{\mu})\end{bmatrix},
\end{aligned} \tag{3.39}$$

where $\mathbf{K}_h(\tilde{z};\boldsymbol{\mu})$ and $\mathbf{q}_h(\tilde{z};\boldsymbol{\mu})$ are the high-fidelity matrix and right hand side in (3.18).

The high-fidelity solution $\mathbf{u}_h(\tilde{z};\boldsymbol{\mu})$ is approximated by $\mathbf{V}\mathbf{u}_N(\tilde{z};\boldsymbol{\mu})$, i.e., the RB approximation is recovered as the linear combination of the columns of $\mathbf{V}$, which is parameter independent. We finally note that the RB problem (3.38) is obtained by enforcing the residual of the high fidelity problem computed on the RB solution to be orthogonal to the subspace V_N generated by the columns of $\mathbf{V}$, i.e.,

$$\mathbf{V}^T\left(\mathbf{q}_h(\tilde{z};\boldsymbol{\mu}) - \mathbf{K}_h(\tilde{z};\boldsymbol{\mu})\mathbf{V}\mathbf{u}_N(\tilde{z};\boldsymbol{\mu})\right) = \mathbf{0}.$$

This classic geometric orthogonality criterion is the reason why RB methods are considered as projection-based methods.

3.4.2 Cotangent lift method

We notice that the reduced matrix $\mathbf{K}_N$ in (3.39) fails to preserve the structure of the high-fidelity matrix $\mathbf{K}$, which causes the reduced solutions to be unstable for some values of $\tilde{z}$ and $\boldsymbol{\mu}$. This unfortunate condition is associated with the fact that the acoustic-elastic problem in the Laplace domain and split in its real and imaginary parts is a non coercive problem. For this type of problems, standard Galerkin projection is known to be potentially unstable and a common strategy to stabilize the problem is to recur to Petrov-Galerkin projection. The peculiarity of this method, often employed to solve the so-called *inf-sup* problems, is to carefully choose a space for the test functions that is different from the space where the solution is sought. Although this strategy is not developed here, we refer the reader to [Qua09; QMN15; HRS15] for a general introduction on Petrov-Galerkin projection in the reduced basis context and to [DPW14; CC19; TY20] for specific examples where Petrov-Galerkin projection is used to stabilize the reduced solution of wave equations. In this thesis instead, stability at the reduced level is guaranteed by resorting to a proper symplectic decomposition (PSD) with a symplectic Galerkin projection, which allow us to preserve the physical properties of the high-fidelity approximation at the reduced level too. In particular, we first apply the *cotangent-lift* method introduced in [PM16], where the snapshot matrix (3.34) is considered in extended form

$$\mathbf{S}_{cl} := [\mathbf{S}^{\alpha}, \mathbf{S}^{y}] \in \mathbb{R}^{N_h \times 2N_s}, \tag{3.40}$$

where $\mathbf{S}^{\alpha}$ and $\mathbf{S}^{y}$ are defined in (3.35). The corresponding POD basis $\mathbf{\Phi} \in \mathbb{R}^{N_h \times N}$ of dimension N is defined as the set of N_h-dimensional vectors

$$\boldsymbol{\zeta}_i^{cl} := \frac{1}{\sigma_i} \mathbf{S}_{cl} \boldsymbol{\psi}_i^{cl} \quad \text{for } i = 1, \dots, N, \tag{3.41}$$

obtained from the first N eigenvectors $\boldsymbol{\psi}^{cl}$ of the correlation matrix $\mathbf{C}_{cl} \in \mathbb{R}^{2N_s \times 2N_s}$, which is defined as

$$\mathbf{C}^{cl} := \mathbf{S}_{cl}^T \mathbf{X}_h \mathbf{S}_{cl} = [\mathbf{S}^{\alpha}, \mathbf{S}^{y}]^T \mathbf{X}_h [\mathbf{S}^{\alpha}, \mathbf{S}^{y}] = \begin{bmatrix} \mathbf{S}^{\alpha,T} \mathbf{X}_h \mathbf{S}^{y} & \mathbf{S}^{\alpha,T} \mathbf{X}_h \mathbf{S}^{y} \\ \mathbf{S}^{y,T} \mathbf{X}_h \mathbf{S}^{\alpha} & \mathbf{S}^{y,T} \mathbf{X}_h \mathbf{S}^{y} \end{bmatrix}. \tag{3.42}$$

Then, the symplectic basis is constructed as

$$\mathbf{V}_{cl} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi} \end{bmatrix} \in \mathbb{R}^{2N_h \times 2N}. \tag{3.43}$$

Although $\mathbf{V}_{cl}$ is $2N$-dimensional, the real and imaginary component are represented by the same set of N basis. Moreover, by construction, $\mathbf{\Phi}^T \mathbf{X}_h \mathbf{\Phi} = \mathbf{I}_N$. Therefore, $\mathbf{V}_{cl}$ is $\mathbf{X}_{2h}$-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 frequency $\tilde{z}$ and a new parameter $\boldsymbol{\mu}$ we need to solve the following reduced system of dimension $2N$:

$$\mathbf{K}_N^{cl}(\tilde{z}; \boldsymbol{\mu}) \begin{bmatrix} \mathbf{u}_N^{\alpha}(\tilde{z}; \boldsymbol{\mu}) \\ \mathbf{u}_N^{y}(\tilde{z}; \boldsymbol{\mu}) \end{bmatrix} = \mathbf{q}_N^{cl}(\tilde{z}; \boldsymbol{\mu}), \tag{3.44}$$

where the reduced matrix $\mathbf{K}_N^{cl}(\tilde{z}; \boldsymbol{\mu}) \in \mathbb{R}^{2N \times 2N}$ is obtained as

$$\begin{aligned} \mathbf{K}_N^{cl}(\tilde{z}; \boldsymbol{\mu}) :=& \mathbf{V}_{cl}^T \mathbf{K}_h(\tilde{z}; \boldsymbol{\mu}) \mathbf{V}_{cl} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi} \end{bmatrix}^T \begin{bmatrix} \mathbf{K}^{\alpha}(\tilde{z}; \boldsymbol{\mu}) & -\mathbf{K}^{y}(\tilde{z}; \boldsymbol{\mu}) \\ \mathbf{K}^{y}(\tilde{z}; \boldsymbol{\mu}) & \mathbf{K}^{\alpha}(\tilde{z}; \boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \mathbf{\Phi} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi} \end{bmatrix} \\ =& \begin{bmatrix} \mathbf{\Phi}^T \mathbf{K}^{\alpha}(\tilde{z}; \boldsymbol{\mu}) \mathbf{\Phi} & -\mathbf{\Phi}^T \mathbf{K}^{y}(\tilde{z}; \boldsymbol{\mu}) \mathbf{\Phi} \\ \mathbf{\Phi}^T \mathbf{K}^{y}(\tilde{z}; \boldsymbol{\mu}) \mathbf{\Phi} & \mathbf{\Phi}^T \mathbf{K}^{\alpha}(\tilde{z}; \boldsymbol{\mu}) \mathbf{\Phi} \end{bmatrix}. \end{aligned} \tag{3.45}$$

Similarly, the reduced right hand side $\mathbf{q}_N^{cl}(\tilde{z}; \boldsymbol{\mu}) \in \mathbb{R}^{2N}$ is obtained as

$$\mathbf{q}_N^{cl}(\tilde{z}; \boldsymbol{\mu}) := \mathbf{V}_{cl}^T \mathbf{q}_h(\tilde{z}; \boldsymbol{\mu}) = \begin{bmatrix} \mathbf{\Phi}^T \mathbf{q}^{\alpha}(\tilde{z}; \boldsymbol{\mu}) \\ \mathbf{\Phi}^T \mathbf{q}^{y}(\tilde{z}; \boldsymbol{\mu}) \end{bmatrix}. \tag{3.46}$$

Again, $\mathbf{K}_h(\tilde{z}; \boldsymbol{\mu})$ and $\mathbf{q}_h(\tilde{z}; \boldsymbol{\mu})$ are the high-fidelity matrix and right hand side in (3.18). In this setting and for the specific choice of the parameter space, presented in Chapter 4, the blocks defining $\mathbf{K}_h(\tilde{z}; \boldsymbol{\mu})$ and $\mathbf{q}_h(\tilde{z}; \boldsymbol{\mu})$ depend affinely on the parameters, i.e., they can be written as

$$\mathbf{K}^p(\tilde{z};\boldsymbol{\mu}) = \sum_{i=1}^{Q_{p,k}} \Theta_i^{p,k}(\tilde{z};\boldsymbol{\mu})\mathbf{K}_i^p \quad \mathbf{q}^p(\tilde{z};\boldsymbol{\mu}) = \sum_{j=1}^{Q_{p,q}} \Theta_j^{p,q}(\tilde{z};\boldsymbol{\mu})\mathbf{q}_j^p, \quad \text{for } p \in \{\alpha, y\}. \tag{3.47}$$

Here $\Theta_i^{p,k}$, $\Theta_j^{p,q}$ for $i = 1,\dots,Q_{p,k}$ and $j = 1,\dots,Q_{q,k}$ are two sets of scalar functions $\Theta_j^{p,k}$, $\Theta_j^{p,q} : \mathbb{C}\times\mathcal{P} \to \mathbb{R}$, $\mathbf{K}_i^p$ for $i = 1,\dots,Q_{p,k}$ is a set of $(\tilde{z};\boldsymbol{\mu})$-independent matrices, and $\mathbf{q}_i^p$ for $j = 1,\dots,Q_{q,k}$ is a sets of et of $(\tilde{z};\boldsymbol{\mu})$-independent vectors for $p \in \{\alpha, y\}$. We note that the specific values are problem-dependent; they can be derived from Equation (3.18) and they depend on the specific choice of the parameter set. Because of this property, known as *affine parametric dependence* of $\mathbf{K}^p(\tilde{z};\boldsymbol{\mu})$ and $\mathbf{q}^p(\tilde{z};\boldsymbol{\mu})$, and by inserting (3.47) into (3.45) and (3.46), we obtain

$$\mathbf{K}_N^{cl}(\tilde{z};\boldsymbol{\mu}) = \begin{bmatrix} \sum_{i=1}^{Q_{\alpha,k}} \Theta_i^{\alpha,k}(\tilde{z};\boldsymbol{\mu})\boldsymbol{\Phi}^T\mathbf{K}_i^\alpha\boldsymbol{\Phi} & -\sum_{i=1}^{Q_{y,k}} \Theta_i^{y,k}(\tilde{z};\boldsymbol{\mu})\boldsymbol{\Phi}^T\mathbf{K}_i^y\boldsymbol{\Phi} \\ \sum_{i=1}^{Q_{y,k}} \Theta_i^{y,k}(\tilde{z};\boldsymbol{\mu})\boldsymbol{\Phi}^T\mathbf{K}_i^y\boldsymbol{\Phi} & \sum_{i=1}^{Q_{\alpha,k}} \Theta_i^{\alpha,k}(\tilde{z};\boldsymbol{\mu})\boldsymbol{\Phi}^T\mathbf{K}_i^\alpha\boldsymbol{\Phi} \end{bmatrix},$$

$$\mathbf{q}_N^{cl}(\tilde{z};\boldsymbol{\mu}) = \begin{bmatrix} \sum_{i=1}^{Q_{\alpha,q}} \Theta_i^{\alpha,q}(\tilde{z};\boldsymbol{\mu})\boldsymbol{\Phi}^T\mathbf{q}_i^\alpha\boldsymbol{\Phi} \\ \sum_{i=1}^{Q_{y,q}} \Theta_i^{y,q}(\tilde{z};\boldsymbol{\mu})\boldsymbol{\Phi}^T\mathbf{q}_i^y\boldsymbol{\Phi} \end{bmatrix}.$$

In this way, in the online phase, the assembling of the reduced system (3.44) will benefit from a considerable speed up since the arrays

$$\boldsymbol{\Phi}^T\mathbf{K}_i^p\boldsymbol{\Phi}, \quad \boldsymbol{\Phi}^T\mathbf{q}_j^p \quad \text{for } i = 1,\dots,Q_{p,k},\ j = 1,\dots,Q_{p,q},\ p \in \{\alpha, y\}$$

can be precomputed and stored once during the offline stage.

Algorithm 2 summarizes the cotangent lift method to construct a symplectic RB basis.

Algorithm 2 Construction of a symplectic basis using the cotangent lift method

```
Inputs: $\mathbf{S}^\alpha, \mathbf{S}^y, \mathbf{X}_h, N_s, \varepsilon_{POD}$
Assemble the snapshot matrix $\mathbf{S}_{cl}$ using (3.40)
Assemble the corresponding correlation matrix $\mathbf{C}_{cl}$ using (3.42)
Solve the eigenvalue problem $\mathbf{C}_{cl}\boldsymbol{\psi}_i^{cl} = \sigma_i^2\boldsymbol{\psi}_i^{cl}$ for $i = 1\dots,N_s$
Define $N$ as the minimum integer that satisfies (3.37)
Compute $\boldsymbol{\zeta}_i^{cl}$, for $i = 1,\dots,N$ using (3.41) and assemble $\boldsymbol{\Phi} = [\boldsymbol{\zeta}_1^{cl} \,|\, \dots \,|\, \boldsymbol{\zeta}_N^{cl}]$
return $\mathbf{V}_{cl}$, defined in (3.43)
```

Remark 4 *Sampling strategy for $\tilde{z}$ and $\boldsymbol{\mu}$.* To recover the time-dependent signals using Weeks method, defined in Section 3.3.2, for a new parameter $\boldsymbol{\mu} \in \mathcal{P}$, we need to solve the N-dimensional reduced system (3.44) N_z times, once per each frequency $\tilde{z}_j$ defined in (3.27). However, since the frequencies $\tilde{z}_j$ only depend on the fixed hyper-parameters α, b, and the number of frequencies N_z, they are essentially fixed. Therefore, the frequency $\tilde{z}$ does not have to be considered as an additional parameter *per se*. Instead, by choosing the number of snapshots N_s to be a multiple of the number of frequencies N_z, we can fix the snapshots to be computed for those exact frequencies that will be needed online. Given $k_z \in \mathbb{N}_+$, we sample $N_s := k_z N_z$ parameters $\boldsymbol{\mu} \in \mathcal{P}$ and pair them with the N_z frequencies so that the snapshot matrix (3.35) becomes

$$\begin{aligned} \mathbf{S}^p := \big[\mathbf{u}_h^p\left(\tilde{z}_0; \boldsymbol{\mu}_0\right) | \dots | \mathbf{u}_h^p\left(\tilde{z}_{N_z-1}; \boldsymbol{\mu}_{N_z-1}\right) | \dots \\ \dots | \mathbf{u}_h^p(\tilde{z}_0; \boldsymbol{\mu}_{(k_z-1)N_z}) | \dots | \mathbf{u}_h^p\left(\tilde{z}_{N_z-1}; \boldsymbol{\mu}_{k_z N_z-1}\right)\big], \quad \text{for } p \in \{\alpha, y\}, \end{aligned} \tag{3.48}$$

where $\tilde{z}_j$ are defined in (3.27) for $j = 0, \dots, N_z - 1$. Provided N_z is sufficiently large to ensure that the high-fidelity time signals, retrieved with Weeks method, are a good approximation of the high-fidelity time signals that could have been obtained with Newmark method, N_z parameters $\boldsymbol{\mu} \in \mathcal{P}$ may still 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 must be used. An alternative approach, which is often used in the control theory community, consists of the construction of several local basis by sampling the parameter space. For a new parameter one proceeds by interpolating the local basis, the local reduced model matrices, or the local reduced model transfer functions. A similar strategy, specific to Weeks framework where the solution is sought only for few frequencies, would be to construct exactly N_z local basis, one per frequency. For a new parameter one would then have to evaluate N_z local reduced models without interpolation. This option is more laborious, but, at the same time, it may result in more stable approximations. We refer the interested reader to [BGW15] and references therein for a detailed comparison of projection-based reduction methods for dynamical systems.

Provided that the symplectic basis (3.43) is constructed, the recovery of the discrete time signals, described in Algorithm 1, can benefit from a computational speed up. For this, we need to replace the high-fidelity solutions in Algorithm 1 with the reduced solutions. Specifically, one needs to replace the full-order system (3.18) with the reduced-order system (3.44) obtained with the cotangent lift method and replace $\tilde{\boldsymbol{u}}_h \in \tilde{V}_{2h}$ with $\tilde{\boldsymbol{u}}_N \in \tilde{V}_{2N}$, defined as

$$\tilde{\boldsymbol{u}}_N(\boldsymbol{x}_i, \tilde{z}; \boldsymbol{\mu}) := \sum_{j=1}^{N} \left(u_j^\alpha(\tilde{z}; \boldsymbol{\mu}) + \imath u_j^y(\tilde{z}; \boldsymbol{\mu})\right) \boldsymbol{\zeta}_j^{cl}(\boldsymbol{x}_i). \tag{3.49}$$

Here, $\boldsymbol{\zeta}_j^{cl}(\boldsymbol{x})$ are the reduced basis functions and $u_j^\alpha(\tilde{z}; \boldsymbol{\mu})$ and $u_j^y(\tilde{z}; \boldsymbol{\mu})$ are the j-th and the $(j+N)$-th entires of the solution of the linear system (3.44), respectively. The high-fidelity Weeks approximation $\hat{\boldsymbol{u}}_h$ is replaced with the reduced counterpart, i.e.,

$$\hat{\boldsymbol{u}}_N(\boldsymbol{x}_i, t^n; \boldsymbol{\mu}) := e^{(\alpha-b)t^n} \sum_{k=0}^{N_z-1} \hat{\boldsymbol{a}}_{k,N} L_k(2bt^n),$$

for $i = 1, \dots, n_s$ and $n = 1, \dots, N_t$, where the coefficients $\hat{\boldsymbol{a}}_{k,N}$ are defined in (3.29) by replacing $\tilde{\boldsymbol{u}}_h$ with $\tilde{\boldsymbol{u}}_N$. Similarly, the approximation of the discrete time history $\hat{\mathbf{g}}_i^m \in \mathbb{R}^{d_\ell \times (N_t+1)}$, obtained by applying Weeks method to the high-fidelity solutions and defined in (3.30), is obtained by replacing $\tilde{\mathbf{u}}_h$ with $\tilde{\mathbf{u}}_N$ in (3.22). We finally remark that both the offline and the online phases of the RB method belong to the database construction phase, which corresponds to the offline phase of the anomaly-detection process (see Figure 2.1).

3.5 Conclusions

In this chapter the mathematical model to efficiently generate synthetic sensor signals is presented. The parametric acoustic-elastic wave equation with an active source is introduced to model the sensor measurements under different environmental and operation conditions. We present the problem both in the time and the Laplace domain. The discretization of the high-fidelity problem in time domain is carried out by means of the finite element method in space and Newmark method in time.

The need to construct a robust synthetic dataset of sensor measurements translates to a many-query problem. To overcome the computational burden associated with the repeated solution of a high-dimensional linear system, we propose a reduced order modeling (ROM) strategy, which combines the translation of the problem to the Laplace domain and the reduced basis (RB) method. Solving the acoustic-elastic wave equation in the Laplace domain allows to reduce the number of times we need to solve the linear space for each parameter $\boldsymbol{\mu}$, while recurring to the RB method allows to reduce the dimension of the linear systems. Weeks method is used to compute the inverse Laplace transform of the problem in frequency domain and thus recover the discrete time signals. The Weeks free parameters are discussed and a problem-dependent strategy to fix them is outlined. We present a POD-Galerkin approach to approximate the FE solution in the frequency domain by exploiting the idea that the parameter dependent solution of the high-fidelity problem can be well approximated by a linear combination of few high-fidelity snapshots. Finally, we rely on the cotangent-lift method, based on a PSD-Galerkin approximation, to construct a stable RB system when one wishes to solve the real counterpart of the complex high-fidelity system, i.e., when the system is split in its real and imaginary components.

CHAPTER 4

ANOMALY DETECTION

The content presented in this chapter is partially based on Sections 4 and 5 of [BH20].

In this chapter, we present data-driven techniques to detect and localize anomalies in a structure by observing healthy configurations only. In particular, Section 4.1 introduces the machine learning techniques commonly used in structural health monitoring, highlighting the distinction between supervised, semi-supervised, and unsupervised learning. Examples from the literature on machine learning algorithms for data-based approaches for SHM are also reported. Motivated by the idea that during its life-time cycle an engineering structure may be affected by many different types and severities of damages, we choose to model only the healthy variations of a structure, i.e., its baseline environmental and operational conditions. For this type of setting, one-class learning methods are used to train the synthetic dataset of healthy configurations. Section 4.2 presents an overview of one-class classification algorithms, focusing on Support Vector Machines, Isolation Forests, Local Outlier Factor, and autoencoders. In Section 4.3 we present our problem formulation, i.e., a semi-supervised learning strategy, based on the synthetic time-signals described in Chapter 3, to both detect, localize, and estimate the severity of damages in a structure of interest. There, we explain how damage localization and damage extent can be assessed by training as many one-class classifiers as the number of sensors. The process of extracting damage-sensitive features from the raw measurements and the consequent dimensionality reduction strategies are also discussed. The classification results obtained with the different one-class learning algorithms for 2D and 3D structures are given in Section 4.4. A comparison of the results is also provided and conclusions are given in Section 4.5.

4.1 An overview of machine learning techniques for SHM

Statistical pattern recognition techniques and machine learning tools have been widely used to generate knowledge from past experiences, focusing on the predictions of new data [Bis06]. The successful real-life examples of machine learning are countless, ranging from image recognition, traffic prediction, speech recognition, or medical diagnosis to name a few. Machine learning algorithms provide a mathematical approach to associate data with class labels by detecting patterns and structures within the data. In the SHM framework, given a dataset of sensor data, the aim of machine learning algorithms is to transform the data into useful information to assess structural damages and make informed decisions. This is achieved by associating sensor data from the structure of interest with some damage state or *class*. When monitoring a structure over time, the most important distinction is between healthy and damaged states, but one may also be interested in introducing more classes, e.g., different types or severities of damage. The idea behind these methods is to *learn* the unknown relationship between some features, extracted from data acquired from the structure measurements, and the state of damage of the structure to map a new observation to the corresponding configuration label. Based on the availability of labels in the sensor data, i.e., the *training data*, we can distinguish three learning setups: (*i*) supervised, (*ii*) semi-supervised, and (*iii*) unsupervised learning. *Supervised learning* refers to the standard classification task where the training data are available from both the healthy and damaged structures. Multi-class supervised learning algorithms can be used to distinguish between multiple types of damages. In *semi-supervised learning*, the training data are solely composed of samples belonging to the undamaged structures. These algorithms, often called anomaly or novelty detection, aim at constructing a model for a single class, i.e., the positive class, to test if a new observation is consistent with that class. The process of learning the intrinsic relationship among data when no labeling information is available is called *unsupervised learning*. Clustering techniques usually belong to this last category, for which there is no sharp distinction between the training and the test phase. We note that the common assumption behind both semi-supervised and unsupervised algorithms is that only a small fraction of the data is abnormal, which makes the separation between these two learning algorithms a thin line.

A variety of data-driven approaches, based on machine learning techniques, have been proposed in structural health monitoring. We refer to [FW12] for a review on machine learning prospectives of SHM. Among the data-driven approaches based on supervised learning algorithms we mention [TPYP18; LAW15; KKW20]. In [TPYP18] the objective of detecting added mass at two predefined locations in a microtruss system is addressed as a two-stage classification process based on synthetic training data. There, the authors compare the accuracy in the performance of classifying both synthetic and experimental test data using several supervised learning algorithms, i.e., one-vs-all Support Vector Machines, decision trees, artificial neural networks, k-nearest neighbors, and nearest-mean classifier. Aerial vehicle sensor data are used in [LAW15; KKW20] to train a data-driven classifier with the goal of estimating the vehicle capabilities in response to possible structural damages or degradations. In [LAW15], several damage scenarios and their associated impacts on the vehicle flight envelope are used to train a vehicle behavioral library in the Bayesian sense, where the goal is to minimize the probability of misclassification. Similarly, in [KKW20], a supervised learning strategy is used to infer which model within a dictionary of models is the best candidate to explain the sensor measurements. By relying on a component-based reduced model approach, the authors include in the training set several regions of damage of varying degrees of severity across the aircraft wings and use them to train an optimal tree classifier.

The anomaly detection approach in which training data belong to undamaged scenarios only is for example presented in [DSC07; LB14; QLAC$^+$19]. A preliminary application of one-class SVMs based on time-frequency information is offered in [DSC07], where the goal is to identify the changes in the signature of the propagating wave to assess damages in composite panels. A semi-supervised approach to detect damages in an steel frame laboratory structure is proposed in [LB14], where experimental sensor measurements are used to train a one-class SVM. In recent work [QLAC$^+$19], proper orthogonal decomposition (POD) modes associated with an undamaged structure are used to calibrate a clustering technique to differentiate between damaged and undamaged scenarios by comparing the projected field of interest.

In this thesis we focus on semi-supervised techniques under the assumption that it would be unreasonable to describe all types of damages and that representing only some damaged configurations would lead to a bias towards certain types and therefore to possible misdetections with high probability. In the simulation-based framework used in this thesis, this choice also accommodates a reduced computational effort. On one hand we can rely on a smaller synthetic dataset which needs to incorporate the different environmental and operational baseline conditions without including the damage scenarios, i.e., fewer online model evaluations are needed to construct a robust dataset. On the other hand, we consider a less complex parametric model, including only the parameters that define the healthy configurations. Thus, the reduced model is also simplified, i.e., fewer basis are needed to describe the reduced model. In the following section we provide a brief description of one-class classification methods in general and focus on the mathematical details of oc-SVM, Isolation Forests, Local Outlier Factor algorithm, and autoencoders.

4.2 One-class classification methods

Anomaly (or novelty) detection indicates the task of identifying substantial differences, the *outliers*, in the test dataset when compared to the data available during training [PCCT14]. An intuitive definition of outliers is given in [Haw80], where an outlier is considered as “an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism”. Anomaly detection methods are usually applied in 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 [DLBM14]. In the context of simulation-based anomaly detection, the combination of these three arguments together with the reasoning provided in Section 4.1 motivate us to opt for a one-class classification approach.

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. Indeed, classic supervised learning algorithms, where every different damage type is associated with a different categorical class, are here replaced with semi-supervised learning techniques, where only healthy states are used to train *one-class classifiers*. 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 anomaly (or novelty) score. Samples whose characteristics deviate too much from those of the positive class will be classified as damaged (outliers) with the idea that they were generated by a different mechanism than the healthy samples, which will instead present characteristics similar to the training data and will therefore be classified as undamaged (inliers). In general, an *ad-hoc* threshold θ is used to distinguish between the two categories, i.e.,

$$\begin{cases} \text{score}(\boldsymbol{x}) \geq \theta & \text{outlier}, \\ \text{score}(\boldsymbol{x}) < \theta & \text{inlier}, \end{cases} \tag{4.1}$$

where $\boldsymbol{x}$ is a measurement belonging either to a healthy or a damaged structure[1] and score($\boldsymbol{x}$) is the anomaly score associated with that sample, i.e., the output of the one-class classifier.

Among many options, see e.g., the reported summaries in [DLBM14; PCCT14; GU16; AGA13], we highlight a few well-known strategies: the one-class Support Vector Machine (oc-SVM) [SWS$^+$00; CS02], the Isolation Forest [LTZ08], based on the principles of the random forest method, and the Local Outlier Factor (LOF) [BKNS00], a nearest-neighbor based 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

---

[1] In this chapter, we use $\boldsymbol{x}$ to represent a generic input for the classifier, but this should not be confused with the sensor spatial location $\boldsymbol{x}_i \in \Omega$ for $i = 1, \ldots, n_s$. In the numerical experiments, $\boldsymbol{x}$ will be replaced with the simulated feature vector $\mathbf{y}_i$, defined in (4.17).

notoriety in the framework of anomaly detection, see, e.g., [JMG95; MVE$^+$15; NJL$^+$18].

In particular, we present the details of oc-SVMs, Isolation Forest, Local Outlier Factor, and autoencoders in Sections 4.2.1, 4.2.2, 4.2.3, and 4.2.4 respectively.

4.2.1 One-class Support Vector Machines

The one-class Support Vector Machines (oc-SVMs) method is derived as a simple modification of the well-known supervised SVM method [CS02], already used in several SHM applications, see, e.g., [HA03; TPYP18]. 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 [Vap98]. Given $\mathcal{X}$, the set of the input training data, and $\mathcal{F}$, the feature space of dimension greater than $\mathcal{X}$, the idea behind this method is known as the *kernel trick*, i.e., the transformation function $\Phi : \mathcal{X} \to \mathcal{F}$ 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

$$k(x, y) := \Phi(x) \cdot \Phi(y). \tag{4.2}$$

A common choice is the Gaussian kernel

$$k(x, y) := \exp\left\{-\frac{\|x-y\|^2}{\hat{\sigma}^2}\right\}, \tag{4.3}$$

where $\hat{\sigma} \in \mathbb{R}_+$ is a free parameter and $\|x-y\|^2$ is the dissimilarity measure.

One-class SVMs, introduced in [SWS$^+$00; SPST$^+$01], 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[2]. Let $p : \mathbb{R}^{d_y} \to \mathbb{R}$ be the separating hyperplane defined as

$$p(\boldsymbol{x}) := w \cdot \Phi(\boldsymbol{x}) - \hat{b}, \tag{4.4}$$

where $w \in \mathcal{F}$ is a vector perpendicular to the hyperplane and $\hat{b}_i \in \mathbb{R}$ is a bias term, obtained by training N_{tr} d_y-dimensional feature-valued samples. Then, the oc-SVM algorithm returns a function $f : \mathbb{R}^{d_y} \to \{-1, 1\}$ that evaluates every new data point to determine on which side of the hyperplane it falls in features space, i.e. it takes values $+1$ in the region capturing most of the data and -1 elsewhere. Hence, in the semi-supervised

---

[2] This formulation is adopted in softwares like Matlab [MAT19] or the Python package `scikit-learn` [PVG$^+$11] and its mathematical formulation is described below. However, an alternative approach is proposed in [TD04], where the goal is to find the minimum hypersphere that best encloses the training data.

setting, the decision function

$$f(\boldsymbol{x}) := \operatorname{sgn}(p(\boldsymbol{x})) \tag{4.5}$$

takes values $+1$ for most of the training samples and values -1 for outliers. The parameters describing the hyperplane (4.4) are obtained by solving the following minimization problem during the training phase

$$\min_{w,\hat{b},\xi_m} \frac{\|w\|^2}{2} + \frac{1}{\hat{\nu} N_{tr}} \sum_{m=1}^{N_{tr}} \xi_m - \hat{b}$$
$$\text{subject to: } w \cdot \Phi(\boldsymbol{x}_m) \geq \hat{b} - \xi_m, \quad \xi_m \geq 0, \quad \text{for } m = 1, \ldots, N_{tr},$$

where $\|w\|$ is a SV type regularization term[3] and $\xi_m \in \mathbb{R}$ are non-zero slack variables that allow soft margins for $m = 1, \ldots, N_{tr}$. Large values of ξ_m allow the m-th data point to lie on the wrong side of the decision boundary and thus those points are considered as outliers. The tradeoff between the number of misclassified training examples and the smoothness of the margin, identified by w, is controlled by the regularization parameter $\hat{\nu} \in]0, 1]$.

The problem can be transformed to a dual form using Lagrangian multipliers and the kernel trick (4.2) as

$$\min_{\boldsymbol{\alpha}} \sum_{m,n=1}^{N_{tr}} \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$
$$\text{subject to: } 0 \leq \alpha_m \leq \frac{1}{\hat{\nu} N_{tr}}, \forall m = 1, \ldots, N_{tr} \quad \text{and} \quad \sum_{m=1}^{N_{tr}} \alpha_m = 1,$$

where the non-zero α_m are the SVs. The latter are required to evaluate any new datum using the SV expansion of the hyperplane (4.4), which becomes

$$p(\boldsymbol{x}) = \sum_{m=1}^{N_{tr}} \alpha_m k(\boldsymbol{x}, \boldsymbol{x}_m) - \hat{b}.$$

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 [SWS+00]. 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. Varying $\hat{\nu}$ controls the trade-off between ξ and $\hat{b}$: 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 [AGA13; KZW+14], a continuous outlier score reveals more information

---

[3] Support vectors (SVs) are data points that are closer to the hyperplane and influence its position and orientation. For more details we refer to [CS02].

than a simple binary label as the output (4.5). Indeed, the absolute value of $p(\boldsymbol{x})$ 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 (4.5). The choice of using (4.5), or another anomaly score based on the hyperplane $p(\boldsymbol{x})$ as a decision strategy depends on the importance given to misclassification errors, i.e., false negative and false positive predictions. False negative predictions, also called false alarms, arise when a healthy structure is classified as damaged and, vice-versa, false positive predictions when damaged structures are classified as healthy. Ideally, one would like to keep both rates low, but from a practical perspective, depending on the engineering application, to keep one of the two errors low, the other one will increase. This trade-off translates into the choice of 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[4].

A relative approach is applied here to compute the anomaly score, i.e., we follow the strategy presented in [AGA13], where, given $\hat{p}$, the maximum distance between the training data and the decision boundary, i.e., $\hat{p} := \max_m p(\boldsymbol{x}_m)$, the anomaly score reads

$$\operatorname{score}(\boldsymbol{x}) := 1 - \frac{p(\boldsymbol{x})}{\hat{p}}. \tag{4.6}$$

Therefore, the points classified as 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 (4.3) may severely impact the generalization performance of oc-SVMs. Indeed, an inappropriate choice of $\hat{\sigma}$ may lead to overfitting (small values) or underfitting (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 [Unn03], a geometry based approach in [KHSM11], a tightness detection strategy, based on the spatial locations of the interior and edge samples [XWX15; AKM$^+$17], and an approach based on the Fisher linear discrimination [WXLZ03]. The first three strategies are observed to be equivalently successful to detect various damage scenarios on a laboratory structure in [LB14]. 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 maximum distance between the two least similar training points [KHSM11]. This strategy is used also in this thesis,

---

[4] For example, in Matlab, this threshold can be adjusted by prescribing a value to the `OutlierFraction` option to the function `fitcsvm`, which implements a post-processing modification to guarantee a specified fraction of outliers in the training set, regardless of the smoothness of the class distribution.

where the kernel factor becomes

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

and $\hat{d}$ is the Euclidean distance between the two least similar training points in the training set.

Classification results in the SHM context and based on the oc-SVMs are presented in Section 4.4.3.

4.2.2 Isolation Forest

Isolation Forest (IF) is a one-class classification algorithm based on an ensemble of decision trees, which stratify the feature space recursively into sub-regions. First presented in [LTZ08], they rely on a fundamentally different model-based method with respect to most of the existing approaches to anomaly detection: instead of constructing a profile of normal instances and then identifying instances that do not conform to the normal profile, they explicitly isolate anomalies. In the remaining of this section we briefly introduce the mathematical background of IFs and define the corresponding anomaly score. We refer the reader to [LTZ08; HKB19] for a more in-depth description.

Different to supervised Random Forest, where partitions in the feature space are chosen to minimize a cost function, to create a branch in the IF setup, the features are randomly selected. Partitions are here generated by selecting a random attribute and then randomly selecting a split value according to a uniform distribution between the minimum and maximum value of the selected attribute. An observation is then divided accordingly to this criterion: if it has lower value of this feature than the random split value, it follows the left branch, otherwise the right one. This process is continued until the given observation is isolated from the rest of the observations or a specified maximum depth is reached. By representing the feature space partitioning with a tree data structure, the number of partitions required to isolate a point correspond to the *path length* from the root to the leaf of the tree. Under the assumption that anomalous observations are few and significantly different from the regular observations, the key principle of IF is that outliers should be identified closer to the root of the tree as compared to normal points. Indeed, an outlier will be characterized by fewer random splits than an inlier, i.e., the number of partitions needed to isolate a sample are fewer for an outlier. A forest of isolation trees, i.e., an isolation forest, is generated by a set of isolation trees with different sets of partitions. The path length, averaged over the isolation forest, is a measure of normality used to define the anomaly score of an instance $\boldsymbol{x}$, i.e.,

$$\mathrm{s}_n(\boldsymbol{x}) := 2^{-\frac{\mathbb{E}[h(\boldsymbol{x})]}{c(n)}},$$

where n is the number of training data, $h(\boldsymbol{x})$ is the path length of an isolation tree and $c(n)$ is the average path length of unsuccessful searches in a Binary Search Tree (BST). As $c(n)$ is a normalization constant approximating the average of $h(\boldsymbol{x})$. When an instance has an anomaly score close to 1, they are likely to be outliers. On the contrary, instances with anomaly scores smaller than 0.5 are regarded as inliers. Similar to the oc-SVM, we propose a relative anomaly score value

$$\text{score}(\boldsymbol{x}) := \frac{s_n(\boldsymbol{x})}{\hat{s}}, \tag{4.7}$$

where $\hat{s} := \max_m s_n(\boldsymbol{x}_m)$ and $\boldsymbol{x}_m, m = 1, \ldots, N_{tr}$ are the training samples. Hence, an anomaly score value larger than 1 indicates an outlier, while a value below 1 reflects an inlier.

We observe that there are only two hyper-parameters in this method: the number of trees to build the forest and the sub-sampling size. In [LTZ08] it is shown that the detection performance converges quickly with a small number of trees and that sub-sampling provides a favourable environment for Isolation Forest. Indeed, since the training data mostly belong to the positive instances, a partial model suffices to achieve high detection performances without isolating all normal points. In the numerical experiments presented in this thesis we choose with a sampling size of 250 and a forest composed of 100 trees. We observe that the method in [LTZ08] is designed for an unsupervised setting where few anomalies are expected in the training set. Nevertheless, the authors also address the scenario when only normal instances are used in the training phase, thus providing a direct method for our semi-supervised setup. Finally, we note that an extension to IF is proposed in [HKB19] to overcome artefacts generated with the criterion for branching operation, typical of the decision trees. By allowing hyperplanes with random slopes to separate the features, the Extended Isolation Forest (EIF) method provides a more robust algorithm and remedy to these artefact.

Classification results in the SHM context and based on the IF are presented in Section 4.4.3.

4.2.3 Local Outlier Factor

The Local Outlier Factor (LOF) algorithm, introduced in [BKNS00], is used to detect outliers in a dataset by measuring the local deviation of a given sample with respect to its neighbors. The algorithm is based on a concept of *local* density of an object with respect to its neighbors. This allows to distinguish regions of similar densities from regions with substantial lower densities. The points in the former correspond to inliers and the points in the latter to outliers, which appear isolated with respect to the surrounding neighborhood. This strategy overcomes the limitations of using a global density measure to detect outliers, which may fail in detecting a certain type of outliers when the data structure is complex, see e.g., [KN98].

Following [BKNS00], we introduce the formal definitions to characterize the LOF method. Throughout this section we use o, p, q to denote objects in a set $\mathcal{D}$ and k is a positive integer.

Definition 4.1 *k-distance of an object p.* The k-distance of object p, denoted as $d_k(p)$, is defined as the distance $d(o, p)$ between p and an object $o \in \mathcal{D}$ such that:

(i) for at least k objects $o' \in \mathcal{D} \backslash \{p\}$ it holds that $d(p, o') \leq d(p, o)$, and
(ii) for at most $k-1$ objects $o' \in \mathcal{D} \backslash \{p\}$ it holds that $d(p, o') < d(p, o)$.

Definition 4.2 *k-distance neighborhood of an object p.* Given the k-distance of p, the k-distance neighborhood of p is defined as the set of objects whose distance from p is not greater than the k-distance, i.e.,

$$N_k(p) := \{q \in \mathcal{D} \backslash \{p\} | d(p, q) \leq d_k(p)\}.$$

The objects q in $N_k(p)$ are called the k-nearest neighbors of p.

Definition 4.3 *Reachability distance of an object p with respect to object o.* The reachability distance of object p with respect to object o is defined as

$$R_k(p, o) := \max \{d_k(o), d(p, o)\}.$$

Intuitively, if an object o is far away from p, then the reachability distance will simply be the actual distances between the two points. When instead the two points are close, the actual distance is replaced by the $k-$distance of o. In this way the statistical fluctuations of $d(p, o)$ for all o' close to p are reduced. Moreover, the higher the value of k, the more similar the reachability distances for objects within the same neighborhood. The local reachability density of p is then defined as the inverse of the average reachability distance among the k-nearesr neighbors of p. Formally it is defined as follows.

Definition 4.4 *Local reachability density of an object p.* The local reachability density of object p is defined as

$$LR_k(p) := \left(\frac{\sum_{o \in N_k(p)} R_k(p, o)}{|N_k(p)|} \right)^{-1}.$$

Ultimately, the LOF of p is defined as the average of the ratio of the local reachability density of p and those of the k-nearest neighbors of p. Formally, using the previous definitions, the LOF is defined as

Definition 4.5 *Local outlier factor of an object p.* The local outlier factor of object p is defined as

$$LOF_k(p) := \frac{1}{|N_k(p)|} \sum_{o \in N_k(p)} \frac{LR_k(o)}{LR_k(p)}.$$

This coefficient provides an information on the *degree of outlierness* of an object p, different from other distance-based and density-based methods, see, e.g., [KN98], where the notion of outlier is considered as a binary property. Intuitively, the lower the local reachability distance of p and the higher the local reachability densities of its k-nearest neighbors, the higher is the LOF_k value of p.

The following lemma shows that for most objects p in a cluster $\mathcal{D}$, the LOF of p is approximately equal to 1.

Lemma 4.1 *Property of local outliers [BKNS00].* Let R_{min} and R_{max} denote the minimum and maximum reachability distance of objects in $\mathcal{D}$, i.e.,

$$R_{min} := \min_{p,q \in \mathcal{D}} R_k(p,q) \quad \text{and} \quad R_{max} := \max_{p,q \in \mathcal{D}} R_k(p,q),$$

respectively. Then for all objects $p \in \mathcal{D}$, such that all the k-nearest neighbors q of p are in $\mathcal{D}$, and all the k-nearest neighbors o of q are also in $\mathcal{D}$, it holds that

$$\frac{1}{1+\varepsilon} \leq LOF_k(p) \leq 1 + \varepsilon,$$

where $\varepsilon := \frac{R_{max}}{R_{min}} - 1$.

For tight clusters, we expect the $k-$nearest neighbors q of p to be in $\mathcal{D}$ as well as the $k-$nearest neighbors o of q. Hence, ε will be quite small and most objects in $\mathcal{D}$ will have LOF_k close to 1. Values significantly larger than 1 indicate less dense regions and thus outliers.

This approach can be extended to a semi-supervised setting by using the relative LOF value for a new measurement $\boldsymbol{x}$, i.e., the anomaly score function in (4.1) becomes

$$score(\boldsymbol{x}) := \frac{LOF_k(\boldsymbol{x})}{\hat{L}_k}, \tag{4.8}$$

where $\hat{L}_k = \max_m LOF_k(\boldsymbol{x}_m)$ and $\boldsymbol{x}_m$ are the training samples for $m = 1, \dots, N_{tr}$. A less conservative approach is to replace $\hat{L}_k$ with the LOF value corresponding to the highest 5% or 10% of LOF values of the training samples. In this way, the false negative rate will decrease, at the expenses of a possible increase in the false positive rate. In both cases the threshold to distinguish inliers from outliers is set to $\theta = 1$.

Finally, we observe that the authors in [BKNS00] provide a practical guideline on how to select a range $[k_{min}, k_{max}]$ for the parameter k. By computing the LOF values within this range, they propose an heuristic for ranking objects by their maximum LOF value, i.e.,

$$k := \operatorname*{arg\,max}_{j \in [k_{min}, k_{max}]} LOF_j(p).$$

Alternative aggregates, such as the minimum or the mean could be also used. In this thesis, after training the model with different values of the hyper-parameter k, we choose $k = 7$.

Classification results in the SHM context and based on the Local Outlier Factor method are presented in Section 4.4.3.

4.2.4 Autoencoders

Autoencoders are a type of artificial neural networks (ANN) that aim to learn the identity map: their output is an accurate approximation of the input [GBCB16]. If an autoencoder succeeds in simply learning to copy the input data to themselves, then it is not particularly useful. Instead, autoencoders are trained to learn an *imperfect* identity map. By doing this in an unsupervised manner, autoencoders learn few underlying salient features that characterize the training data by a dimensionality-reduction procedure that comprises two tasks: data compression and data recovery. These two objectives are achieved by employing a pair of two connected neural networks: an *encoder* f that maps the original space $\mathcal{X}$ into a latent space of lower dimension $\mathcal{F}$, and a *decoder* g to map back to $\mathcal{X}$, such that their composition $g \circ f$ is *close to* the identity on the input. Autoencoders have been applied to various non-intrusive model reduction methods based on data-driven techniques. For example, long short-term memory (LSTM) autoencoders are a type of recurrent neural networks (RNNs), successfully used in the context of speech recognition or text translation, see e.g., [GJM13]. A modular model consisting of a deep convolutional autoencoder and a modified LSTM network is proposed in [GB18] to predict the dynamics of fluid systems with large parameter-variations. A hierarchical dimensionality reduction based on autoencoders and principal component analysis (PCA) to learn the dynamics and consequently recovering computational fluid dynamics data is proposed in [CJK$^+$19]. Reference [LC20] proposes a framework for projecting dynamical systems onto arbitrary non-linear manifolds where convolutional autoencoders are used to compute a representative low-dimensional non-linear manifold.

In the context of anomaly detection, autoencoders exploit the idea that the training data (positive examples) are approximately distributed on a low-dimensional non-linear manifold, parametrized my means of the so-called latent variables, learned by redundancy compression. Therefore, for a given data point $\boldsymbol{x}$, the autoencoder is trained to output a value $g(f(\boldsymbol{x}))$, which is expected to belong to the manifold of healthy configurations in virtue of the common nature of normal data. Under the assumption that positive instances are expected to be reconstructed accurately, while negative instances, i.e., abnormal data, are not, the anomaly score is based on the reconstruction error, e.g., the mean squared error of the difference between a new datum and its reconstruction, i.e.,

$$s(\boldsymbol{x}) := \|\boldsymbol{x} - g(f(\boldsymbol{x}))\|^2.$$

Similar to the other one-class classification methods presented in this thesis, one can choose a relative anomaly score value which is scaled with the highest reconstruction error obtained for the training samples, i.e.,

$$\text{score}(\boldsymbol{x}) := \frac{s(\boldsymbol{x})}{\hat{s}}, \tag{4.9}$$

where $\hat{s} = \max_m s(\boldsymbol{x}_m)$ and $\boldsymbol{x}_m$, $m = 1, \dots, N_{tr}$ are the training samples. A new sample with score larger than 1 will be classified as an outlier, and otherwise as an inlier. The main advantage of using a reconstruction-based anomaly detection approach like the autoencoders is that specific engineering-based damage indicator features do not need to be specified by the user, different from others one-class methods mentioned above. Instead, by learning the features which suffice to describe and reconstruct the input, autoencoders provide a purely data-driven feature extraction method. Hence, raw measurements such as sensor time-signals can be used directly.

Autoencoders have recently been used for structural health monitoring applications in a supervised learning framework to detect and localize damages. We refer to [NLL$^+$19], where a deep autoencoder is trained to remove redundancy in the input data in a supervised learning setting. The damages in a laboratory steel frame structure are then identified by a regression layer that maps the low-dimensional inputs to the structural stiffness parameters, used as a proxy for damage detection. The training set is composed of synthetic natural frequencies and mode shapes generated from both undamaged and damaged scenarios. In a recent work presented in [MLNM20], a variational autoencoder (VAE) is used to compress the input data and detect the damage of a beam-like bridge subjected to a moving load. The authors consider a multiple damage scenario with two cracks, and a moving-window method is proposed to locate the cracks in the bridge. Only the response data from the damaged structure is used to train the VAE model, without any information from the healthy structure. To our knowledge, no attempt has been made to use autoencoders in a semi-supervised learning setting for SHM.

In the remaining of this section we briefly present the mathematical setting that defines autoencoders and variational autoencoders, and we refer the interested readers to Chapter 14 of [GBCB16] and references therein for a detailed overview.

An autoencoder is a feedforward neural network that aims to learn the identity map, i.e., $h : \mathcal{X} \to \mathcal{X}$ with $h(\boldsymbol{x}) \simeq \boldsymbol{x}$. As mentioned above, this type of networks consists of two parts: an encoder function $f : \mathcal{X} \to \mathcal{F}$ with $\dim(\mathcal{F}) \leq \dim(\mathcal{X})$ that maps a high-dimensional vector $\boldsymbol{x}$ to a low-dimensional *code* $\hat{\boldsymbol{x}}$; and a decoder $g : \mathcal{F} \to \mathcal{X}$ that produces a reconstruction $g(\hat{\boldsymbol{x}}) = \tilde{\boldsymbol{x}}$ of the original high-dimensional vector. The resulting autoencoder is obtained by composition, i.e., $h = g \circ f$. The encoder and decoder are classic neural networks with one input layer, L hidden layers, and one output layer. An encoder with L_f hidden layers takes the form

$$f(\boldsymbol{x}; \boldsymbol{\theta}_f) = s^{(L_f)}\left(\cdot; \boldsymbol{\theta}_f^{(L_f)}\right) \circ s^{(L_f-1)}\left(\cdot; \boldsymbol{\theta}_f^{(L_f-1)}\right) \circ \cdots \circ s^{(1)}\left(\boldsymbol{x}; \boldsymbol{\theta}_f^{(1)}\right),$$

where $s^{(l)}\left(\cdot; \boldsymbol{\theta}_f^{(l)}\right) : \mathbb{R}^{N^{(l-1)}} \to \mathbb{R}^{N^{(l)}}$ denotes the function applied at layer l of the neural network and $N^{(l)}$ represents the dimension of the output layer l for $l = 1, \ldots, L_f$. The input has dimension $\dim(\mathcal{X})$ and the output $\dim(\mathcal{F})$. Here, $\boldsymbol{\theta}_f = \left[\boldsymbol{\theta}_f^{(1)}, \ldots, \boldsymbol{\theta}_f^{(L_f)}\right]$ represents all the parameters of the encoder network, with $\boldsymbol{\theta}_f^{(l)}$ being the subset of parameters employed at level l for $l = 1, \ldots, L_f$. In general, $s^{(l)}$ can be any parametrized function, but in this work we consider $s^{(l)}$ to model the action of a single-layer perceptron described by the set of weights and biases $\boldsymbol{\theta}_f^{(l)}$. Similarly, a decoder with L_g layers is constructed as follows

$$g(\hat{\boldsymbol{x}}; \boldsymbol{\theta}_g) = s^{(L_g)}\left(\cdot; \boldsymbol{\theta}_g^{(L_g)}\right) \circ s^{(L_g-1)}\left(\cdot; \boldsymbol{\theta}_g^{(L_g-1)}\right) \circ \cdots \circ s^{(1)}\left(\hat{\boldsymbol{x}}; \boldsymbol{\theta}_g^{(1)}\right),$$

with $\boldsymbol{\theta}_g = \left[\boldsymbol{\theta}_d^{(1)}, \ldots, \boldsymbol{\theta}_d^{(L_d)}\right]$. For fully-connected layers, i.e., for the classic multilayer perceptron (MLP), one has

$$s_j^{(l)}\left(\boldsymbol{x}^{(l-1)}; \boldsymbol{\theta}^{(l)}\right) = \phi\left(\sum_{i=1}^{N^{(l)}} \boldsymbol{w}_{i,j}^{(l)} \boldsymbol{x}_i^{(l-1)} + \boldsymbol{b}_i^{(l)}\right), \tag{4.10}$$

where ϕ is a non-linear activation function, $N^{(l)}$ is the number of neurons in the layer l, $\boldsymbol{w}^{(l)}$ and $\boldsymbol{b}^{(l)}$ are the weights and bias at layer l, respectively. The values of the vector at layer l are obtained from the values at the previous level simply using the feedforward update:

$$x_j^{(l)} = s_j^{(l)}\left(\boldsymbol{x}^{(l-1)}; \boldsymbol{\theta}^{(l)}\right) \quad \text{for} \quad j = 1, \ldots, N^{(l)}.$$

For the encoder, $N^{(0)} = \dim(\mathcal{X})$ and $N^{(L_f)} = \dim(\mathcal{F})$. Conversely, for the decoder, $N^{(0)} = \dim(\mathcal{F})$ and $N^{(L_g)} = \dim(\mathcal{X})$. Alternatively, convolutional layers can replace fully connected layers as in MLP autoencoders, thus allowing to reduce the number of parameters to be estimated by means of *parameter sharing*. When the amount of training data is limited, convolutional autoencoders are known to better generalize to unseen data, see e.g. [LC20].

Similar to classic feedforward networks, an estimate of the autoencoder parameters $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_g$ is determined in the training phase by minimizing a loss function $\mathcal{L}\left(\boldsymbol{x}, g\left(f\left(\boldsymbol{x}\right)\right)\right)$

that penalizes $g\left(f\left(\boldsymbol{x}\right)\right)$ for being dissimilar from $\boldsymbol{x}$, i.e.,

$$\hat{\boldsymbol{\theta}}_f, \hat{\boldsymbol{\theta}}_g = \underset{\boldsymbol{\theta}_f, \boldsymbol{\theta}_g}{\arg\min} \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{x}_n - g\left(f\left(\boldsymbol{x}_n; \boldsymbol{\theta_f}\right); \boldsymbol{\theta}_g\right)\|^2, \tag{4.11}$$

where $N = \dim(\mathcal{X})$. Traditionally, the stochastic gradient descent, where gradients are computed by back-propagation on minibatches of the training data, is employed to minimize the loss function. Designing an autoencoder whose code dimension is less than the input dimension, forces the autoencoder to capture the most relevant properties of the training data. However, autoencoders may fail to extract useful information if they are allowed too much capacity. To avoid overfitting, regularized autoencoders add a penalty term to the loss function in addition to the reconstruction error term to encourage the model to have other properties, such as sparsity or robustness to noise, on top of the ability to copy the input to itself.

In addition to regularized autoencoders, generative models can be seen as a particular type of autoencoders. Without going into too much detail, we mention variational autoencoders (VAE) [KW13; ABY$^+$16] and Generative Adversarial Networks (GAN) [GPAM$^+$14]. Variational autoencoders are autoencoders with a continuous latent space, allowing generation of new data similar to that observed during the training phase by sampling from the latent space. This is achieved by designing an encoder stochastic model that returns a distribution over the latent space instead of a single vector, e.g., for a Gaussian distribution, the code may be defined by a mean and a standard deviation vectors. Then, the input of the decoder model is sampled from such distribution. In other words, VAEs infer the characteristics of the code $\hat{\boldsymbol{x}}$ by observing the input $\boldsymbol{x}$, i.e., the goal is to compute the posterior distribution $p(\hat{\boldsymbol{x}}|\boldsymbol{x})$ and variational inference is applied to estimate this intractable distribution.

In the numerical experiments we use a variational autoencoder with an LSTM encoder and an LSTM decoder. The input (and the output) of the autoencoder are d_ℓ-dimensional vectors with $\hat{N}_t = 200$ time steps, chosen as a sub-sample of the original $N_t = 30'000$ steps, as explained in Section 4.4.1. In particular, we consider an encoder network with one hidden layer with 100 neurons and an output layer with 20 neurons. In a symmetric way, the decoder network has an input layer of 20 neurons, one hidden layer with 100 neurons and an output layer of size to $d_\ell \times \hat{N}_t$ to match the input of the encoder. We employ the hyperbolic tangent as the activation function ϕ in (4.10) and, to compute the parameters (4.11), we use an Adam optimizer with a learning rate equal to 0.001 for 1000 epochs. We refer to [GBCB16] for an in-depth description of these technical details. Classification results based on the LSTM VAE are presented in Section 4.4.3. We acknowledge that other architectures and other activation functions, e.g., the sigmoid or the ReLU functions, could lead to different or improved results compared to the ones presented here.

4.3 Local semi-supervised method for damage detection and localization

In this section we explain how the various anomaly detection methods described in the previous sections can be used to detect structural anomalies in a semi-supervised setting and in particular we connect these models to the synthetic time signals defined in Chapter 3. We also present a dimensionality reduction strategy based on ad-hoc feature extraction used as a pre-training phase for the synthetic sensor measurements.

The (one-class) classification of multidimensional time signals captured by different sensor units is often a challenging machine learning task with a wide variety of applications, including audio and speech recognition, seismology, and seizure classification from electroencephalography among others. In structural health monitoring, sensor measurements often correspond to displacements, accelerations, or strains. In particular, when relying on a guided-wave monitoring approach as in this thesis, sensor measurements correspond to multi-dimensional time signals. In machine learning, dealing with high-dimensional training datasets often leads to good performances on the training data, but poor generalization to new observations, a phenomenon called overfitting. When considering a limited number of samples, to avoid the curse of dimensionality in one-class as in standard binary classification, it is fundamental to express multi-dimensional data with few variables extracted from the signals. The process of extracting features which are on one hand sensitive to damages and on the other hand robust to noise and healthy variations is discussed in Section 4.3.1. Further dimensionality compression by means of principal component analysis (PCA) is presented in Section 4.3.2.

For sensor data, there exist two main approaches for combining the information collected at different sensor locations: feature-level fusion and decision-level fusion [GB01; PZÖ$^+$18]. The former combines data after feature extraction and considers one global classifier (sensor independent), thus exploiting the correlations across sensors. On the other hand, in decision-level fusion, the signals are classified for each sensor location by a local classifier (sensor dependent) and the results are then combined into a decision output. We note that, for the second strategy, the anomaly score functions and the corresponding thresholds may be sensor dependent. In the context of anomaly detection, the two strategies are summarized in Figure 4.1. While the superiority of one method over the other one depends strongly on the problem at hand, it is important to observe that the feature-based fusion approach further increases the problem dimensionality, thus accentuating the aforementioned overfitting problem. The decision-level fusion scheme instead keeps the dimensionality low. Moreover, the decision-level fusion setup facilitates the use of a hierarchical classification approach, where increasing levels of damage identification can be defined. Indeed, by constructing a separate classifier for each sensor location, we gain information on the local state of damage of each sensor, which can then be combined to define a final decision strategy.

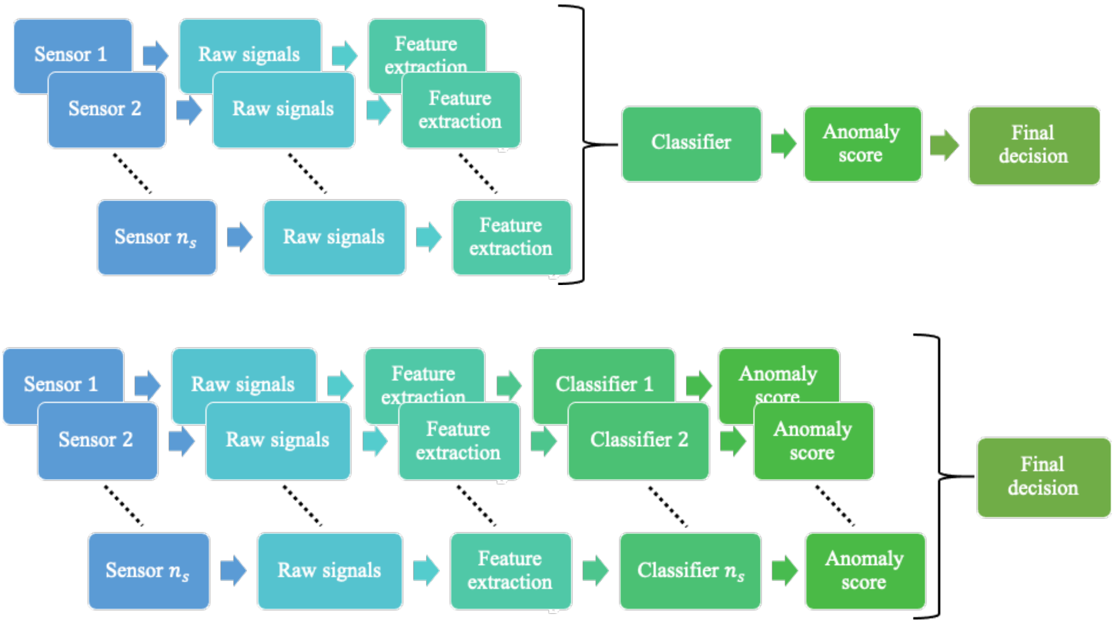


Figure 4.1 – Flowchart to compare the feature-level (*top*) and the decision-level (*bottom*) fusion approaches for the semi-supervised damage detection strategy with multi-dimensional training data captured by n_s sensors.

For the reasons above, in this thesis, we use the decision-level fusion scheme to analyze multi-dimensional sensor data. This approach is also employed in [LB14], where the authors use white noise stimulations to locate damages without including any data belonging to damaged configurations in the offline phase. In practice, for the decision-level fusion approach, we train n_s one-class algorithms, one for each sensor location. Training n_s separate classifiers implies that the global classification model (4.1) is replaced with n_s local detection models

$$\begin{cases} s_i(\boldsymbol{x}_i) \geq \theta_i & \text{damage in the proximity of the } i^{th}\text{sensor}, \\ s_i(\boldsymbol{x}_i) < \theta_i & \text{health in the proximity of the } i^{th}\text{sensor}, \end{cases} \quad \text{for } i = 1, \dots, n_s, \qquad (4.12)$$

where $s_i : \mathbb{R}^{d_y} \to \mathbb{R}$, with d_y is the size of the input vector $\boldsymbol{x}_i$, are the local anomaly score functions, i.e., (4.6) for the oc-SVMs, (4.7) for the IFs, (4.8) for the LOF algorithm, and (4.9) for the autoencoders. In principle, different algorithms could be chosen for different sensor locations, but in this thesis we limit ourselves to one algorithm for all sensor locations. For all methods we consider a relative anomaly score and a fixed threshold $\theta_i := 1$ for all sensors, i.e., for all $i = 1, \dots, n_s$. We finally observe that the training phase of the n_s classifiers belongs to the offline phase depicted in Figure 2.1 and it is low cost since the process is fully parallelizable given that each classifier can be trained independently.

As mentioned above, using local models provides insight into the local properties and ultimately allows to achieve a *multi-level hierarchical identification process.* In [FW12], the authors describe a five-step process to asses the damage state of a system. This process attempts to answer five questions to estimate the existence, the location, the type, the extent, and the prognosis of damage. In this thesis, we consider three hierarchical levels of damage identification: the detection, the extent of damage, and its localization. We note that, different from supervised strategies, it is less clear how to identify multiple types of damage with one-class classification methods without including their labels. For the first level, the outputs of the n_s sensors can be used to guide a binary decision, e.g., one can choose to consider a structure as damaged if at least one of the n_s sensor measurements is classified as an outlier. For example, when the one-class algorithm is the oc-SVMs, this would mean that one among the n_s anomaly scores (4.6) is greater than $\theta = 1$. For the second level, 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., when the anomaly score is much greater than the threshold, and *mild outliers*, i.e., values slightly above the threshold. For this option, one has to introduce an additional threshold value θ_2, which may be difficult to choose by only looking at the anomaly scores of healthy samples. For example, for the oc-SVMs, this could correspond to $\theta_2 = 2$, and strong outliers would be those with anomaly score above θ_2 while mild outliers those with anomaly score between θ and θ_2. For the third level, damage localization can be achieved by conjecturing that damage is expected to be closer to those sensors which are classified as anomalous. Nevertheless, we note that when the damage is severe, many measurements may be classified as outliers, thus preventing localization.

In this work, the raw signals of Figure 4.1 correspond to the synthetic output of interests recovered at each sensor location using the methodology presented in Chapter 3. Precisely, the signals $\hat{\mathbf{g}}_i^m \in \mathbb{R}^{d_\ell \times (N_t+1)}$ are acquired using Algorithm 1 and by including the appropriate modifications to leverage the RB framework described in Section 3.4 for $i = 1, \dots, n_s$ and $m = 1, \dots, N_{tr}$, where $N_{tr} \in \mathbb{N}_+$ is the size of the training set. Then, the n_s synthetic databases $\hat{\mathcal{D}}_i$ are defined as the set of N_{tr} signals, i.e.,

$$\hat{\mathcal{D}}_i \coloneqq \{\hat{\mathbf{g}}_i^m\}_{m=1}^{N_{tr}}, \quad \text{for } i = 1, \dots, n_s. \tag{4.13}$$

The cost associated to the construction of synthetic databases (4.13) includes the solution of $N_z N_{tr}$ reduced systems (3.44) and the computation of $n_s N_z N_{tr}$ expansion coefficients (3.29), where N_z is the number of frequencies used to approximate the inverse Laplace transform with Weeks method as described in Section 3.3.2. We highlight that the number of linear systems to solve is independent of the number n_s of sensors. Indeed, the expansion coefficients in (3.49) are obtained by solving a linear system of size N which depends only on the frequency $\tilde{z}_j$ and the parameter $\boldsymbol{\mu}_m$, and $\tilde{\boldsymbol{u}}_N(\boldsymbol{x}_i, \tilde{z}_j; \boldsymbol{\mu}_m)$ is obtained by evaluating the reduced basis functions at $\boldsymbol{x}_i$ for $i = 1, \dots, n_s$.

Depending on the chosen one-class algorithm, it may be necessary to pre-process the measurements before training the classifiers to overcome the issues related to the curse of dimensionality. In the remainder of this section, we describe data compression by means of ad-hoc engineering-based feature extraction and unsupervised dimensionality reduction in Sections 4.3.1 and 4.3.2, respectively. For each parameter $\boldsymbol{\mu}_m$, we summarize these transformations as

$$\underbrace{[\hat{g}_i(t^0;\boldsymbol{\mu}_m),\dots,\hat{g}_i(t^{N_t};\boldsymbol{\mu}_m)]}_{d_\ell\times(N_t+1)} \xrightarrow{\mathcal{F}} \underbrace{[\mathcal{F}_1(\hat{\mathbf{g}}_i^m),\dots,\mathcal{F}_Q(\hat{\mathbf{g}}_i^m)]}_{d_\ell\times Q} \xrightarrow{PCA} \underbrace{\mathbf{y}_i^m}_{d_y}$$

where the feature functions $\mathcal{F}_j : \mathbb{R}^{d_\ell\times(N_t+1)} \to \mathbb{R}^{d_\ell}$ for $j = 1,\dots,Q$ are defined in (4.15), and $\mathbf{y}_i^m = [\mathcal{F}_1(\hat{\mathbf{g}}_i^m),\dots,\mathcal{F}_Q(\hat{\mathbf{g}}_i^m)]\,\mathbf{P}_{PCA}$ with $\mathbf{P}_{PCA} \in \mathbb{R}^{d_\ell Q\times d_y}$ the matrix whose columns represent the first $d_y < d_\ell Q$ principal component coefficients. In Remark 5 we explain which data compression applies to which one-class algorithm. For some methods the feature extraction boxes in Figure 4.1 comprise both the ad-hoc feature extraction and the unsupervised dimensionality reduction. For others, raw data are used directly or only damage-sensitive features are extracted.

4.3.1 Feature extraction

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 [FW12]. Common choices for the damage-sensitive features can be found for example in [LB14; LHSL02]. In [LB14], the authors use nine values to train multiple one-class classifiers: the crest factor and two different integrals of the transmissibility function, along the three spacial axis. In [LHSL02], thirteen characteristic values are extracted from each of the transient surface response, i.e., the first three relative maximum and minimum values of the response, the corresponding arrival times, and the distance between source and receiver. These features are used to train a supervised learning algorithm based on four outputs, i.e., the existence of crack, the location of the crack with respect to the receiver, the length and the depth of the crack. These choices are motivated by observing that, in the presence of a crack (and in particular for larger depths of the crack-tips), 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, see, e.g., [ZBA13; LHSL02]. Moreover, in [ZBA13], the authors observe phase delay in addition to attenuation in amplitude as the piezoelectric transducers (i.e., the sensors) become more separated.

In this thesis, each of the d_ℓ components of the raw displacement signals $\hat{\mathbf{g}}_i^m$ are processed into a Q-dimensional feature vectors with $Q := 6$ for $i = 1,\dots,n_s$ and $m = 1,\dots,N_{tr}$. We consider the following characteristic values: the crest factor, which indicates how extreme the peaks are in a waveform, the maximum and minimum values of the time response, the corresponding arrival times, i.e., the onset, and the number of peaks and valleys in the signals. For each sample $\hat{\mathbf{g}}_i^m$, the crest factor $\mathbf{c}_i^m \in \mathbb{R}^{d_\ell}$ is defined as

$$\mathbf{c}_i^m := \frac{|\hat{\mathbf{g}}_i^m|_{peak}}{(\hat{\mathbf{g}}_i^m)_{rms}}, \qquad \text{where} \quad \begin{cases} |\hat{\mathbf{g}}_i^m|_{peak} := & \max_n |\hat{g}_i(t^n;\boldsymbol{\mu}_m)| \\ (\hat{\mathbf{g}}_i^m)_{rms} := & \sqrt{\frac{1}{N_t+1}\sum_{n=0}^{N_t}\left(\hat{g}_i(t^n;\boldsymbol{\mu}_m)\right)^2} \end{cases}, \tag{4.14}$$

where $\hat{g}_i(t^n;\boldsymbol{\mu}_m)$ is the n-th entry of $\hat{\mathbf{g}}_i^m$. The arrival time $\mathbf{a}_i^m \in \mathbb{R}^{d_\ell}$, the number of peaks $\mathbf{p}_i^m \in \mathbb{R}^{d_\ell}$ and valleys $\mathbf{v}_i^m \in \mathbb{R}^{d_\ell}$ are defined using the `peakfinder` Matlab function [MAT19].

Precisely, $\mathbf{a}_i^m \in \mathbb{R}^{d_\ell}$ 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 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 $\boldsymbol{\varepsilon}_i$ of the signal values, i.e., $\mathbf{a}_i^m := \arg\min_n \{|\hat{g}_i(t^n;\boldsymbol{\mu}_m)| \geq \boldsymbol{\varepsilon}_i\}$, where $\hat{g}_i(t^n;\boldsymbol{\mu}_m)$ is the n-th entry of $\hat{\mathbf{g}}_i^m$.

To summarize, the feature functions $\mathcal{F}_j : \mathbb{R}^{d_\ell \times (N_t+1)} \to \mathbb{R}^{d_\ell}$ for $j = 1, \dots, Q$ are defined as

$$\begin{gathered} \mathcal{F}_1(\hat{\mathbf{g}}_i^m) := \mathbf{c}_i^m, \quad \mathcal{F}_2(\hat{\mathbf{g}}_i^m) := \max_n \hat{g}_i(t^n;\boldsymbol{\mu}_m), \quad \mathcal{F}_3(\hat{\mathbf{g}}_i^m) := \min_n, \hat{g}_i(t^n;\boldsymbol{\mu}_m), \\ \mathcal{F}_4(\hat{\mathbf{g}}_i^m) := \mathbf{a}_i^m, \quad \mathcal{F}_5(\hat{\mathbf{g}}_i^m) := \mathbf{p}_i^m, \quad \mathcal{F}_6(\hat{\mathbf{g}}_i^m) := \mathbf{v}_i^m, \end{gathered} \tag{4.15}$$

where $\hat{g}_i(t^n;\boldsymbol{\mu}_m)$ is the n-th entry of $\hat{\mathbf{g}}_i^m$ for $m = 1, \dots, N_{tr}$. The feature-based database becomes

$$\hat{\mathcal{D}}_i^{\mathcal{F}_Q} := \{[\mathcal{F}_1(\hat{\mathbf{g}}_i^m), \dots, \mathcal{F}_Q(\hat{\mathbf{g}}_i^m)]\}_{m=1}^{N_{tr}}, \tag{4.16}$$

where each entry has dimension $d_\ell Q$, i.e., we consider a total of $N_f := d_\ell Q \in \mathbb{N}$ features.

We observe that features extracted directly from the raw signals in frequency domain, i.e., before applying Weeks method for reconstruction, are not considered in this thesis. Nevertheless, such features, e.g., the transmissibility defined for example in [LB14], could be also included either by direct extraction for simulated samples or by pre-applying a

Laplace transform for experimental sensor signals, which are typically available only in time domain.

4.3.2 Dimensionality reduction

Among the N_f selected features, dimensionality reduction is often needed to generate robust classifiers. For example, we observe that the oc-SVM strategy does not capture anomalies well if applied directly to the feature-based datasets (4.16). It has been shown, see, e.g., [TM03], that using too many features may introduce excessive noise in the dataset and lead to overfitting. To overcome this problem, one should apply some dimensionality reduction method, e.g., principal component analysis (PCA), or alternatively some *feature selection* strategies, e.g., rank revealing QR (RRQR) algorithm. PCA is a powerful tool, traditionally used in data mining, to reduce the dimensionality of multivariate data by expressing the original features in terms of fewer components, called *principal components*, which are an uncorrelated linear combination of the original variables. The principal components describe the variability of the data observations in terms of a new set of orthogonal directions, which account for a given amount of the variation in the sample. On the other hand, the RRQR decomposition algorithm, first proposed in [GR70], can be applied to a matrix of feature vectors to compute a QR factorization and a pivoting column which provide a strategy to select the most important features. In this method, the original data remain unmodified, but possibly more dimensions need to be retained to guarantee the same variability as PCA.

We note that, in general, this traditional method may 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. There exist several studies, see, e.g., [MMP02; KKAG14], in which the information carried by low-variance directions is emphasized. For example, in [KZW$^+$14] the authors apply three popular dimensionality reduction techniques, i.e., PCA, random projections (RP) and piecewise aggregate approximations (PAA), to supervised and unsupervised SVM for damage detection approaches in SHM. These techniques are applied directly to the original time signals, without extracting damage-sensitive features. In their setup, where the number of features roughly corresponds to the number of time steps, RP results in the best reduction technique in terms of achieving a good balance between reducing the computational time and maintaining the detection accuracy. While for very large datasets RP are known to achieve best performances, see, e.g., [Ach03], in many cases, removing redundant features by projecting the data on the high-variance directions remains beneficial. Hence, given our choice of relatively few features, i.e., $N_f = 12$ as described in Section 4.4, PCA transformation is more appropriate. We note that we use the Matlab function `pca` [MAT19] to perform this task.

To apply PCA to the feature-based databases (4.16), we first normalize the training data

oc-SVM	IF	LOF	VAE
$\hat{\mathcal{D}}_i^{\mathcal{F}}$	$\hat{\mathcal{D}}_i^{\mathcal{F}_Q}$	$\hat{\mathcal{D}}_i^{\mathcal{F}}$	$\hat{\mathcal{D}}_i$

Table 4.1 – Summary of the datasets used to train different one-class algorithms.

so that each feature has zero mean and unit standard deviation. Then, we apply the PCA and store the first d_y principal coefficients in $\mathbf{P}_{PCA} \in \mathbb{R}^{N_f \times d_y}$. The value d_y should be chosen by observing the decay of the PCA eigenvalues: a rapid decay suggests that few components suffice to describe all the features. For the numerical solutions presented in this chapter, for all sensors, we observe a rapid decay of the PCA eigenvalues, which motivates our choice of retaining only one principal component, i.e., $d_y = 1$.

Ultimately, the datasets used to train each one of the n_s one-class classifiers are defined as

$$\hat{\mathcal{D}}_i^{\mathcal{F}} = \{\mathbf{y}_i^m\}_{m=1}^{N_{tr}}, \quad \text{for } i = 1, \dots, n_s, \tag{4.17}$$

where $\mathbf{y}_i^m := \mathcal{F}_i^m \mathbf{P}_{PCA} \in \mathbb{R}^{d_y}$. We finally remark that all the data transformations, including data normalization, which are applied to the training set, must be also applied to the test datasets before class prediction.

Remark 5 *Interpretability of one-class algorithms.* Depending on the chosen one-class algorithm, different data compression strategies are needed to build a robust classifier. In particular, we observe that (*i*) for oc-SVMs, both the feature extraction and further dimensionality reduction are implemented, (*ii*) for the IFs, we perform only the feature extraction step, (*iii*) for the LOF algorithm, both the feature extraction and further dimensionality reduction are implemented, and (*iv*) for VAEs, raw signals are used directly without pre-selecting the features. The different datasets, used for each algorithm are summarized in Table 4.1. Intuitively, since damage-sensitive features are directly used to train the Isolation Forest algorithm, this kind of algorithms can be seen as an interpretive machine learning method in the sense that one can observe which features are mostly contributing to a decision. Autoencoders instead are trained to identify the best low-dimensional features automatically, thus making interpretability difficult. One-class SVMs and the LOF algorithm, if trained on very few features, could also be considered as interpretable learning methods. However, in many situations, it is not evident how to select the features that explain the salient properties of the training data and, to avoid overfitting, further dimensionality reduction is employed. In this sense, QR pivoting and other feature selection strategies could be an appropriate solution. However, feature interpretability goes beyond the objectives of this thesis and good results are obtained by employing data compression with PCA.

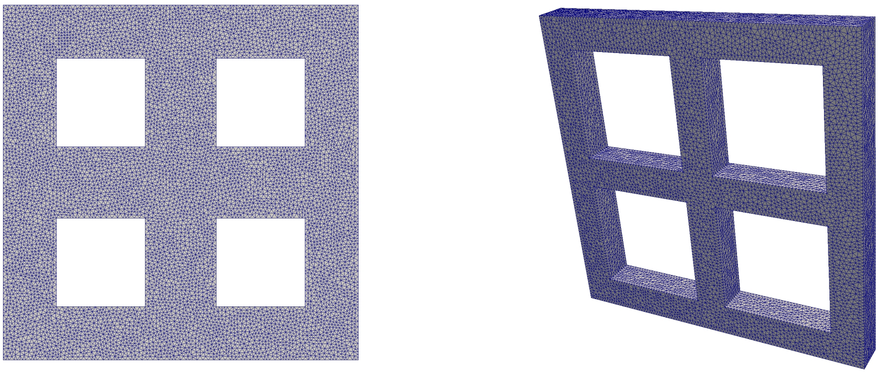


Figure 4.2 – Meshes for the healthy 2D (*left*) and 3D (*right*) problems. The former represents the section of a simplified beam and the latter is obtained by extruding a similar 2D domain, but with larger holes, along the third direction. The 2D problem has normalized dimensions 1×1, while the 3D one has dimensions $1 \times 1 \times 0.1$.

4.4 Numerical results

In this section we first present the 2D and 3D geometrical domains used in this thesis as a generic structure of interest equipped with a fixed number of sensors. Then, we define the parametric source of excitations and the other input parameters used to describe the natural variations of the structure. In Sections 4.4.1 and 4.4.2, we provide the details for the construction of the training and test sets by using the method proposed in Chapter 3, including the feature extraction and data compression procedures presented in Sections 4.3.1 and 4.3.2, respectively. In Section 4.4.2 we highlight the qualitative and quantitate differences between healthy and damaged sensor signals. The classification results obtained with the different strategies presented in this chapter, i.e., oc-SVM, IF, LOF, and VAE, are reported and compared in Section 4.4.3.

The meshes for the healthy domain $\Omega \subset \mathbb{R}^d$ are reproduced in Figure 4.2 for $d = 2, 3$. The domain is discretized using tetrahedral cells; a FE approximation by $\mathbb{P}_1$ elements is used for the high-fidelity solver, resulting in 30'912 and 217'344 DOFs for $d = 2, 3$, respectively. We remark that, since we solve (3.18), for the high-fidelity problem, half of the DOFs represent the real part and the other half the imaginary part of the $d-$dimensional solution. For all numerical experiments we use the homogeneous free-slip boundary conditions (3.3), i.e., $\boldsymbol{g}_N = \mathbf{0}$, and we fix the density and damping coefficients as $\rho := 1$ and $\eta := 0.1$, respectively. All the other parameters are defined below.

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 4.3, 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

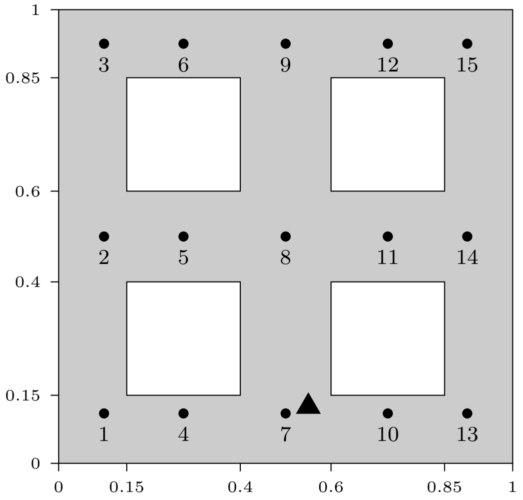


Figure 4.3 – Sketch of sensors numbering system and source placement for the 2D problem. The chosen 15 sensor locations (*numbered black circles*) and the location of the center of the active source, which corresponds to $\bar{S} = [0.54, 0.125]$, (*black triangle*) are shown.

the sensors location are restricted to the model surface, i.e., $\boldsymbol{x} := (x_i, y_j, z_k)$, represented in Figure 4.4, 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 homogeneous 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[5]. Since for the sake of simplicity we use as the output of interests the displacement time-signals, i.e., the output map $\ell : \mathbb{R}^d \to \mathbb{R}^{d_\ell}$ is the identity map and $d_\ell = d$, one of the d components of each of the Q features (4.16), extracted from each sensor signal, is identically zero. Hence, for the 3D problem, we consider features of dimension $(d-1) \times Q$, i.e., $N_f = 12$ features for both the 2D and 3D case.

Aiming at representing the different environmental and operational conditions, necessary to make reliable damage predictions, we choose three parameters of variation, i.e., $\boldsymbol{\mu} := [E, \nu, k] \in \mathcal{P} \subset \mathbb{R}^{d_\mu}$ with $d_\mu = 3$. The first two define the Lamé constants (3.2), i.e., E is the Young's Modulus and ν the Poisson's ratio, and k is a parameter of the source function $h(t; \boldsymbol{\mu})$, whose expression is given in (4.20). In the generation of the dataset, the parameter space (3.4) is chosen as

$$\mathcal{P} := [0.999, 1.001] \times [0.329, 0.331] \times [1.9, 2.1], \tag{4.18}$$

and the parameter set is based on uniform random samples from $\mathcal{P}$, i.e.,

$$\Xi^{n_\mu} := \{E_m, \nu_m, k_m\}_{m=1}^{n_\mu} \in \mathcal{P}, \tag{4.19}$$

---

[5] This phenomenon is an artefact of the choice of the boundary conditions and it does not affect the 2D problem simply because no sensor is chosen on the edges, where the boundary conditions $\boldsymbol{u} \cdot \boldsymbol{n} = \boldsymbol{0}$ are applied.

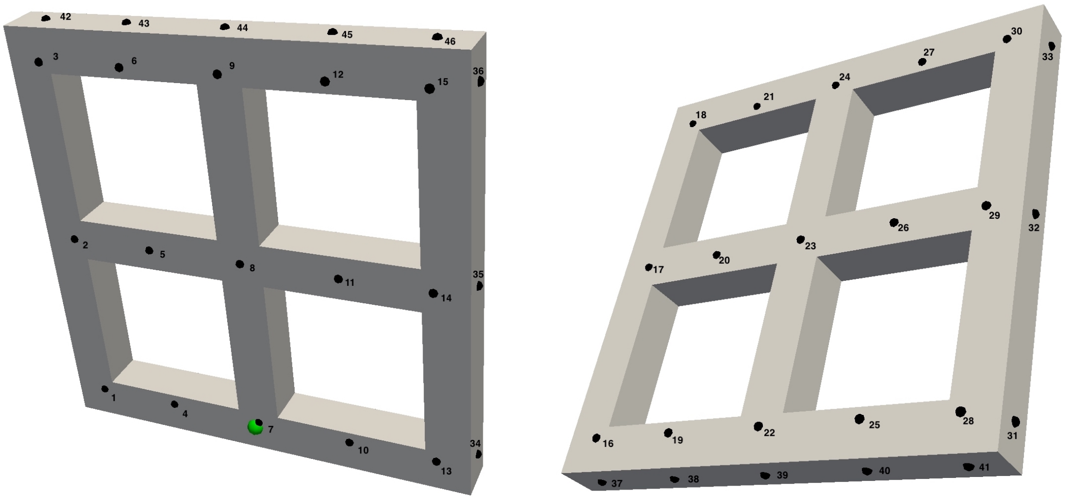


Figure 4.4 – Sketch of sensors numbering system and source placement for the 3D problem. The chosen 46 sensor locations (*numbered black semi-spheres*) and the location of the center of the active source, which corresponds to $\bar{S} = [0.51, 0.06, 0]$, (*green large semi-sphere*) are shown. The face with coordinate $z = 0$ is shown on the left, while the face with coordinate $z = 0.1$ is shown on the right.

where n_μ is either equal to N_{tr} or N_{test}. A more realistic parameter space could be provided by relying on model calibration, based on the combination of experimental data with prior knowledge. While this goes beyond the scope of this thesis, some background insights and preliminary work are provided in Appendix A.

The excitation of the structure is necessary to generate waveforms which propagate in the medium and are measured at the sensors for signal diagnostic. In this thesis, as previously discussed in Chapter 3, we consider active sources, as an alternative to passive continuous sources such as wind or tides. In several wave-based non-destructive evaluation tests, electromechanical shakers are used to inject pure white Gaussian noise, e.g., [LB14; OCDS17]. Alternatively, sources based on sinusoidal waves are also used, e.g., [TPYP18; ZBA13]). In this thesis, we focus on the second type of sources because of the impossibility of translating white noise in frequency domain. Moreover, in the SHM framework, short pulse impulses are often used for non-destructive evaluation and testing, e.g., the more sophisticated Hanning-windowed sinusoidal tone-bursts used in [ZBA13], in combination with the damage-sensitive features described in Section 4.3.1. 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 [DFPS96; LY01].

In this work, the source functions $s(\boldsymbol{x};\boldsymbol{\mu})$ and $h(t;\boldsymbol{\mu})$, introduced in (3.14), are chosen as

$$s(\boldsymbol{x};\boldsymbol{\mu}) := \frac{\exp\left\{-\sum_{i=1}^{d}\frac{(\boldsymbol{x}_i-\bar{S}_i)^2}{2\bar{\sigma}_i^2}\right\}}{2\pi\bar{\sigma}^d}, \quad h(t;\boldsymbol{\mu}) := k_s \sin(k\pi t)\, t^{k_1} e^{-t}, \tag{4.20}$$

where $\bar{\sigma} := 0.01$ represents the width of a Gaussian centered at $\bar{S} := [0.54, 0.125]$ and $\bar{S} := [0.51, 0.06, 0]$ 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, $k \in \mathbb{R}_+$, 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 homogeneous initial conditions, i.e., $\boldsymbol{u}_0 = \boldsymbol{v}_0 = 0$. The corresponding Laplace transform of $\tilde{h}$ is

$$\tilde{h}(z;k) = k_s \frac{2\pi k(z+1)}{(\pi^2k^2 + (z+1)^2)^2}. \tag{4.21}$$

Given $z := \alpha + \imath y$, (4.21) can be split in its real and imaginary parts, i.e., $h^{\alpha}(\alpha+\imath y;k)$ and $h^{y}(\alpha+\imath y;k)$, required in (3.19). Their explicit formulation can be computed using a symbolic software and is given as

$$h^{\alpha}(\alpha+\imath y;k) = C\left(2k\pi(1+\alpha)C_{\alpha}\right) \quad \text{and} \quad h^{y}(\alpha+\imath y;k) = C\left(2k\pi y C_y\right),$$

where $C \in \mathbb{R}_+, C_{\alpha} \in \mathbb{R}_+$, and $C_y \in \mathbb{R}_+$ are given as

$$\begin{aligned}
C &= \left(y^4 + \left(1 + (k\pi)^2 + 2\alpha + \alpha^2\right)^2 + 2y^2\left(1 - (k\pi)^2 + 2\alpha + \alpha^2\right)\right)^{-(k_1+1)},\\
C_{\alpha} &= -3y^4 + 2y^2(-1 + (k\pi)^2 - 2\alpha - \alpha^2) + \left(1 + (k\pi)^2 + 2\alpha + \alpha^2\right)^2,\\
C_y &= -3 - 2(k\pi)^2 + (k\pi)^4 + y^4 - 4(3 + (k\pi)^2)\alpha - 2(9 + (k\pi)^2)\alpha^2\\
&\quad - 12\alpha^3 - 3\alpha^4 - 2y^2\left(1 + (k\pi)^2 + 2\alpha + \alpha^2\right).
\end{aligned}$$

In this thesis we fix $k_1 = 1$.

Figure 4.5 shows the source function in time and frequency domain when the real part of the frequency $\tilde{z}$ is fixed, i.e., $\alpha = 0.26$, obtained using (3.33), and for different values of $k \in [1.9, 2.1]$.

4.4.1 A training set of healthy configurations

We present here the details to construct the training set for the 2D and 3D problems. For both problems, we first generate a dataset using the RB strategy presented in Section 3.4. For this, we set $k_z = 3$ which leads to a total of $N_s = 600$ and $N_s = 1500$ snapshots in 2D and 3D, respectively, given the choices $N_z = 200$ and $N_z = 500$, presented at the end of Section 3.3.2. Our sampling strategy is summarized in Remark 4, where the snapshot matrix (3.48) is obtained by combining the N_s solutions of the high-fidelity

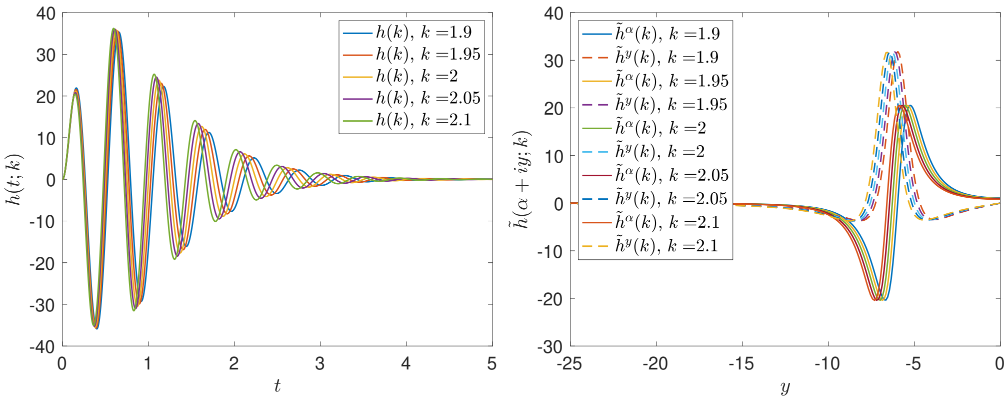


Figure 4.5 – Source function for various values of the parameter k. The source function $h(t;k)$ is plotted as a function of time (*left*) and its corresponding Laplace transform, split in its real (*full lines*) and imaginary (*dashed lines*) components, is plotted as a function of y, i.e., the imaginary part of frequency z for a fixed α value (*right*).

	N_h	k_z	N_z	N_s	ε_{POD}	N	Δ_t	N_t
2D	30’912	3	200	600	$1e-11$	159	$1e-3$	30’000
3D	217’344	3	500	1500	$1e-11$	255	$1e-3$	22’500

Table 4.2 – Summary of the parameters used to generate the synthetic 2D and 3D sensor measurements.

problems (3.18), obtained for different values of the frequencies $\tilde{z}$, defined in (3.27), and the parameter $\boldsymbol{\mu}_m$, obtained by uniform sampling as in (4.19) for $m = 1, \ldots, N_s$. By applying Algorithm 2 for a fixed tolerance $\varepsilon_{POD} = 1e-11$, we obtain $N = 159$ and $N = 251$ basis for the 2D and 3D case, respectively. A summary of the parameters used for the construction of the synthetic signals in 2D and 3D is given in Table 4.2.

The training datasets $\hat{\mathcal{D}}_i$ for $i = 1, \ldots, n_s$, defined in (4.13), are obtained by applying Algorithm 1 for $N_{tr} := 1000$ different input parameters. We remark that this requires solving the reduced problem (3.44) $N_z N_{tr}$ times. Depending on the choice of the one-class algorithm, described in Remark 5, we extract the damage-indicator features as explained in Section 4.3.1 and apply the PCA reduction to the normalized dataset, as detailed in Section 4.3.2. We refer to Table 4.1 for a summary of which transformation to apply to each algorithm.

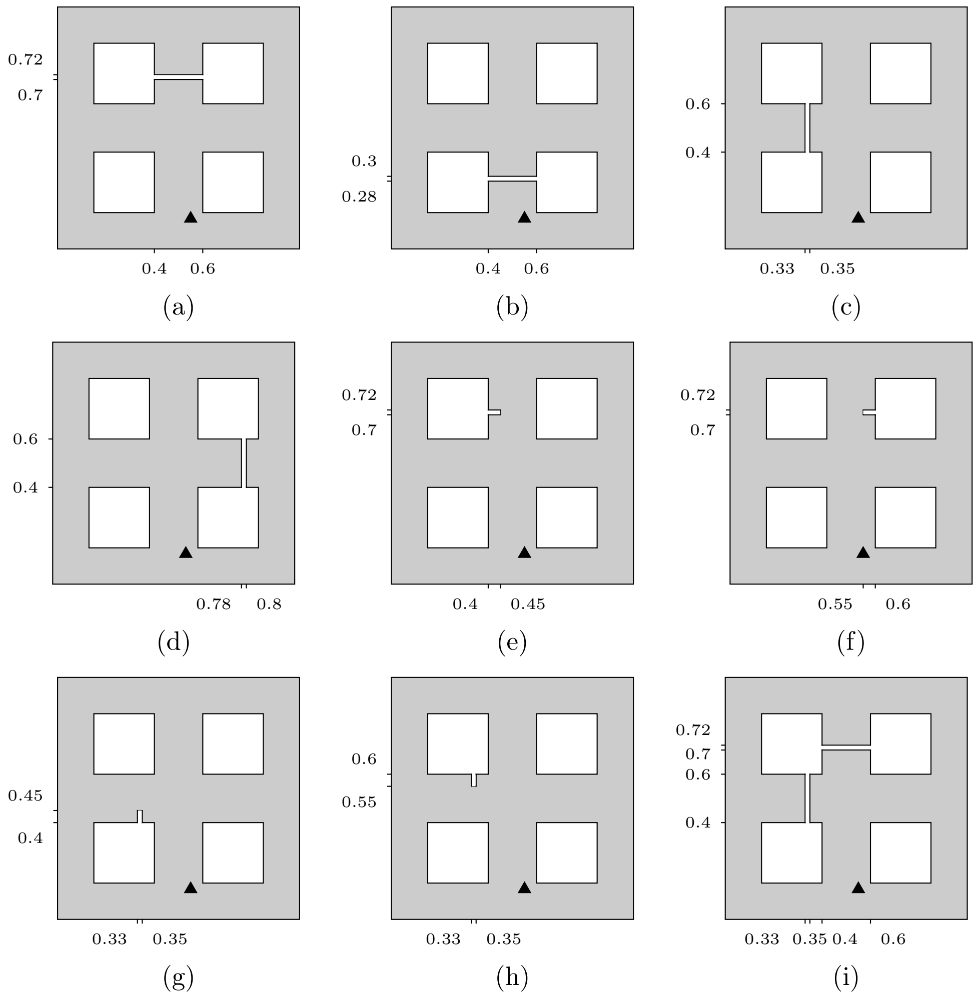
A clarification should be made for the VAE setup, for which we pre-process the discrete time-signals $\hat{\mathbf{g}}_i^m$ by shortening the time windows from $T = 30$ to $T = 5$ and by sub-sampling the signals at every 25 steps with $\Delta t = 0.025$. This choice allows to significantly reduce the dimensionality of each input signal, which is now a d_ℓ-dimensional vector with $\hat{N}_t = 200$ steps, without compromising the global information carried by the measurements. Indeed, as shown in Figures 4.7 and 4.8, the main differences between healthy and damaged configurations can be observed in the first 5 seconds. Dealing with low-dimensional inputs facilitates the task of the autoencoder, i.e., instead of using a deep network with the inconvenience of a long learning time, a more shallow network can provide good classification results. We finally remark that the same transformation is applied to the test samples described in the following section.

4.4.2 A test set of healthy and damaged configurations

The test set is composed of both healthy and damaged synthetic sensor measurements, but we expect this strategy to hold also for experimental measurements. The discrete time signals are obtained by solving the high fidelity problem (3.18) for N_μ new input parameters, sampled from the same parameter distribution used offline. As mentioned 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 $\varepsilon_i \in \mathcal{N}(0, \gamma_i^2)$, where γ_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 $N_{conf} := 10$ scenarios, 9 of which are damaged. These configurations are sketched in Figure 4.6, and among them 4 are considered as 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 $n_\mu = 10$ samples for a total of $N_{test} := N_{conf} N_\mu = 100$ test samples. In 3D, the test set is composed of $N_{conf} := 4$ scenarios: 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 $n_\mu := 10$ samples for each configuration are considered. The geometries are shown in Figure 4.12.

We observe that in this chapter the effect of cracks is obtained by modifying the geometry of the structure to include discontinuities. This approach is also used in [AVCT20], where artificial damages on the blade of a wind turbine are implemented via a trailing edge opening. Alternatively, especially in the vibration-based monitoring approach context, damages manifest themselves as changes to the mass, stiffness, and energy dissipation properties of the structure, see, e.g., [RMMC20], where stiffness degradation is used to simulate different damage scenarios. Indeed, reduction in stiffness is generally associated with a decrease in the natural frequencies and modification of the modes of vibration of the structure [SLZ00].

We compare the high-fidelity solutions obtained in Laplace domain for healthy and damaged structures in 2D, before and after applying Weeks method. In particular, the signals retrieved at the 9^{th} sensor, i.e., $\boldsymbol{x}_9 = (0.5, 9.25)$, are provided in Figure 4.7. 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} sensor is broken



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.6 – 2D geometries of nine damage configurations for the 2D problem. Figures $(a - d)$ correspond to major damages, while $(e - h)$ correspond to minor damages. Figure (i) is the union of two major damages, i.e., damages (a) and (c). The location of the center of the active source is the same for nine damaged geometries and the undamaged configuration, which corresponds to $\bar{S} = [0.54, 0.125]$ (*black triangle*).

Sensor number	$\mathbf{c}_1$	$\mathbf{a}_1$	$\mathbf{c}_2$	$\mathbf{a}_2$	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 4.3 – Comparison of crest factor **c** and arrival time **a** for high-fidelity reconstructed 2D signals at four different sensor locations for the healthy structure (see Figure 4.3) and a structure with a major damage (see geometry (a) in Figure 4.6). The subscript indicates the signal component. The retrieved signals are obtained using two input parameters, i.e. $\boldsymbol{\mu}^* = [1, 0.33, 2]$ and $\boldsymbol{\mu}^{**} = [0.9993, 0.3307, 2.07]$.

(see damage a in Figure 4.6) 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 4.8. 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 are reflected at the crack. Moreover, after computing the crest factor (4.14) and arrival time of these signals, we observe that these values are significantly different when looking at the damaged signals or the healthy ones, as summarized in Table 4.3. This observation supports our choice of using, among others, the crest factor and arrival time as damage-indicator features.

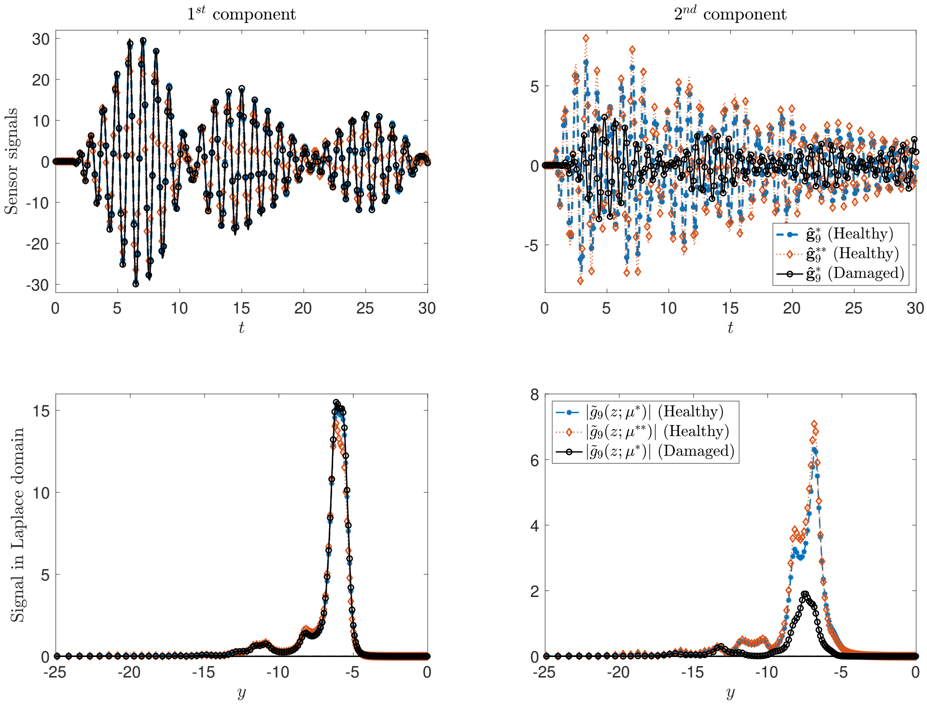


Figure 4.7 – Comparison of 2D signals both time and frequency domain, retrieved at the 9th sensor, obtained either from the healthy structure or from a structure with a major damage in the proximity of the 9th sensor (i.e., geometry (a) in Figure 4.6). From left to right, the first row shows the reconstructed signals obtained using Weeks method on the first and second component of the solution, 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, 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*).

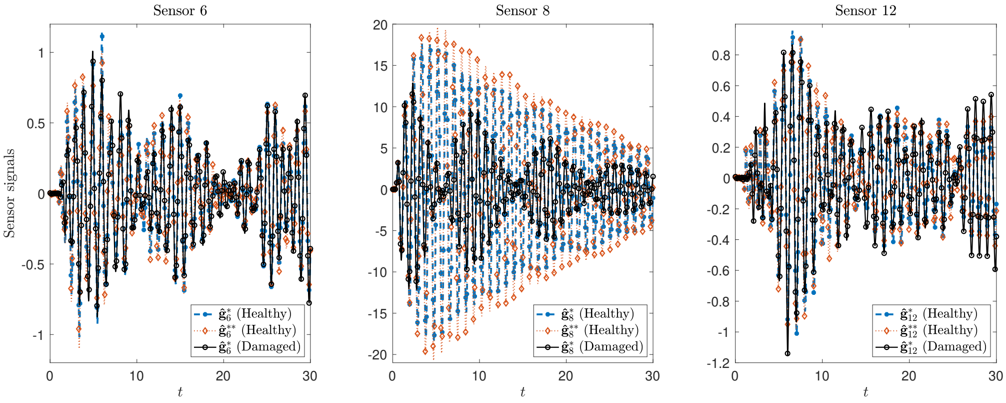


Figure 4.8 – Comparison of the second component of 2D reconstructed signals retrieved at the 6th, 8th and 12th, sensors, obtained from the healthy structure or from a structure with a major damage in the proximity of the 9th sensor. For the three plots, we show two healthy signals, obtained with two different parameters, 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*).

4.4.3 Classification results

In this section we present the one-class classification results on the test sets, sensor by sensor. We recall that, in 2D, the test set is composed of $N_{test} = 100$ samples, i.e., $n_\mu := 10$ samples for each one of the $N_{conf} := 10$ configurations (1 healthy and 9 damaged). In 3D, the test set is composed of $N_{test} = 40$ samples, i.e., $n_\mu := 10$ samples for each one of the $N_{conf} := 4$ configurations (1 healthy and 3 damaged). In both cases, each of the n_μ samples is obtained by solving the high fidelity problem with different input parameters $\boldsymbol{\mu}_m$ for $m = 1, \ldots, n_\mu$. A summary of the parameters used to generate the training and test set for each one-class algorithm is given in Table 4.4.

We first comment on the results obtained for the 2D and 3D problems with one of the four methods, the oc-SVMs. Then, we provide a comparison of the results obtained with

		n_s	N_{tr}	N_{test}	N_{conf}	n_μ	Q	N_f	d_y
2D	**oc-SVM**	15	1000	100	10	10	6	12	1
	IF	15	1000	100	10	10	6	12	-
	LOF	15	1000	100	10	10	6	12	1
	VAE	15	1000	100	10	10	-	-	-
3D	**oc-SVM**	46	1000	40	4	10	6	12	1

Table 4.4 – Summary of the parameters used to generate the synthetic train and test datasets in 2D and 3D, sensor by sensor, for four different one-class algorithms.

the other three algorithms for the 2D problem.

Classification results with oc-SVMs in 2D and 3D

To visualize the output of the classifier, we first present the oc-SVM relative anomaly scores (4.6), for the training and test sets, sensor by sensor. Figure 4.9 shows the histograms of the outputs of the classifiers obtained for the N_{tr} training samples. By construction, all the relative anomaly scores are below the threshold $\theta = 1$, represented by a dashed red line. A similar plot comparing the anomaly scores of the test samples is given in Figure 4.10. The two superposed histograms show the outputs of the classifiers obtained for the n_μ samples associated with either the healthy domain or the first damaged domain, i.e., the geometry shown in Figure 4.6 (a). As expected, the anomaly scores associated with the healthy geometry are almost all below the threshold for all the sensors and the few exceptions correspond to false alarm errors, e.g., one sample captured at sensor 12. Instead, the anomaly scores associated with the damaged geometry are partially above the threshold. In particular, the anomaly scores relative to the sensors closer to the crack, i.e., sensors 3, 6, 9, 12, and 15, are almost all above the threshold. This suggests that the classifiers allows to both detect and localize damages, as demonstrated by the results in the following tables and figures.

Tables 4.5 and 4.6 show, for each type of damage, the fractions of test samples classified as outliers, i.e., with an anomaly score greater than $\theta := 1$, while the mean values for each damaged configurations are shown in Figures 4.11 and 4.12, for the 2D and 3D problems, respectively. Sensors whose average anomaly score (4.6) is greater than θ are represented with red markers (outliers), while blue markers identify the sensors with average anomaly score smaller than θ (inliers). To capture the severity of the damage we introduce an arbitrary value $\theta_2 := 2$, i.e., sensors with mean anomaly score greater or equal than θ_2 are defined as strong outliers, while mild outliers have mean anomaly score greater than or equal to θ, but smaller θ_2. Strong outliers are represented with red squares, while mild outliers with red asterisks in 2D and red semi-spheres in 3D. Alternatively, for visualization purposes, a continuous colour scale could be used to represent the mean classification results.

Both in 2D and 3D, we observe that damages are generally 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 are classified as strong outliers. Even if not reported in Figure 4.11, all sensors of the 2D healthy configuration are, on average, classified as inliers, while the average result for the 3D healthy configuration (see geometry (a) in Figure 4.12) presents 1 misclassified sensor. 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.

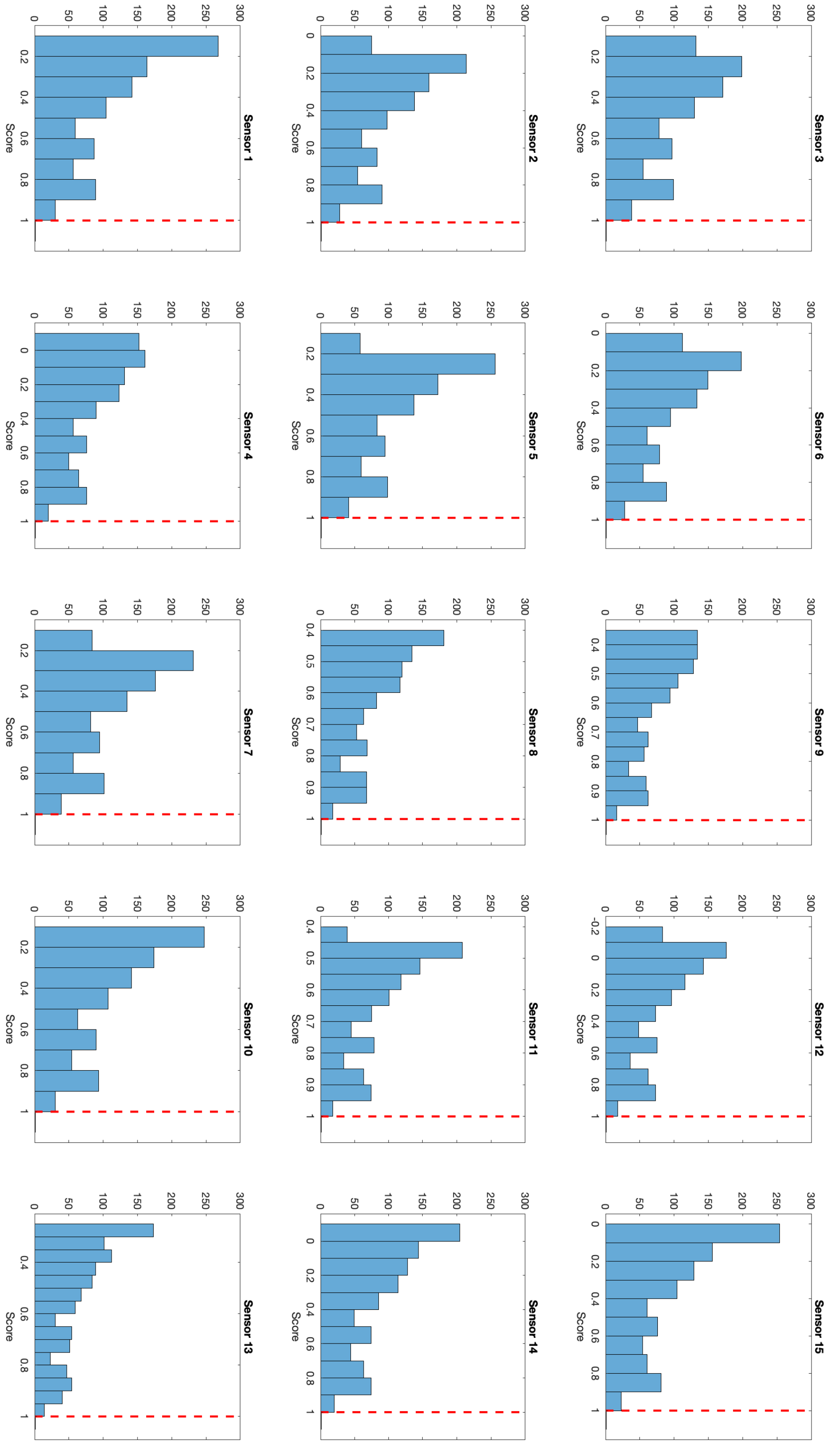


Figure 4.9 – Relative anomaly scores of the $N_{tr} = 1000$ training samples obtained using the oc-SVM classifiers, sensor by sensor. The threshold $\theta = 1$ is shown by means of a red dashed line.

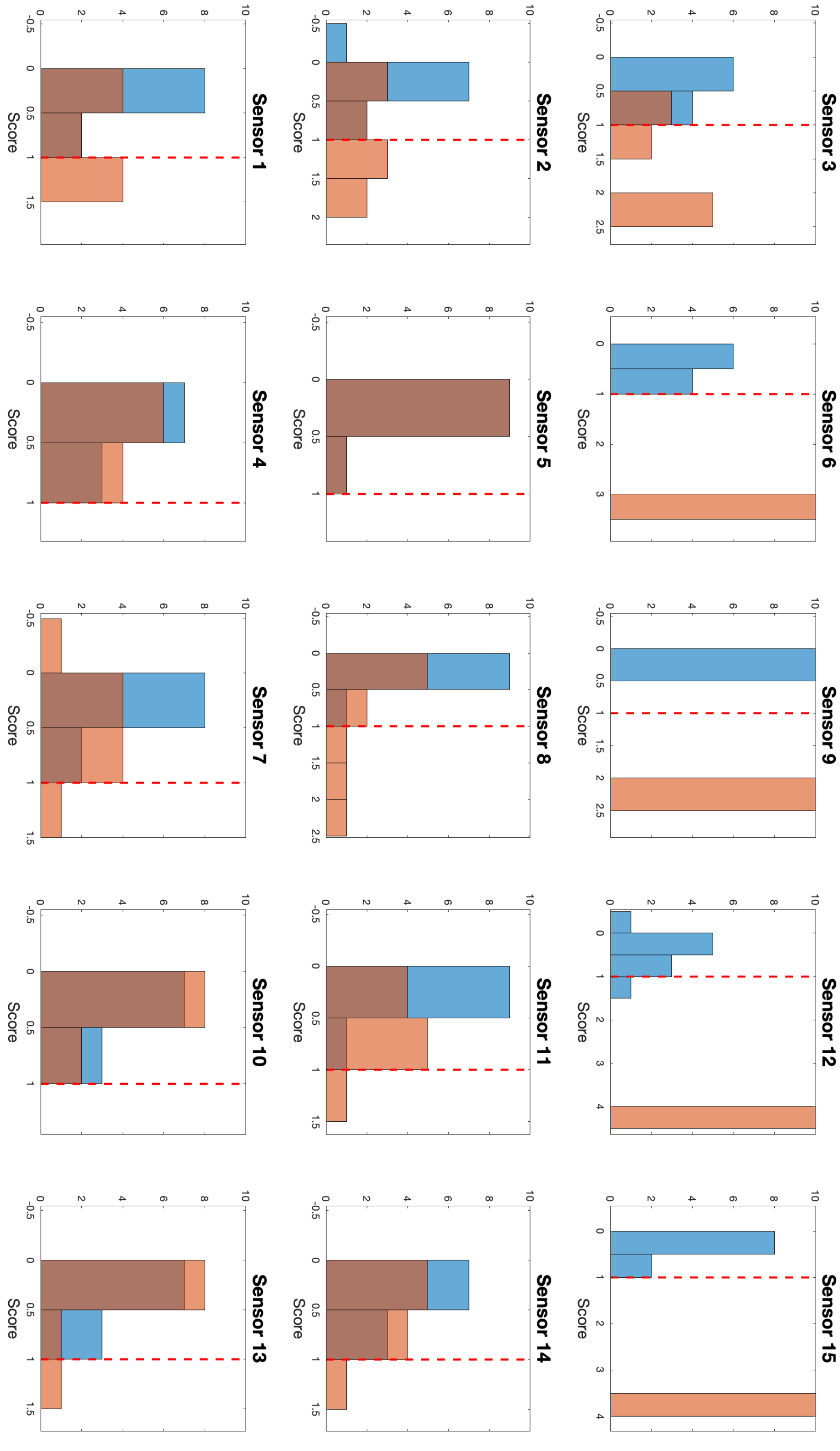


Figure 4.10 – Relative anomaly scores of the $n_\mu = 10$ test samples associated with the healthy geometry (*blue*) and the $n_\mu = 10$ test samples associated with the damaged geometry shown in Figure 4.6 (a) (*orange*). Both results are obtained using the oc-SVM classifiers, sensor by sensor. The threshold $\theta = 1$ is shown by means of a red dashed line.

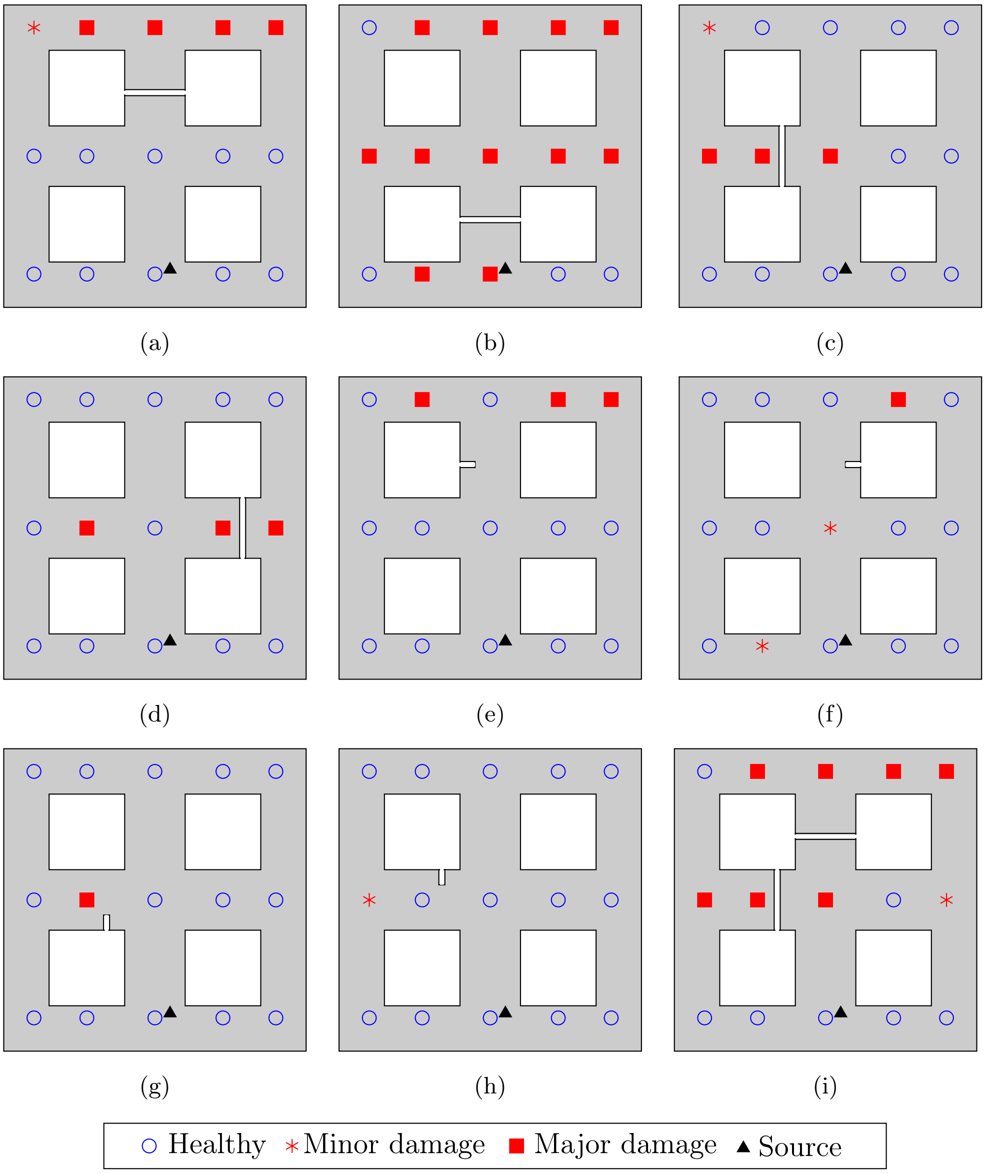
Sensor	Healthy	Major damage (a)	Major damage (b)	Major damage (c)	Major damage (d)	Minor damage (e)	Minor damage (f)	Minor damage (g)	Minor damage (h)	Combined damage (i)
1	0	0.4	0.4	0	0.1	0.1	0	0	0.1	0.1
2	0	0.5	1	1	0	0.2	0	0	0.5	1
3	0	0.7	0.1	0.7	0	0	0	0	0.1	0.3
4	0	0	0.9	0.4	0	0.1	0.5	0.1	0.1	0.1
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 4.5 – Fractions of test samples classified as outliers using oc-SVMs for the 2D problem with one healthy configuration (see Fig. 4.3) and nine damaged configurations (see Fig. 4.6). A set of uniformly sampled input parameters $\{\mu^\star_m\}_{m=1}^{n_\mu} \in \mathcal{P}$ is used to average the results over $n_\mu = 10$ test samples per configuration.

The relative position of the source, the sensors, and the damage is important to successfully locate the damage based on guided-waves. Indeed, in 2D, for the major damages (a), (c), (d), and (i) only the sensors *behind* the damage are classified as outliers, allowing for localization. Instead, with the major damage (b) positioned too close to the source, 11 out of 15 sensors are, on average, classified as outliers, thus preventing localization. The position of the source plays an important role in classification and therefore, to localize damage (b), the source should be placed differently. 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 [SFB$^+$06], 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 Chapter 7 of [FW12].

Comparison of the four one-class classification methods in 2D

Tables 4.7, 4.8, and 4.9 show, for each type of damage, the fractions of test samples classified as outliers, i.e., with an anomaly score (4.12) greater than $\theta := 1$, using the IF, LOF algorithm, and VAE, respectively. The mean value for each damaged configuration for the corresponding algorithms are shown in Figures 4.13, 4.14, and 4.15. To qualitatively distinguish between mild outliers and strong outliers, we arbitrarily choose the threshold $\theta_2 := 1.1$ for the IF method and $\theta_2 := 5$ for the LOF and VAE methods. We note that, in the multi-sensor semi-supervised setting of the examples presented here, it is not entirely clear how to quantify the accuracy of the classification by means of a unique index,



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.11 – Sketch to summarize the one-class SVMs classification average results on test data for nine 2D damage configurations. For each configuration, major damages correspond to sensors classified as strong outliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) \geq 2$ (*red filled squares*), minor damages to sensors classified as mild outliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) \in [1, 2[$ (*red asterisks*), and undamaged configurations to sensors classified as inliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) < 1$ (*blue empty circles*). For all types of damages we can identify at least one sensor classified as an 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. 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. The location of the center of the active source is the same for the nine damaged geometries and the undamaged configuration, which corresponds to $\bar{S} = [0.54, 0.125]$ (*black triangle*).

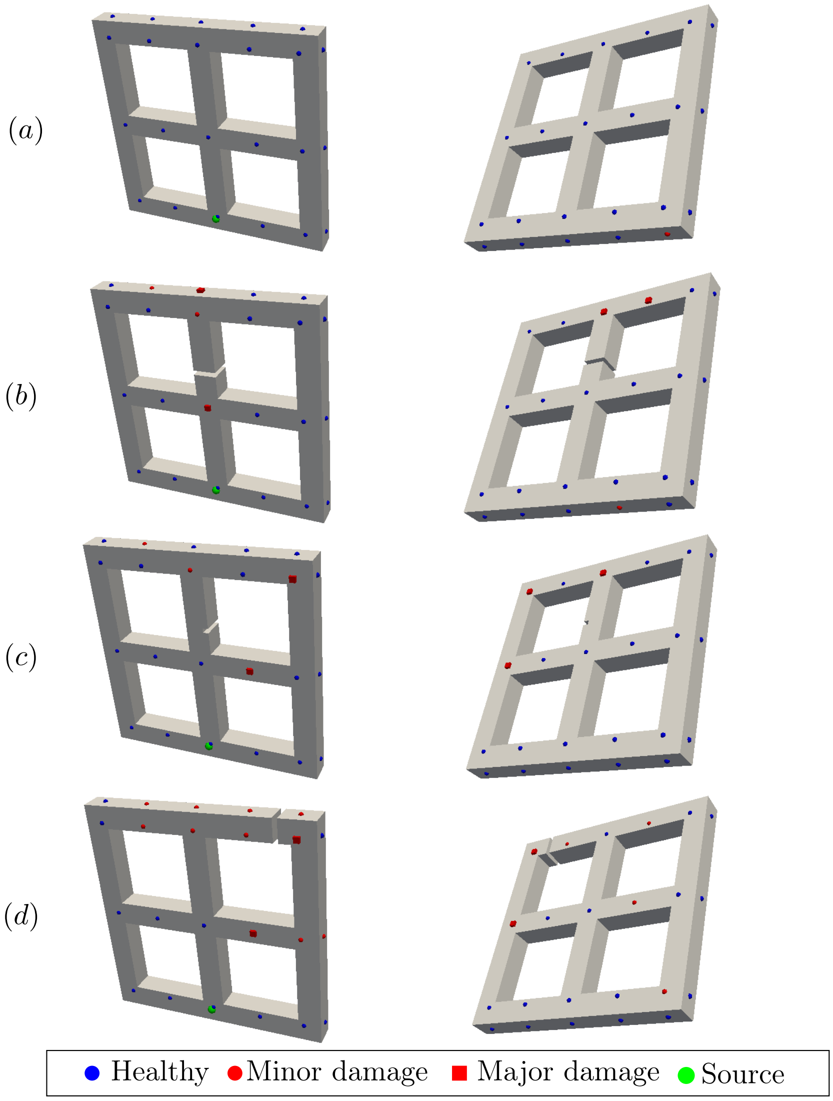


Figure 4.12 – Sketch to summarize the geometries of the four configurations used in the test set for the 3D damage problem, together with the one-class SVMs classification average results. For each configuration, major damages correspond to sensors classified as strong outliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) \geq 2$ (*red filled squares*), minor damages to sensors classified as mild outliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) \in [1, 2[$ (*red filled semi-spheres*), and undamaged configurations to sensors classified as inliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) < 1$ (*blue filled semi-spheres*). 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). The location of the center of the active source is the same for the three damaged geometries and the undamaged configuration, which corresponds to $\bar{S} = [0.51, 0.06, 0]$ (*large green semi-sphere*).

Sensor	Healthy (a)	Major damage (b)	Minor damage (c)	Major 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 4.6 – Fractions of test samples classified as outliers using oc-SVMs for the 3D problem with one healthy configuration and three damaged configurations (see Fig. 4.12). A set of uniformly sampled input parameters $\{\mu_m^\star\}_{m=1}^{10} \in \mathcal{P}$ is used to average the results over $n_\mu = 10$ test samples per configuration.

which makes the comparison of the four algorithms less trivial. Indeed, if we were to use standard indices in machine learning, e.g., the area under the curve (AUC), we should decide a priori which sensors we expect to be classified as outliers for each damaged configuration. This choice has an impact on the expected trade-off between false positives and false negatives and should be driven by the specific maintenance objectives. Without going into such detail, in the remainder we provide a high-level comparison of the four methods.

First, we note that all methods perform well in detecting damaged configurations, i.e., all the four methods classify as inliers almost all the sensors belonging to the healthy configuration and at least one sensor is classified as outlier for all the damaged configurations. Second, the location and extent of damages are qualitatively better captured by oc-SVMs and IFs, while for the LOF and VAEs, almost all sensors of each damaged configurations are classified as outliers, preventing localization. Nevertheless, it is interesting to observe that for VAEs the distinction between mild and strong outliers provides a qualitative way to identify the position of damages, i.e., the damage is closer to sensors classified as strong outliers. Third, from a computational cost perspective, the first three methods are very fast and all the n_s models are trained in a few minutes. The variational autoencoders require instead a few hours for training. Last, the fine tuning of hyper-parameters is a very important aspect of all the methods to avoid overfitting. In addition, while we are required to fix the values of only a small number of hyper-parameters for the first three models, the complex architecture of VAE presents a much larger set of choices.

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

Table 4.7 – Fractions of test samples classified as outliers using IFs for the 2D problem with one healthy configuration and nine damaged configurations. All the results are averaged over $n_\mu = 10$ test samples per configuration.

Sensor	Healthy	Major damage (*a*)	Major damage (*b*)	Major damage (*c*)	Major 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	0.8	1	0.1	1	0.1	0.3	0.2	0	0
4	0.1	1	1	0.1	0	0.6	1	0	0	1
5	0	0.2	0.1	0.1	0	0.1	0.2	0	0.1	0.1
6	0	0.3	1	0.1	1	0.2	0	0.1	0.3	0.8
7	0	1	1	0.1	0.1	1	0	0	0	1
8	0	0.7	0.4	0.7	0.3	0.2	0.2	0	0.1	0.5
9	0	0	0.9	0.4	0	0.1	0.5	0.1	0.1	0.1
10	0.1	0.1	1	1	1	0.2	0.5	1	0	1
11	0	1	1	0	0	1	0	0	0	1
12	0	0	1	0	0	0	0	0	0	0
13	0	0.2	1	1	0.2	0	0.2	0.1	0.5	1
14	0	1	1	0	0	0.1	0.2	0	0	1
15	0	0	0	0.3	0.1	0.1	0	0.2	0.1	0.1

Table 4.8 – Fractions of test samples classified as outliers using the LOF algorithm for the 2D problem with one healthy configuration and nine damaged configurations. All the results are averaged over $n_\mu = 10$ test samples per configuration.

Sensor	Healthy	Major damage (*a*)	Major damage (*b*)	Major damage (*c*)	Major damage (*d*)	Minor damage (*e*)	Minor damage (*f*)	Minor damage (*g*)	Minor damage (*h*)	Combined damage (*i*)
0.5	0	1	1	1	1	1	1	1	1	0.5
1	0	1	1	1	1	1	1	1	1	1
1	0	1	1	0.9	1	1	1	1	0.4	1
1	0	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1
0.9	0	1	1	0.7	0.9	1	1	1	0.9	0.9
0	0	1	1	1	1	1	1	1	1	0
0	0	1	1	0	1	1	1	1	0.1	0
0.5	0	1	1	0.5	0	1	1	1	0	0.5
1	0	1	1	1	1	1	0.2	1	1	1
0.7	0	1	1	1	0.9	1	1	1	1	0.7
0.1	0	1	1	0.4	1	1	1	1	0	0.1
0.8	0	1	1	1	1	1	1	1	1	0.8
0	0	1	1	0	0.5	1	1	1	0.1	0

Table 4.9 – Fractions of test samples classified as outliers using VAE for the 2D problem with one healthy configuration and nine damaged configurations. All the results are averaged over $n_\mu = 10$ test samples per configuration.

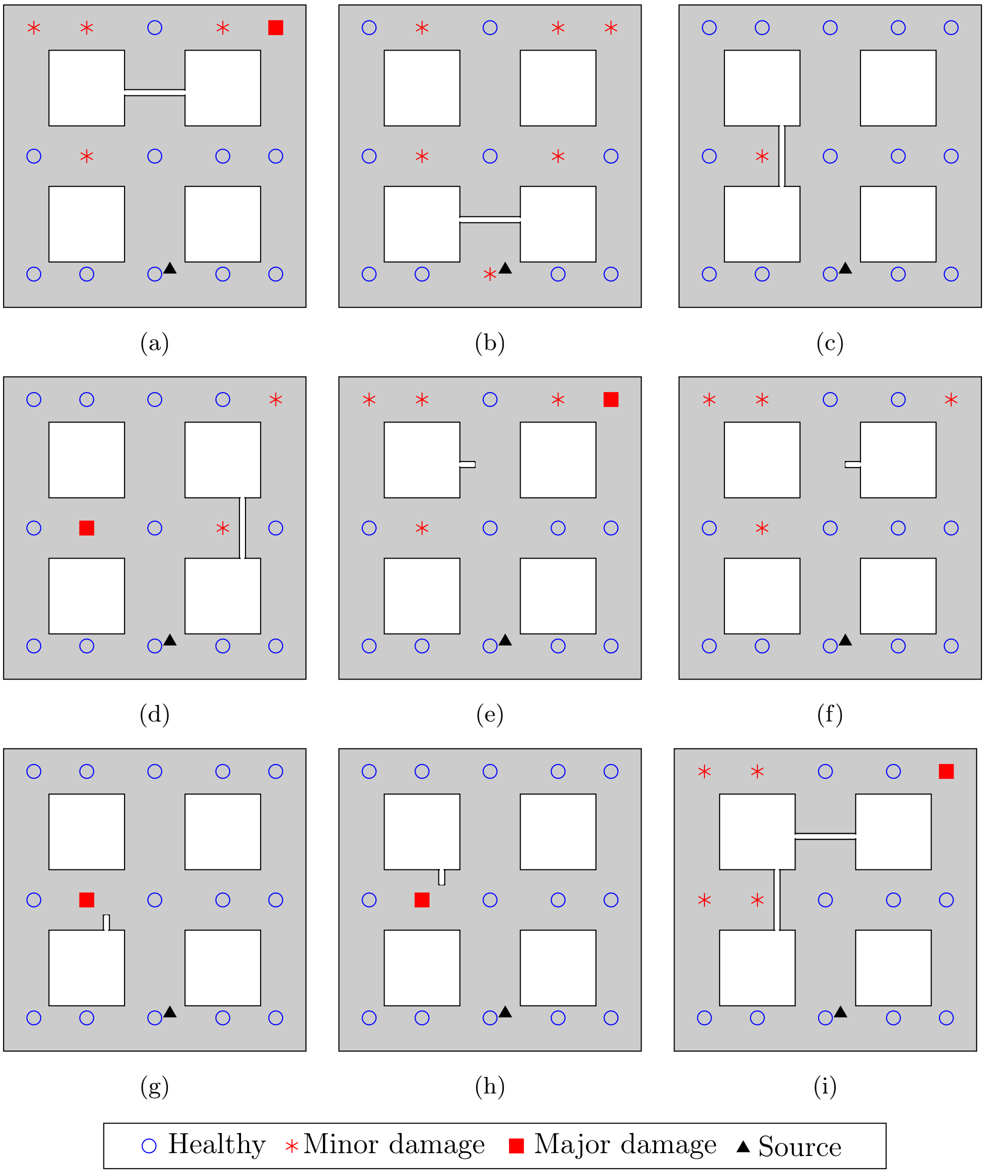


Figure 4.13 – Sketch to summarize the IF classification average results on test data for nine 2D damaged configurations. For each configuration, major damages correspond to sensors classified as strong outliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) \geq 1.1$ (*red filled squares*), minor damages to sensors classified as mild outliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) \in [1, 1.1[$ (*red asterisks*), and undamaged configurations to sensors classified as inliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) < 1$ (*blue empty circles*). For all types of damages we can identify at least one sensor classified as an outlier. For most of the damaged configurations, we observe a connection between the location of the damages and the sensors classified as outliers. The location of the center of the active source is the same for the nine damaged geometries and the undamaged configuration, which corresponds to $\bar{S} = [0.54, 0.125]$ (*black triangle*).

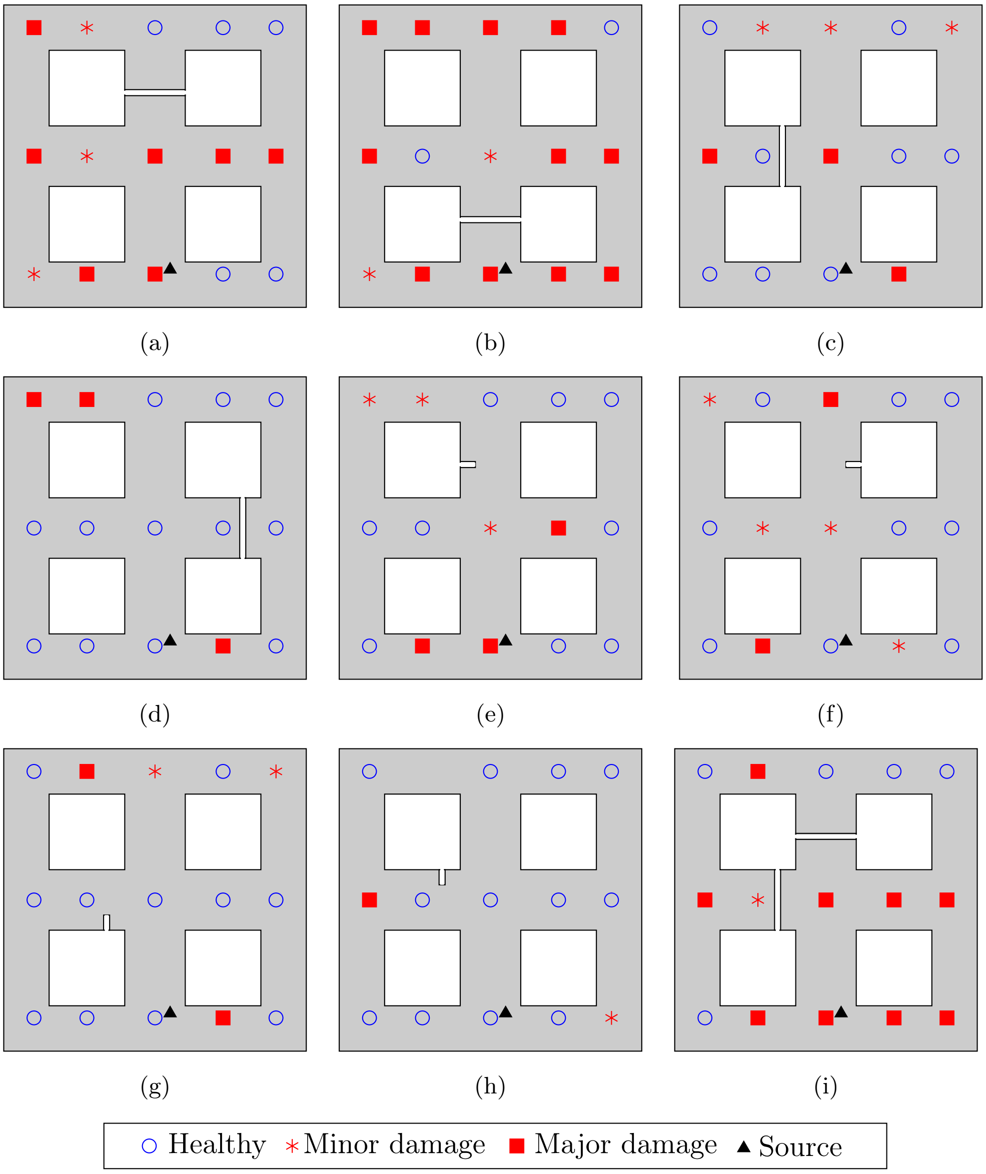


Figure 4.14 – Sketch to summarize the LOF classification average results on test data for nine 2D damaged configurations. For each configuration, major damages correspond to sensors classified as strong outliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) \geq 5$ (*red filled squares*), minor damages to sensors classified as mild outliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) \in [1, 5[$ (*red asterisks*), and undamaged configurations to sensors classified as inliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) < 1$ (*blue empty circles*). For all types of damages we can identify at least one sensor classified as an outlier. For most of the damaged configurations, the results are not conclusive to locate the position and estimate the extent of the different damages. The location of the center of the active source is the same for the nine damaged geometries and the undamaged configuration, which corresponds to $\bar{S} = [0.54, 0.125]$ (*black triangle*).

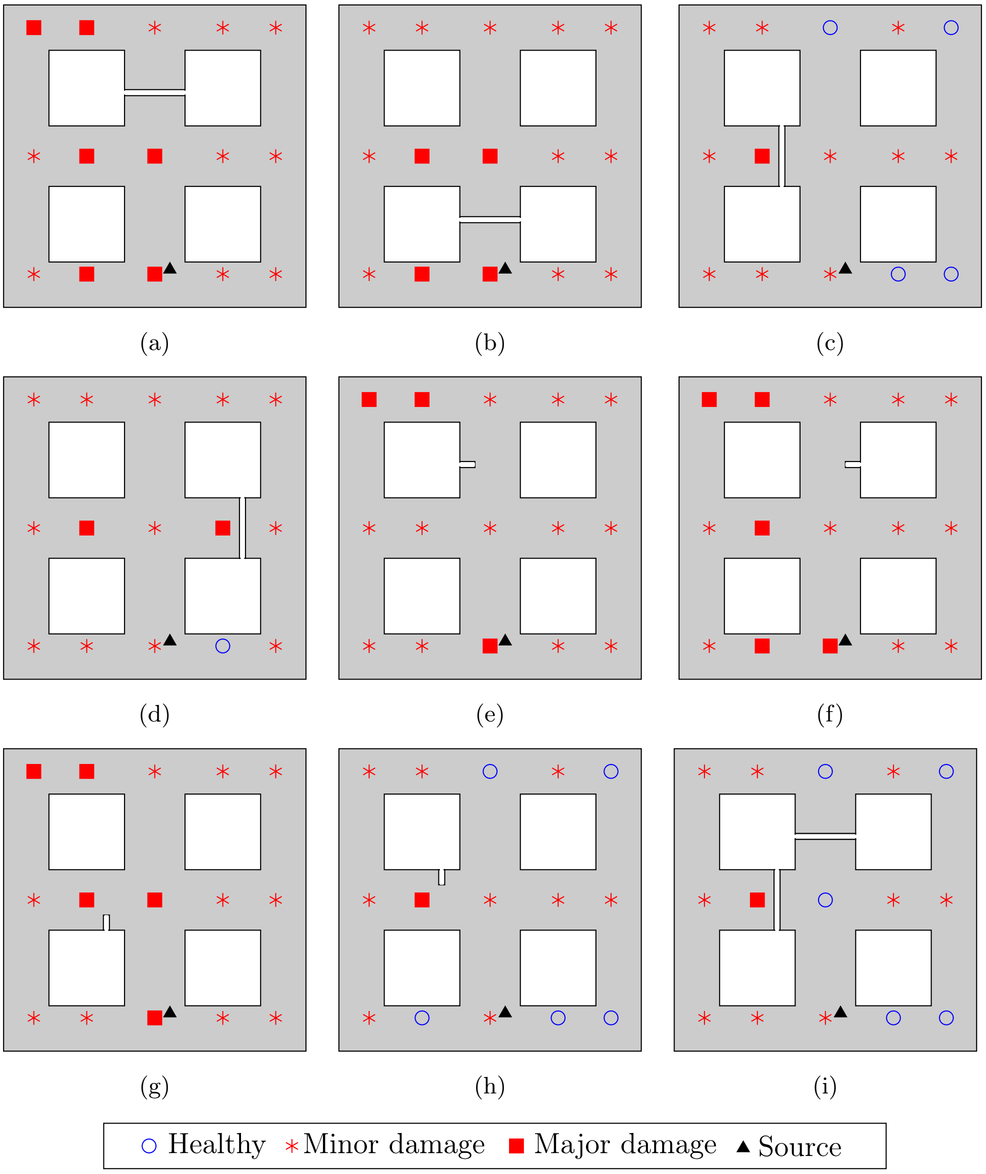

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 4.15 – Sketch to summarize the VAE classification average results on test data for nine 2D damaged configurations. For each configuration, major damages correspond to sensors classified as strong outliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) \geq 5$ (*red filled squares*), minor damages to sensors classified as mild outliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) \in [1, 5[$ (*red asterisks*), and undamaged configurations to sensors classified as inliers, i.e., with an average anomaly score $s_i(\boldsymbol{x}_i) < 1$ (*blue empty circles*). For all types of damages we can identify at least one sensor classified as an outlier. For most of the damaged configurations, the results are not conclusive to locate the position and estimate the extent of the different damages. The location of the center of the active source is the same for the nine damaged geometries and the undamaged configuration, which corresponds to $\bar{S} = [0.54, 0.125]$ (*black triangle*).

4.5 Conclusions

In this chapter, 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, 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 that emulate the sensor response of the structure to active sources, i.e., we analyze the structural response to the propagation of guided-waves. As described in Chapter 3, these training databases are constructed offline by repeatedly solving PDEs in the frequency domain for different input parameters and by exploiting ROM techniques for speedup. The reconstruction of time signals is compared using Weeks method, as 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 four different one-class classifiers to compare the healthy training dataset with new unseen 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. In particular, among the methods we tested, oc-SVMs and IFs seem to provide the best results in terms of localization. We conjecture that a finer tuning of the hyper-parameters could improve the results obtained with LOF and VAEs. We observe that, using active sources, localization is possible only for damages which are sufficiently far from the source. To address this limitation, the possibility of introducing a network of sources placed at different locations should be further investigated. 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. The method is highly generalizable to other examples and more realistic experiments should be carried out within a laboratory environment to further validate our approach. In particular, the availability of experimental measurements may be used to better tune the threshold value of θ. Moreover, the offline-online decoupling of tasks and the ROM techniques allow us to compute the sensor response under different operational and environmental conditions in a fast and inexpensive manner. By exploiting this advantage, in the next chapter we present a framework for the optimal placement of sensors, needed to both retrieve maximum information about the potential structure damages and guarantee a robust network of sensors, which seeks 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, autoencoders could still provide an automatic tool to identify the underlying characteristics of healthy signals. Otherwise, the features used as damage-indicators would need to be adapted.

CHAPTER 5

SENSOR PLACEMENT

The content presented in this chapter is mainly based on [BZH20].

In this chapter we present a sensor placement strategy for structural health monitoring applications. In particular, Section 5.1 presents a literature overview of sensor placement strategies in the structural health monitoring context. It also describes the connections of our main contribution to other applications of sensor placements. Sparse Gaussian process (GP) approximations are introduced in Section 5.2 with a particular emphasis on variational sparse GP regression, which is used in this chapter. We explain how variational approximations are used for sensor placement in the absence of damage states in Section 5.3 and provide numerical evidence of the quality of this method in Section 5.4. Conclusions are given in Section 5.5.

5.1 A survey of sensor placement for SHM

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, as discussed in Chapter 1. Despite the existence of different approaches and various types of sensors, all SHM techniques depend on the information acquired by a network of sensors. Hence, performance depends critically on the quality of the information collected by these sensors. Clearly, both improving sensor sensitivity and deciding where to place sensors play a key role in the digital twin industry.

Motivated by the opportunities of cost reduction for SHM systems and the improvement in the quality of the monitoring outcome, optimization of sensor placement (OSP) has received increasing interest during the last decades. The exhaustive review [OSM19] provides a collection of examples of OSP applied to SHM, classified based on the different techniques employed for the sensor placement optimization itself, among which the vibration-based and the wave-based monitoring are the most commonly used. While the former depend on the dynamics of the structure using passive sources, e.g., only the ambient loads on the structures are considered, the latter are usually used in the active sensing domain. Where as vibration monitoring techniques seek to identify changes in the natural frequencies and mode shapes with respect to a baseline, in the wave-based monitoring field, vibrations are generated by a controlled source, e.g., a sinusoidal wave or a short impulse, and signal-processing techniques are used to differentiate baseline time-dependent responses from the reflections and refraction of the wave caused by the presence of damages. Since the non-destructive impulses used to excite a structure have a high damping effect, i.e., it is difficult to observe the effect of the guided-wave far from the source, wave-based monitoring techniques are usually employed to monitor pipes or plate-like components with complex geometries, e.g., in aeronautical applications [MM07; TKA16]. On the contrary, large-scale assets, e.g., dams, bridges, etc., are usually monitored by vibration-based techniques, see e.g., [CCMA17], or by static approaches, see e.g., [HKP13].

Despite their fundamental differences, the general deployment of an OSP strategy is similar for both approaches. The OSP process can be split into a sequence of a few stages going from the choice of sensor types, over the definition of operational parameters, e.g., the candidate sensor locations, and, finally, to the characterization of a suitable cost function and optimization algorithm, e.g., gradient-based techniques are chosen when the cost function is continuous and differentiable, while meta-heuristic optimizations might be necessary otherwise. We discuss here the state of the art of OSP for both the vibration- and the wave-based monitoring techniques. Among the most popular placement strategies for the former, we note the effective independence method (EFI), the kinetic energy method (KE), and the more recent information theory approach, which seeks an optimal placement of sensors by minimizing the information gain within a Bayesian experimental design framework, see e.g., [Pap04; CCMA17; ACZP18]. For active sensing based on guided waves, we focus on [FT10] and [LS07]. In the former, the authors propose an optimization procedure where the sensor locations are chosen to minimize the risk of false alarms and mis-detections. The latter proposes a strategy to increase the sensitivity to damage by using simulation-based techniques, in which, by comparing the numerical solution of the guided-wave propagation in undamaged versus damaged scenarios, sensors are placed where the largest increase in the signal amplitude is observed. When the wave propagation patterns are very complicated, it has been proposed to maximize the area of coverage (MAC) within a sensor network, see e.g., [TKA16], where physical properties of Lamb wave propagation and complex geometrical properties are taken into account, or [SMKO18], where the ellipse equations with the sensor actuator pair as the foci are used to compute the coverage area.

We note that, with the exception of the strategies which maximize the coverage area, all these OSP techniques require knowledge about the characteristics of the damage, e.g., its type, its location, its severity, or its size. Consequently, these approaches do not generalize well when other types of damages occur and, even though engineering knowledge can certainly direct the attention to damages that are more likely to occur, it seems unreasonable to characterize them all. In particular, when relying on numerical simulations to describe the effect of a particular damage on a structure, including many damage types and all possible combinations becomes computationally intractable. An alternative is to resort to anomaly detection techniques, where damages are identified only by looking at the output of multiple undamaged scenarios, collected under different standard conditions, which may represent environmental or operational healthy variations. We refer to Chapter 4 for a description of how to address the damage detection problem with anomaly detection learning strategies for a fixed network of sensors. However, many questions arise if one wishes to find the optimal sensor locations in the absence of any damage information. In particular, the definition of new operational parameters and their corresponding cost function must be considered.

In this chapter we propose a novel strategy for sensor placement in the context of anomaly detection applied to SHM when a fixed budget is given, i.e., the number and type of sensors is fixed. The sensor locations are systematically identified as the spacial positions for which the reconstruction error of an output of interest at all *unsensed* locations is minimized. The quantity of interest that defines the cost function for the sensor placement optimization algorithm is the same quantity used to train the anomaly detection classifier which distinguishes healthy configurations from damaged ones. As such, the proposed placement strategy is based on an appropriate indicator of the damage detection performance of a given network. More precisely, we employ the variational inference of sparse Gaussian process regression (GPR) for a damage-sensitive quantity of interest representing a healthy scenario, and we use the inducing inputs as the sensor locations. With the variational formulation, sensor locations are selected by minimizing the Kullback-Leibler (KL) divergence between the exact posterior distribution and the variational distribution. Therefore, placing sensors at the corresponding location of the inducing inputs addresses both the information compression of the whole domain and the total variance reduction at the sensor locations. We also rely on an Expectation-Maximization (EM)-like algorithm for the training phase, which, on one hand, prevents a combinatorial search in the case of a discrete admissible set of points and, on the other hand, allows the inclusion of domain restrictions in the optimization to avoid placing sensors in areas difficult to reach or unsuitable for monitoring. Furthermore, we extend the proposed algorithm to take into account the natural variations of the model parameters, e.g., loads, boundary conditions, material properties, etc., by means of an unsupervised clustering algorithm. To conclude, we present some numerical examples to test the validity of the proposed method. In particular, we resort to a wave-propagation based strategy to place sensors on both 2D and 3D structures and to a static monitoring approach with passive sources to place sensors on a 3D representation of an offshore jacket.

We finally observe that the problem of sensor placement has been addressed in the literature also from perspectives different from SHM. We mention here three strategies, each relying on a different method and designed for a different application. Among them, only the first one proposes to use of GPs, even if in a different way and context with respect to the method proposed in this chapter.

First, the use of GPs for sensor placement has been proposed in [Cre91; KSG08], where either the maximum entropy principle or a mutual information criterion are applied to identify near-optimal locations. In contrast, our work replaces the classic GPR model with a sparse variational approximation, which at the same time identifies the optimal sensors as the inducing points automatically and adresses challenges with large data set. Additionally, the strategy presented in [KSG08] is used to monitor diffusion-like spatial phenomena, e.g., temperature in an indoor environment, while the SHM applications involve more complex phenomena, for which the training of a GPR may not be straightforward.

Second, in the recent work [ABDC$^+$18], the authors investigate a greedy method to place sensors in a systematic manner to assist field experts in placing sensors in nuclear reactors. In particular, they propose to use the magic points of the generalized empirical interpolation method (GEIM) as sensor locations and show the effectiveness of this strategy on multidimensional examples based on synthetic measurements. Different from the approach proposed in this thesis, these interpolation points often tend to cluster on the border of the domain, thus leading to waste in sensed information.

Third, sparse approaches for sensor placement have been proposed in [BBPK16], where the authors exploit the low-dimensional structure exhibited by many high-dimensional systems to compress a signal to very few measurements if the sole objective is classification. Despite the use of sparsity-promoting techniques, this work is entirely based on classification, which is different from the scope of the work discussed this chapter.

5.2 Gaussian process regression

The sparse GP regression has received increasing attention in the last decades thanks to its ability to overcome the computational limitation of a standard GP. Indeed, given the number of training samples n, the computational complexity of generating a GP model is $\mathcal{O}(n^3)$ and the associated storage requirement $\mathcal{O}(n^2)$, which becomes intractable for large data sets. The corresponding sparse methods instead rely on a small set of $m \ll n$ points to facilitate the information gain of the whole data set, thus allowing for a complexity reduction, i.e., $\mathcal{O}(nm^2)$. After a short introduction of GP regression in Section 5.2.1, we detail the properties and advantages of its sparse variation in Section 5.2.2. We discuss the formulation of variational inference of a sparse approximation in Section 5.2.3, which is of relevance to the method proposed in this chapter.

5.2.1 A short review of GP regression models

A GPR model is a supervised machine learning approach, whose goal is to construct a regression model to predict continuous quantities of interest given a set of observations. A GP is a set of random variables, any finite subset of which follows a Gaussian distribution. We observe that a GP is fully defined by its first and second moments. Without loss of generality, we take the mean function $m(\boldsymbol{x})$ to be zero. The covariance function $k(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$, also called the kernel function, is parametrized by a small set of hyper-parameters $\boldsymbol{\theta}$, e.g., the variance of the kernel and the lengthscales of the input dimensions, thus incorporating prior knowledge of the smoothness of the stochastic process and the similarity between data points.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote a training data set of d-dimensional inputs $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ and the corresponding real-valued realization $\mathbf{y} = [y_1, \ldots, y_n]^T$ of a latent function $f(\boldsymbol{x})$ corrupted by some Gaussian white noise ε, i.e.,

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \text{where } \varepsilon_i \sim \mathcal{N}(0, \sigma_y^2),$$

where σ_y^2 is the variance of the noise. We assume a zero-mean GP prior over the latent function, i.e., $f(\boldsymbol{x}) \sim \mathrm{GP}\left(0, k(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})\right)$. Given the noisy dataset, this can be expressed by the marginal likelihood

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{nn} + \sigma_y^2 \mathbf{I}_n),$$

where $\mathbf{K}_{nn}$ is the $n \times n$ covariance matrix with $[\mathbf{K}_{nn}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$, and $\mathbf{I}_n$ is the $n-$dimensional identity matrix. For the sake of convenience, we consider the variance of the noise σ_y^2 as an additional hyper-parameter belonging to the set $\boldsymbol{\theta}$. The performance of a GPR model, i.e., its ability to make accurate predictions, strongly depends on the hyper-parameters. The optimal hyper-parameters are estimated from the training data $\mathcal{D}$ by minimizing the negative log likelihood over the space of hyper-parameters:

$$\boldsymbol{\theta}_{\text{opt}} = \arg\min_{\boldsymbol{\theta}} - \log\left[p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\right],$$

where

$$\begin{aligned} \log\left[p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\right] &= \log\left[\mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{nn} + \sigma_y^2 \mathbf{I}_n)\right] \\ &= -\frac{1}{2}\mathbf{y}^T(\mathbf{K}_{nn} + \sigma_y^2 \mathbf{I}_n)^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_{nn}| - \frac{n}{2}\log 2\pi. \end{aligned} \tag{5.1}$$

To predict the function values at p new test inputs $\mathbf{X}_* = [\mathbf{x}_{*1}, \ldots, \mathbf{x}_{*p}]$, one assumes a joint GP prior of the latent function values for the training data $\mathbf{f}_n = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^T$ and the unobserved function values $\mathbf{f}_* = [f(\mathbf{x}_{*1}), \ldots, f(\mathbf{x}_{*p})]$, i.e.,

$$p(\mathbf{f}_n, \mathbf{f}_*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{nn} & \mathbf{K}_{n*} \\ \mathbf{K}_{*n} & \mathbf{K}_{**} \end{bmatrix}\right). \tag{5.2}$$

Here, $\mathbf{K}_{*n} = \mathbf{K}_{n*}^T$ is the covariance matrix between the new inputs $\mathbf{X}_*$ and the training samples $\mathbf{X}$, i.e., $[\mathbf{K}_{*n}]_{ij} = k(\mathbf{x}_{*i}, \mathbf{x}_j; \boldsymbol{\theta}_{\text{opt}})$. Thus, the noise-free posterior distribution is obtained by conditioning the targets $\mathbf{f}_*$ on the observations $\mathbf{y}$. It has the predictive posterior mean and variance estimates

$$\begin{aligned} m_{\mathbf{y}}(\boldsymbol{x}_*) &= \mathbf{K}_{*n}(\mathbf{K}_{nn} + \sigma_y^2 \mathbf{I}_n)^{-1}\mathbf{y}, \\ k_{\mathbf{y}}(\boldsymbol{x}_*, \boldsymbol{x}_*) &= k(\boldsymbol{x}_*, \boldsymbol{x}_*; \boldsymbol{\theta}_{\text{opt}}) - \mathbf{K}_{*n}(\mathbf{K}_{nn} + \sigma_y^2 \mathbf{I}_n)^{-1}\mathbf{K}_{n*}. \end{aligned} \tag{5.3}$$

We finally remark that the performance of the predictive distribution peaks with a correct choice of the kernel function followed by an accurate estimation of the hyper-parameters. Among the commonly used covariance functions, we consider the *automatic relevance determination squared exponential (ARD-SE)* kernel and the *ARD exponential (ARD-E)* kernel, defined as

$$\begin{gathered} k_{\text{ARD-SE}}(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) := \sigma_f^2 \exp\left(-\frac{1}{2} r\right) \text{ and } k_{\text{ARD-E}}(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) := \sigma_f^2 \exp\left(-\sqrt{r}\right), \\ \text{where } r = \sum_{j=1}^{d} \frac{(\boldsymbol{x}_j - \boldsymbol{x}_j')^2}{\sigma_j^2}, \end{gathered} \tag{5.4}$$

respectively. Above, $\boldsymbol{\theta} := [\sigma_f^2, \sigma_1^2, \ldots, \sigma_d^2]$, where σ_f^2 is the output variance, which determines the average distance of the function away from its mean and σ_j^2 are the characteristic lengthscales for $j = 1, \ldots, d$. We observe that the term ARD refers to the possibility of using a separate lengthscale for each predictor. For more details on GPR models and kernel functions we refer the reader to [WR06; WR96; Mur12].

5.2.2 Sparse GPR models

The non-parametric nature of GPR models makes them popular for the prediction of continuous functions. However, the training of a GPR model leads to a cubic scaling of the computational cost with the number of training samples. This complexity prevents GPRs to be used for big data sizes. To overcome this disadvantage, sparse approximations of GPR methods have been developed, providing an efficient training process that scales linearly with the number of training data. These methods rely on $m \ll n$ auxiliary latent variables, evaluated at some inputs $\mathbf{Z} \subset \mathbb{R}^m$, referred to as the *inducing inputs* or equivalently as *inducing points* or *pseudo-inputs*, to reduce the computational requirements to $\mathcal{O}(nm^2)$. This makes the sparse GPR competitive among machine learning methods for large data sets.

Following [QCR05], we present an overview of well-known sparse GPR methods. These approaches seek to modify the prior from (5.2) in a way to reduce the computational requirements of computing the posterior distribution (5.3). The latent variables $\mathbf{f}_m$, evaluated at the corresponding inducing points $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_m]^T$, are values of the Gaussian process (as also $\mathbf{f}_n$ and $\mathbf{f}_*$). Hence, we can recover the exact prior $p(\mathbf{f}_n, \mathbf{f}_*)$ simply by marginalizing out $\mathbf{f}_m$ from the joint GP prior $p(\mathbf{f}_n, \mathbf{f}_*, \mathbf{f}_m)$

$$p(\mathbf{f}_n, \mathbf{f}_*) = \int p(\mathbf{f}_n, \mathbf{f}_*, \mathbf{f}_m) \mathrm{d}\mathbf{f}_m = \int p(\mathbf{f}_n, \mathbf{f}_* | \mathbf{f}_m) p(\mathbf{f}_m) \mathrm{d}\mathbf{f}_m,$$

where the inducing prior $p(\mathbf{f}_m) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{mm})$. A crucial assumption common to all the sparse models models is that the training latent variables $\mathbf{f}_n$ and the test variables $\mathbf{f}_*$ are conditionally independent given the inducing variables $\mathbf{f}_m$, i.e., they can be expressed in two separate conditional distributions. This allows to approximate the joint prior as follows

$$p(\mathbf{f}_n, \mathbf{f}_*) \simeq \hat{p}(\mathbf{f}_n, \mathbf{f}_*) = \int \hat{p}(\mathbf{f}_* | \mathbf{f}_m) \hat{p}(\mathbf{f}_n | \mathbf{f}_m) p(\mathbf{f}_m) \mathrm{d}\mathbf{f}_m. \tag{5.5}$$

The various traditional sparse GP approaches differ by the choice of the conditional distribution approximations $\hat{p}(\mathbf{f}_* | \mathbf{f}_m)$ and $\hat{p}(\mathbf{f}_n | \mathbf{f}_m)$. We mention here three algorithms, by chronological appearance, which build upon one another to achieve better approximations of the true prior.

The sparse greedy approximation to GPR proposed in [SB01] formulates the approximate joint prior (5.5) as

$$\hat{p}_1(\mathbf{f}_n, \mathbf{f}_*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \hat{\mathbf{K}}_{nn} & \hat{\mathbf{K}}_{n*} \\ \hat{\mathbf{K}}_{*n} & \hat{\mathbf{K}}_{**} \end{bmatrix}\right).$$

Here $\hat{\mathbf{K}}_{ab} = \mathbf{K}_{am} \mathbf{K}_{mm}^{-1} \mathbf{K}_{mb}$ is the Nyström approximation of the true prior covariance $\mathbf{K}$, which leverages the information provided by the m inducing inputs. Intuitively, $\hat{\mathbf{K}}_{nn}$ and $\hat{\mathbf{K}}_{**}$ quantify how much information $\mathbf{f}_m$ provides about $\mathbf{f}_n$ and $\mathbf{f}_*$, respectively. The main drawback of this approach is that $\hat{\mathbf{K}}$ has only m degrees of freedom, i.e., the joint prior is degenerate, which results in overconfident predictions over a very limited family of functions.

An alternative approximation is proposed in [CO02; SWL03], where the exact prior variance matrix $\mathbf{K}_{**}$ is employed instead of its approximaton by the inducing variables:

$$\hat{p}_2(\mathbf{f}_n, \mathbf{f}_*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \hat{\mathbf{K}}_{nn} & \hat{\mathbf{K}}_{n*} \\ \hat{\mathbf{K}}_{*n} & \mathbf{K}_{**} \end{bmatrix}\right). \tag{5.6}$$

In this way $\mathbf{f}_*$ retains its own prior variance, leading to more reasonable predictive uncertainties than those given by the previous model, even if the predictive means are identical.

Further improvements on the joint kernel approximation have been made in [SG06] with the Sparse Pseudo-input Gaussian processes (SPGPs) approximation, where a more sophisticated joint prior is employed:

$$\hat{p}_3(\mathbf{f}_n, \mathbf{f}_*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \hat{\mathbf{K}}_{nn} + \mathrm{diag}[\mathbf{K}_{nn} - \hat{\mathbf{K}}_{nn}] & \hat{\mathbf{K}}_{n*} \\ \hat{\mathbf{K}}_{*n} & \mathbf{K}_{**} \end{bmatrix}\right). \tag{5.7}$$

Note that, as opposed to the previous two methods, the diagonal of $\hat{\mathbf{K}}_{nn}$ is corrected to be the exact one, thus imposing an additional independence assumption about the training conditional distribution $\mathbf{f}_n$ given $\mathbf{f}_m$.

A particular note should be made about the inducing variables, which, depending on the approach, can either be a subset of the training set $\mathbf{X}$ or arbitrary locations in the input space. The former selection strategy leads to a prohibitive combinatorial optimization, for which sub-optimal greedy-like solutions have been proposed to alleviate the computational complexity, see e.g., [SS00; SB01; SWL03; Tit09]. Nevertheless, relaxing the constraint on the inducing variables to be a subset of the training data can potentially lead to a better local optimum, as the optimization is continuous and the target space is now larger. However, we observe that, in both cases, reaching the global minimum is intractable and one can only expect to converge to a good local minimum. This limitation is common to the optimization of marginal likelihood functions, which are often non-convex with respect to the hyper-parameters. A common approach to overcome this issue is to use multiple starting points for both the hyper-parameters and the inducing inputs [CW18]. Ultimately, by considering the inducing inputs $\mathbf{Z}$ as extra kernel hyper-parameters that parametrize the covariance, their optimal values can be obtained simultaneously by minimizing the negative log likelihood, i.e.,

$$\begin{aligned} (\mathbf{Z}_{\mathrm{opt}}, \boldsymbol{\theta}_{\mathrm{opt}}) &= \underset{\mathbf{Z},\boldsymbol{\theta}}{\arg\min} - \log\left[\hat{p}(\mathbf{y}|\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta})\right] \\ &= \underset{\mathbf{Z},\boldsymbol{\theta}}{\arg\min} - \log\left[\mathcal{N}(\mathbf{y}|\mathbf{0}, \hat{\mathbf{K}}_{nn} + \boldsymbol{\Lambda} + \sigma_y^2\mathbf{I}_n)\right], \end{aligned} \tag{5.8}$$

where, $\boldsymbol{\Lambda}$ is specific specific to the chosen sparse approach, e.g., $\boldsymbol{\Lambda} = \mathrm{diag}[\mathbf{K}_{nn} - \hat{\mathbf{K}}_{nn}]$ for the SPGP method with prior covariance $\hat{p}_3$, defined in (5.7).

We finally remark that the quantities in (5.8) are trained at cost $\mathcal{O}(nm^2)$, while the computational complexities for the predictive mean and variance are $\mathcal{O}(m)$ and $\mathcal{O}(m^2)$, respectively. We refer the reader to [QCR05] and references therein for more details on the similarities and differences of various sparse methods for GPR.

5.2.3 Variational inference of sparse GPR

An alternative to the exact inference is variational inference. Instead of minimizing the negative log likelihood (5.8), variational inference seeks an approximation of the true GP posterior $p(\mathbf{f}_*|\mathbf{y})$ among a given family of distributions. Observing the differences between the marginal log likelihoods (5.1) and (5.8), one can interpret the sparse algorithms as an exact inference with an approximated prior with respect to the full GP prior, as suggested in [QCR05]. Therefore, a continuous optimization of (5.8) with respect to **Z** will not converge to the true GP model. Variational inference instead seeks to overcome this by considering the inducing inputs as variational parameters, whose optimal values are to be estimated jointly with the hyper-parameters.

Following [Tit09], a variational Gaussian distribution $q(\mathbf{f}_n)$ is chosen to approximate the exact posterior $p(\mathbf{f}_n|\mathbf{y})$ on the training function values $\mathbf{f}_n$, such that, with the assumption of conditional independence of $\mathbf{f}_n$ and $\mathbf{f}_*$ given the inducing variables $\mathbf{f}_m$, $p(\mathbf{f}_n|\mathbf{y})$ can be approximated by the variational posterior

$$q(\mathbf{f}_n) = \int p(\mathbf{f}_n|\mathbf{f}_m)q(\mathbf{f}_m)\mathrm{d}\mathbf{f}_m.$$

The optimized inducing variables and hyper-parameters are thus obtained by minimizing the Kullback-Leibler (KL) divergence between the true posterior and the variational posterior. In [Tit09], it is proposed to minimize the KL divergence of the augmented true posterior $p(\mathbf{f}_n, \mathbf{f}_m|\mathbf{y})$ and the augmented variational posterior $q(\mathbf{f}_n, \mathbf{f}_m) = p(\mathbf{f}_n|\mathbf{f}_m)q(\mathbf{f}_m)$. This is equivalent to maximize the variational lower bound

$$\mathcal{L}(\boldsymbol{Z}, \boldsymbol{\theta}) = \log\left[\mathcal{N}(\mathbf{y}|\mathbf{0}, \hat{\mathbf{K}}_{nn} + \sigma_y^2\mathbf{I}_n)\right] - \frac{1}{2\sigma_y^2}\mathrm{Tr}(\mathbf{K}_{nn} - \hat{\mathbf{K}}_{nn}), \tag{5.9}$$

where the second term is the negative trace of $\mathbf{K}_{nn} - \hat{\mathbf{K}}_{nn}$ scaled with $(2\sigma_y^2)^{-1}$ and $\hat{\mathbf{K}}_{nn}$ is defined as in Section 5.2.2. The resulting $(\mathbf{Z}_{\mathrm{opt}}, \boldsymbol{\theta}_{\mathrm{opt}})$ can be used to build the predictive distribution, which is given by

$$q(\mathbf{f}_*|\mathbf{y}) = \mathcal{N}\left(\hat{\mathbf{K}}_{*n}(\hat{\mathbf{K}}_{nn} + \sigma_y^2\mathbf{I}_n)^{-1}\mathbf{y}, \mathbf{K}_{**} - \hat{\mathbf{K}}_{*n}(\hat{\mathbf{K}}_{nn} + \sigma_y^2\mathbf{I}_n)^{-1}\hat{\mathbf{K}}_{n*}\right). \tag{5.10}$$

We note that this is exactly what it used in [CO02; SWL03], i.e., the approximation with a joint prior (5.6). In terms of the predictive distribution the two methods are the same. However, the variational method, with the extra regularization term, relies on a very different selection of the inducing inputs and the hyper-parameters. As opposed to the exact inference defined in (5.8), this additional trace term acts as a regularizer of the log likelihood, i.e., it summarizes the total variance of the conditional prior $p(\mathbf{f}_n|\mathbf{f}_m)$ and, as such, it can be viewed as an accuracy indicator of predicting $\mathbf{f}_n$ given $\mathbf{f}_m$. Minimizing this term prompts a good overall estimate of the statistics of the training data. We further note that, in the variational inference setting, the inducing variables **Z** determine the flexibility of both $p(\mathbf{f}_n|\mathbf{f}_m)$ and $q(\mathbf{f}_m)$, and, hence, the posterior $q(\mathbf{f}_n|\mathbf{y})$.

5.3 Variational approximation for systematic sensor placement

We now discuss a systematic sensor placement strategy in the context of anomaly detection for SHM. We therefore assume that only synthetic data generated by undamaged configurations under different environmental and operational conditions are available, i.e., we have no information about the type and severity of the anomalies. As discussed in the previous chapters, this is a realistic assumption because it is likely that many different types of damages will occur in the life time of a structure. If, on one hand, simulating all possible damages and locations would not be computationally feasible, it would on the other hand not be reasonable to make the hypothesis that including in the training set only a few representative damage types will generalize well to other types and locations; instead, it is more likely that mis-detections would occur. On the contrary, as described in Sections 4.1 and 4.2, anomaly detection strategies detect damages by characterizing the similarities among healthy samples and identify as damaged new samples with significantly different properties from the undamaged ones, see e.g., [PCCT14]. Mathematically, this corresponds to unsupervised or semi-supervised learning techniques as opposed to supervised algorithms, where a different class is assigned to every different type (or location) of damage. This poses a significant challenge in the context of sensor placement where one has to define a suitable cost function to be optimized with respect to the operational parameters, e.g., the candidate locations for the sensor placement, the available number of sensors and so on. Indeed, existing cost functions are usually formulated in terms of damage detectability, see e.g., [OSM19], which is a well defined concept only when a finite number of damages is assumed.

To overcome this obstacle, we propose to train a sparse GPR model of the monitoring phenomena, represented here by a chosen quantity of interest, e.g., displacement, stress or a function of those, by means of variational inference. By fixing the number m of inducing variables as the number of sensors that the user wishes to place on the structure, we identify the sensor locations with the local optima $\mathbf{Z}_{\text{opt}}$, obtained from the optimization of the variational lower bound (5.9). Then, the learned sparse GP model can be used to predict the effect of having placed sensors at particular locations $\mathbf{Z}_{\text{opt}}$. We recall that the optimal inducing variables $\mathbf{Z}_{\text{opt}}$ are such that the KL divergence between $q(\mathbf{f}_n)$ and the true posterior $p(\mathbf{f}_n|\mathbf{y})$ is minimized. On one hand, if $q(\mathbf{f}_n)$ is a good approximation of the exact posterior distribution $p(\mathbf{f}_n|\mathbf{y})$, this implies that the inducing variables provide enough statistics for the observed data, i.e., the information in the training data $\mathbf{f}_n$ can be compressed well in $\mathbf{f}_m$. As a consequence, the sensor locations $\mathbf{Z}$ do not cluster on the boundaries of the input domain, thus preventing "waste" in the sensed information. On the other hand, minimizing the regularizing trace term in (5.9), which represents the total variance of the conditional prior distribution $p(\mathbf{f}_n|\mathbf{f}_m)$, ensures that the mean square error of reconstructing the training latent values $\mathbf{f}_n$ from the inducing variables $\mathbf{f}_m$ is small. Indeed, the variational approximation guarantees that the sparse predictive distribution is as close as possible to the exact predictive distribution. This minimizes the reconstruction error not only at the sensor locations, but in the rest of the domain too. Hence, leveraging the variational sparse GPR for optimal sensor placements provides a tool to maximize the statistical information gain on the whole computational domain when using a fixed number of sensors, while reducing the computational requirements when compared to a traditional GP kernel based method.

In this section we elaborate on how the numerical data obtained from healthy structures, as described in Chapters 3 and 4, and the variational sparse GPR presented in Section 5.2.3 are combined in an optimal sensor placement strategy. After introducing the notation, in Section 5.3.1 we present details on placing sensors through variational inference of sparse GP for one particular structure configuration, while in Section 5.3.2 we describe how we handle the parametric dependency characteristic of each configuration in the context of optimal sensor placement. In Section 5.3.1, emphasis is given to an *ad-hoc* optimization setup which allows, on one hand, to constrain sensors to lie on a specific portion of the domain and, on the other hand, to deal with extremely large input data. Both requirements are indeed common in the context of SHM, where structures may be represented by billions of degrees of freedom and only certain locations might be admissible for sensor placement. We conclude with a description on how this procedure can be used to provide information about the sensitivity of a fixed network of sensors in Section 5.3.3.

Let us consider the d-dimensional spacial domain $\Omega \subset \mathbb{R}^d$, introduced in Chapter 3. Consider a suitable triangulation $\mathcal{T}_{\bar{h}}$, where $\bar{h}$ represents the mesh size, leading to a total of n_{dof} mesh points[1] which we indicate as $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{dof}}}]$. Moreover, let $\mathcal{P} \subset \mathbb{R}^{d_\mu}$ be a d_μ-dimensional domain representing the space of natural variations of the parameters of a healthy structure, e.g., different operational loads, external excitements and material properties. Then, for a given parameter combination $\boldsymbol{\mu} \in \mathcal{P}$, the generic output of interest is defined as

$$\mathbf{Y}(\boldsymbol{\mu}) = [\mathbf{y}_1(\boldsymbol{\mu}), \dots, \mathbf{y}_{n_{\text{dof}}}(\boldsymbol{\mu})], \tag{5.11}$$

where $\mathbf{y}_i(\boldsymbol{\mu})$ for $i = 1, \dots, n_{\text{dof}}$ are the d_y-dimensional outputs of interest measured at location $\mathbf{x}_i$, e.g., displacements of an elastic structure or features extracted from time-dependent signals. We further assume that the inputs and outputs are mapped through a function f and that this process is corrupted by some Gaussian white noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_y^2)$, i.e.,

$$\mathbf{y}_i(\boldsymbol{\mu}) = f(\mathbf{x}_i; \boldsymbol{\mu}) + \boldsymbol{\varepsilon}, \quad \text{for } i = 1, \dots, n_{\text{dof}}. \tag{5.12}$$

In this thesis, $\mathbf{y}_i(\boldsymbol{\mu})$ are defined as the raw signal after feature extraction and these values are defined in (4.17).

---

[1] The triangulation $\mathcal{T}_{\bar{h}}$ should not be confused with $\mathcal{T}_h$, used for the numerical experiments in Chapter 4. $\mathcal{T}_{\bar{h}}$ provides here a set of points to be used to train the variational sparse GP and they corresponds to the vertices of a coarser mesh than $\mathcal{T}_h$. In general, $N_h \gg n_{\text{dof}}$.

We point out that, in contrast to most of the cases where GPRs are employed, in this work, the training outputs $\mathbf{Y}(\boldsymbol{\mu})$ are not experimental, but simulated. As a direct consequence, for a given parameter $\boldsymbol{\mu}$, the map from inputs to outputs is known exactly. Therefore, we do not focus on constructing a GPR model to predict the mean and variance of the outputs at new spatial locations. The novelty of our approach lies in the fact that the sparse GPR is adopted to place sensors systematically; placing a Gaussian prior on the input-output map, i.e., $f(\boldsymbol{x}) \sim \mathrm{GP}(\mathbf{0}, k(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}))$, allows us to employ the variational inference algorithm presented in Section 5.2.3 and thus to identify the location of sensors as the inducing inputs.

5.3.1 Constrained variational approximation

The variational learning of the hyper-parameters and the inducing inputs are obtained by maximizing the variational lower bound (5.9), which is in general an unconstrained non-convex optimization problem. Indeed, even if we may have positivity constraints on some hyper-parameters, e.g., the variance and lengthscales of the kernel function, the fact that we approximate the log value of those hyper-parameters transforms the problem to an unconstrained optimization. While this may not be an issue for the aforementioned hyper-parameters, which appear squared in the kernel functions (5.4), we do need to impose some locality constraints on the inducing points to prevent them from being outside the input domain, especially when this is non-convex. Moreover, in some particular scenarios in the framework of SHM, one has to consider that it may be only possible to place sensors on a portion of the asset, e.g., sensors should not be placed inside a solid 3D structure, or they could only be placed on the above-surface structure of an offshore wind turbine, or only on the core of a nuclear reactor, avoiding the reflector subdomain [ABDC+18].

We consider sensor placement for a specific configuration, i.e., the input parameter $\boldsymbol{\mu}$ is fixed in (5.12). For succinctness, we neglect the parameter dependence in this part, i.e., $\mathbf{y}_i = \mathbf{y}_i(\boldsymbol{\mu})$. Let n_s be the number of sensors to be placed and $\Omega_s \subset \Omega$ the admissible domain for sensor locations. To overcome the issues related to unconstrained optimization mentioned above, the minimization of the negative variational lower bound (5.9) is modified as

$$(\mathbf{Z}_{\mathrm{opt}}, \boldsymbol{\theta}_{\mathrm{opt}}) = \underset{\mathbf{Z} \in \Omega_s^{n_s}, \boldsymbol{\theta}}{\arg\min} -\mathcal{L}(\mathbf{Z}, \boldsymbol{\theta}), \tag{5.13}$$

where $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_{n_s}]^T \subset \mathbb{R}^{d \times n_s}$ is the collection of the n_s sensor locations and each one of them is constrained to belong to Ω_s. Depending on the complexity of Ω_s, the optimization problem (5.13) can be solved using different optimization algorithms. In general, when Ω_s is a continuous domain, classic gradient-based constrained optimization algorithms, see e.g., [VGO+20], can be employed. However, in real-life engineering applications, due to the complexity of Ω_s, it may be cumbersome to specify its boundaries analytically and, in such cases, it is worth to replace Ω_s with a discrete counterpart comprising a finite number of admissible points $|\Omega_s| \gg n_s$. This clearly poses a challenge for gradient-based techniques, which are not very efficient in discrete settings. To deal with real-world problems, classic iterative methods should be replaced with discrete optimization methods. Heuristic algorithms, e.g., the genetic algorithm (GA), have received increasing attention during the recent decade in the field of discrete optimization, see, e.g., [GZZZ04; OSM19], where GA has been used to address several optimal sensor placement problems. We refer to [Dav91; SD08] for a detailed description on the GA, a type of evolutionary optimization algorithm that takes inspiration in the natural selection and undergoes three main stages: selection, crossover, and mutation.

In this work, when the admissible domain is discrete, we propose to combine the gradient-based optimization with the GA to form an Expectation-Maximization (EM-)like algorithm. At first, we fix the inducing points $\mathbf{Z}$ and employ a gradient-based algorithm to optimize the hyper-parameters $\boldsymbol{\theta}$. We then fix the hyper-parameters and use the GA to find the optimal inducing points. We iterate over these two steps until convergence is reached. This approach is summarized in Algorithm 3. We observe that, in the discrete case, it is possible to add an additional constraint to include prior knowledge on the importance of each sensor location. When the available sensor locations are assigned with a specific cost value, a cost-constrained sensor placement approach could be included in the framework, see, e.g., [CABK18]. In this work however, we consider the case in which the training points either belong to the admissible domain or not, without assigning a specific relevance to each location. For the sake of completeness, we observe that, in case of continuous admissible domains Ω_s, one can either choose to combine the two optimization steps or to keep them separate by replacing the GA with another gradient-based constrained optimization to estimate the inducing points. The second approach is advantageous for continuous problems with a faster convergence and it is employed in this work when Ω_s is continuous. We finally remark that DEAP (Distributed Evolutionary Algorithms in Python) [FRG$^+$12] is the framework used for the numerical implementation of the GA examples presented in this work.

Algorithm 3 Variational approximation for systematic sensor placement

```
Input: training dataset $\{\mathbf{X}, \mathbf{Y}\}$, admissible set $\Omega_s$, and max iteration number $k_{\max}$
Set $k = 0$ and randomly initialize $\mathbf{Z}_k$ s.t. $\mathbf{z}_i \in \Omega_s$ for $i = 1, \dots, n_s$
while not converged and $k < k_{\max}$ do
    Compute the optimal hyper-parameters $\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{\theta}} -\mathcal{L}(\boldsymbol{Z}_k, \boldsymbol{\theta})$.
    Compute the optimal constrained locations $\boldsymbol{Z}_{k+1} = \arg\min_{\mathbf{z} \in \Omega_s \forall \mathbf{z} \in \mathbf{Z}} -\mathcal{L}(\boldsymbol{Z}, \boldsymbol{\theta}_{k+1})$
    Set $k = k + 1$
Set $\mathbf{Z}_{\text{opt}} = \mathbf{Z}_k$, $\boldsymbol{\theta}_{\text{opt}} = \boldsymbol{\theta}_k$
return $(\mathbf{Z}_{\text{opt}}, \boldsymbol{\theta}_{\text{opt}})$, the optimal constrained inducing points and hyper-parameters
```

5.3.2 Including parameter dependence in sensor placement

Let us reintroduce the parameter dependency and consider a set of n_μ parameters $\Xi^{n_\mu} = \{\boldsymbol{\mu}_m\}_{j=1}^{n_\mu}$, where $\boldsymbol{\mu}_j \in \mathcal{P}$ for $j = 1, \cdots, n_\mu$. Applying Algorithm 3 for all these parameters, we obtain a set of n_μ parameter-dependent inducing points $[\mathbf{Z}_{\text{opt}}(\boldsymbol{\mu}_1), \ldots, \mathbf{Z}_{\text{opt}}(\boldsymbol{\mu}_{n_\mu})]$, where $\mathbf{Z}_{\text{opt}}(\boldsymbol{\mu}_j)$ corresponds to the n_s optimal locations for the specific parametric underlying system defined by $\boldsymbol{\mu}_j \in \mathcal{P}$. Having a continuous mapping from the inputs to the outputs and under the assumption that the parameters in $\mathcal{P}$ only vary some accessory properties without altering the topology of the structure, it is reasonable to assume that each one of the n_s inducing points $\mathbf{Z}_{\text{opt}}(\boldsymbol{\mu}_j)$ lie in the neighborhood of the corresponding inducing point obtained for a different input parameter, i.e., $\mathbf{Z}_{\text{opt}}(\boldsymbol{\mu}_k)$ for $j \neq k$ and $j, k = 1, \ldots, n_\mu$. Therefore, to include the parametric dependency and summarize the information from this set of $n_s n_\mu$ into a set of n_s locations, we propose to employ the K-medoids algorithm to find n_s clusters and its corresponding centers.

Similar to the K-mean algorithm, K-medoids is a clustering algorithm that breaks the data set into a user-defined number of groups and minimizes the distance of the center of each cluster and the points in it. The difference between these two clustering algorithms is that the K-means algorithm averages points within a cluster as the center, whereas K-medoids selects only data points as cluster centers. In comparison, K-medoids is more robust as the algorithm seeks to minimize the sum of dissimilarities of all points inside a cluster instead of the sum of squared Euclidean distances, as used in the K-means algorithm. For this reason the K-means algorithm is considered to be more sensitive to noise and outliers [AV16]. We point out that, in the numerical examples, the clustering step is carried out in Matlab [MAT19] by employing the built-in function `kmedoids`. For more details on K-medoids algorithm, we refer the readers to [PJ09; AV16].

We summarize the algorithm for sensor placement that incorporates parameter variation of a solid structure in Algorithm 4. We notice that given different initial conditions, the K-medoids algorithm can lead to different clusters. The final decision can be made by either fixing the initial condition or by engineering experience across the resulting clusters. Finally, we observe that, in case of high variability in the training set with respect to the input parameters, the solution of the non convex optimization problem (5.13) may provide very different results in terms of the location of the inducing points for each input parameter, thus leading to a challenging clustering task. While for the problems considered in this thesis the smoothness of the mechanic systems and the relatively small parameter space contribute to having inducing points close to each other for various input parameters, in a different context, it may be worth to investigate alternative methods to find the optimal sensor locations for all n_μ parameters at once. We mention for example the possibility of replacing the parametric output of interest with the expectation of the output of interest with respect to the parameter.

Algorithm 4 Parametrized variational approximation for systematic sensor placement

```
Input: parametric training dataset ${\mathbf{X}, \mathbf{Y}(\boldsymbol{\mu}_j)}_{j=1}^{n_\mu}$ and admissible set $\Omega_s$
for $j = 1, \cdots, n_\mu$ do
    Apply Alg. 3 to ${\mathbf{X}, \mathbf{Y}(\boldsymbol{\mu}_j)}$ to get $n_s$ $\Omega_s$-constrained inducing inputs $\mathbf{Z}_{\text{opt}}(\boldsymbol{\mu}_j)$
Apply K-medoids algorithm to the $n_s n_\mu$ inducing inputs $[\mathbf{Z}_{\text{opt}}(\boldsymbol{\mu}_1), \ldots, \mathbf{Z}_{\text{opt}}(\boldsymbol{\mu}_{n_\mu})]$ to get $n_s$ clusters
Set $\boldsymbol{Z}_{\text{opt}}$ = cluster centers
return $\mathbf{Z}_{\text{opt}}$, optimal constrained sensor locations
```

5.3.3 A tool for sensor sensitivity

The technology proposed here can also be applied to answer a few related questions: (*i*) how many sensors are needed to achieve a prescribed precision? (*ii*) what is the expected sensitivity of a fixed sensor network? (*iii*) when a fixed network of n_s sensors already exists, given a budget of n_s^{extra} additional sensors, where should these be placed to achieve optimal coverage? Properly addressing these queries is of great importance in the maintenance of real-life engineering problems.

The first point refers to the need of defining a suitable measure to quantify the quality of the locations, whether they are obtained with the proposed variational approach or already placed on the monitored structure. A straightforward choice is to compute the reconstruction of the quantity of interest, i.e., $m^q_{\mathbf{Y}(\boldsymbol{\mu}_j)}(\mathbf{x}_i)$ at all training points $\mathbf{x}_i \in \mathbf{X}$, for $i = 1, \ldots, n_{\text{dof}}$. Here $m^q_{\mathbf{Y}(\boldsymbol{\mu}_j)}(\mathbf{x}_i)$ is the mean of the posterior distribution (5.10) of the sparse model based on the variational parameters, i.e., outcome of Algorithm 4. Hence, the relative reconstruction error of the quantity of interest at *unsensed* locations can be used as an indicator of the sensor sensitivity. On one hand this quantity increases as we move away from the sensors and, on the other hand, increasing the number n_s of sensors is expected to improve the global coverage. Moreover, we define the average relative reconstruction error over the n_μ samples as

$$R := \sum_{j=1}^{n_\mu} \frac{1}{n_\mu} \frac{\|\mathbf{Y}(\boldsymbol{\mu}_j) - m^q_{\mathbf{Y}(\boldsymbol{\mu}_j)}(\mathbf{X})\|}{\|\mathbf{Y}(\boldsymbol{\mu}_j)\|}, \tag{5.14}$$

where $\mathbf{Y}(\boldsymbol{\mu}_j)$ is the simulated quantity of interest (5.11). A low R value is an indicator of a good global placement which takes the parametric dependency of the structure into account. An additional indicator to quantify the quality of sensor placement is the point-wise relative variance reduction, defined as

$$V_i := \frac{\mathbf{K}_{im}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mi}}{\mathbf{K}_{ii}}, \quad \text{for } i = 1, \ldots, n_{\text{dof}}, \tag{5.15}$$

where $\mathbf{K}$ is the kernel matrix with optimized hyper-parameters defined in Section 5.2. This quantity expresses the level of variance reduction that can be achieved by including the chosen sensor locations. A relative variance reduction close to one indicates that the inducing variables alone can reproduce the full GP prediction well.

Finally, we note that in the variational inference framework of the proposed approach, it is possible to jointly optimize some inducing inputs and keep the already existing sensor locations fixed. Thus, the strategy presented in this chapter can be efficiently implemented to systematically place additional sensors while accounting for the already existing structural coverage.

5.4 Numerical results

In Sections 5.4.1, 5.4.2, and 5.4.3, we provide examples of sensor placement in two and three dimensions for which we use the methodology presented in Section 5.3. A wave-based monitoring strategy is employed for the 2D and 3D examples given in Sections 5.4.1 and 5.4.2, respectively. Here, we resort to the mean reconstruction error and the relative variance reduction to test the quality of the sensor locations. Section 5.4.3, instead, presents a real-life engineering example, for which a static monitoring approach is used. Taking into account the complexity of the geometry and the large number of degrees of freedom, tests to assess the good quality of the placement are performed by looking at the achieved accuracy in detecting damages. The synthetic databases used in the training phase are constructed following the procedure given in Chapters 3 and 4 and we refer to Section 4.4 for the implementation details.

5.4.1 Two-dimensional examples for the guided-wave problem

The examples in this section follow the wave-based monitoring approach, for which we train a variational sparse GP model with compressed signals. We consider the same governing problem (3.1) for three different geometries, shown in Figure 5.1, and we refer to these problems as Problems 1a, 1b, 1c, whose domains will be identified by Ω_a, Ω_b, and Ω_c, respectively. We note that Problem 1a is the 2D geometry used in Section 4.4.

As for the numerical example in Section 4.4, for each problem, we consider zero initial conditions for both the displacement and the velocity and prescribe homogeneous free slip boundary conditions (3.3). The high fidelity numerical solutions of (3.1) are computed using the FE approximation by $\mathbb{P}_1$ elements over a domain discretized in tetrahedral cells with a total of $N_h = 30'912$ degrees of freedom, while for the RB solver we rely on 267 basis for Problem 1a [2]. Similar order of magnitudes of these parameters are used for the other two problems: $N_h = 31'200$ and 284 basis for Problem 1b and $N_h = 26'072$

---

[2] With respect to Section 4.4, here we increase the number of frequencies used for Weeks method to 500, thus leading to a slighly larger number of reduced basis.

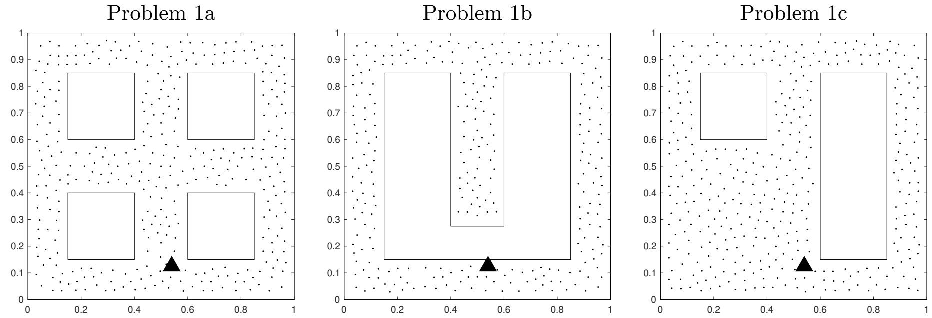
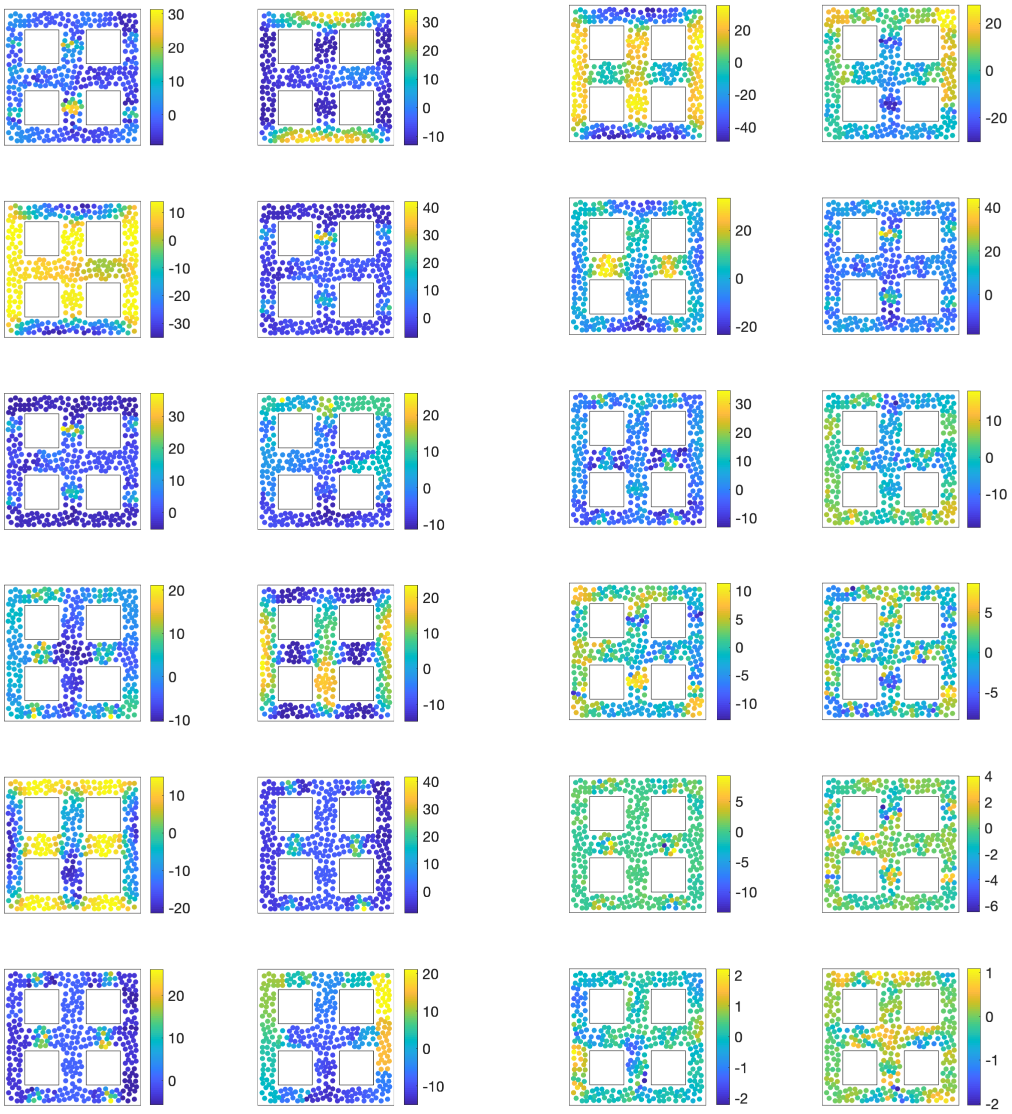


Figure 5.1 – 2D examples with different geometries: Problem 1a relies on 360 training inputs (*small black dots*), corresponding to the vertices of a coarse mesh over the domain, while Problems 1b and 1c have 286 and 375 training points, respectively. The location of the center of the active source is the same for three geometries and corresponds to $\bar{S} = [0.54, 0.125]$ (*black triangle*).

and 306 basis for Problem 1c. For the discretization in time, we consider $N_t = 20'000$ and $T = 20$ for the three problems. The natural variations are described by the three parameters, i.e., $\boldsymbol{\mu} = [E, \nu, k] \in \mathcal{P}$, where $\mathcal{P}$ is defined in (4.18). For each problem we consider $n_\mu = 100$ samples and, to obtain a well balanced dataset, we sample using a Sobol's sequence [JK08], i.e., a base-2 digit sequence which provides a successively finer uniform partition of the intervals $\mathcal{P}$.

The training points $\mathbf{X} \subset \mathbb{R}^{n_{\text{dof}}} \subset \Omega_i$ with $i = a, b, c$ are obtained by fixing the same size of a coarse mesh for the three problems, thus recovering $n_{\text{dof}} = 360$, $n_{\text{dof}} = 286$, and $n_{\text{dof}} = 375$ mesh points, for Problems 1a, 1b, and 1c respectively. We observe that the mesh points on the boundary are not included in the training set. This results to a practical choice due to the free-slip boundary conditions, for which at least one of the two displacement directions will be identically zero on each boundary edge. For each geometry, we consider d_y-dimensional feature vectors with $d_y := 3$, i.e., we consider the first three principal components of the $N_f = 12$ features, extracted from the $\boldsymbol{\mu}_1, \ldots \boldsymbol{\mu}_{n_\mu}$ discrete time-dependent displacement signals (3.30), are used to train the variational sparse GP (see Sections 4.3.1 and 4.3.2). We note that for Problems 1a, 1b, and 1c, the first three principal components account for more than 80% of the variability. By way of example, Figure 5.2 shows the normalized features over the n_μ samples and the corresponding principal components for Problem 1a with $\boldsymbol{\mu} = [1, 0.33, 2]$. Normalization is performed by features, i.e., the means $\bar{m}_1, \ldots, \bar{m}_{N_f}$ and variances $\bar{\sigma}_1, \ldots, \bar{\sigma}_{N_f}$ are computed for each one of the N_f features over all training points (e.g., $n_{\text{dof}} = 360$ for Problem 1a) and all simulations obtained for n_μ input parameters. We note that this is different to what described in Section 4.3.2, where normalization is performed for each datasets (4.13), sensor by sensor.

In terms of setup for the GPR, we note that for all the three examples, we use the ARD-



(a) Normalized features. (b) Principal components.

Figure 5.2 – Example of normalized features, extracted from the solution obtained by solving the acoustic-elastic problem on the geometry 1a with $\boldsymbol{\mu}_1 = [1, 0.33, 2]$ in (a). The first and second coloumn show the $Q = 6$ features related to the displacement along the x and y directions, respectively for a total of $N_f = 12$ features. The N_f corresponding principal components are shown in (b) ordered from left to right and from top to bottom. The first three principal components account for 60.5%, 13.3%, and 11.5% of the variability, respectively for a total of more than 85%. Similar values are obtained for all the other samples and, for the other two geometries, i.e., Problems 1b and 1c, the importance of the three components is more balanced. The mean and standard deviation used for the normalization are based on the features extracted from $n_\mu = 100$ samples, obtained using the first 100 parameters of a Sobol sequence based on $\mathcal{P}$.

Exponential kernel (5.4), which provides the best performance on the training set with respect to other popular choices, e.g., the Squared Exponential, Matérn-32 ad Matérn-52, either with automatic relevance detection (ARD), i.e., with a separate lengthscale for each predictor, or with the same lengthscale for each predictor.

By applying the sensor placement methodology, described in Section 5.3 for $\{\mathbf{X}, \mathbf{Y}(\boldsymbol{\mu}_j)\}_{j=1}^{n_\mu}$, we obtain the systematic placement of sensors shown in Figure 5.3. For each geometry, the plots overlay the locations of the $n_s = 4, 9, 16, 25$ inducing points obtained by applying Algorithm 3 n_μ times over the admissible domains Ω_a, Ω_b, and Ω_c, i.e., a total of $n_s n_\mu$ inducing inputs, sometimes overlapping, is shown. The sets of inducing points are compared with the corresponding centroids, obtained by applying Algorithm 4. As an example, the inducing points obtained by applying Algorithm 3 for the first Sobol's parameter $\boldsymbol{\mu}_1 = [1, 0.33, 2]$ are also shown. While for larger numbers of inducing points, clusters appear to be more visible, for smaller n_s, the location of the $n_s n_\mu$ inducing inputs shows more variability. This can be explained by the fact that the optimal inducing inputs are optimized to reconstruct different quantities of interests, which depend on the input parameter $\boldsymbol{\mu}$. However, one also has to consider that, when trying to reconstruct a non-trivial quantity of interest over a complex structure with only few n_s points, the sparse model might get stuck in a local minimum without reaching convergence. For example, the inducing points obtained for $\boldsymbol{\mu}_1$ for Problem 1a and $n_s = 9$ are not very well distributed over the entire domain. Nevertheless, the centroids seem to be a good summary of the entire underlying phenomena suggesting that our clustering approach provides a way not only to take into account different healthy variations, but also to mitigate the unfortunate outcome of being stuck in potential local minima in the optimisation process. Indeed, as shown in Figure 5.4, the optimal centroids obtained by clustering the results over the first $n_\mu = 10$ or the entire parameter domain, i.e., over $n_\mu = 100$ sample, are almost indistinguishable. We note that purple stars in Figure 5.4 correspond to the same centroids shown in Figure 5.3, i.e., obtained by averaging the results of $n_\mu = 100$ samples.

As mentioned in Section 5.3.3, two ways to quantify the quality of the sensor placement outcome are by means of the reconstruction error and the variance reduction. Figure 5.5 shows the point-wise mean reconstruction of the first sample for Problems 1a, i.e., $m^q_{\mathbf{Y}(\boldsymbol{\mu}_1)}(\mathbf{x}_i)$ with $\mathbf{x}_i \in \mathbf{X}$. We observe that as n_s increases, the different characteristics of the three principal components become visible in the reconstruction. We also note that the reconstruction accuracy achieved for the first principal component $\mathbf{Y}_1$ is higher than the one for the other two. Indeed, the highest variability of the first principal component corresponds to a less noisy field, which is simpler to reconstruct by means of GPR. We remark that similar results are obtained for Problems 1b and 1c, as shown in Figures 5.6 and 5.7, respectively. Figure 5.8 shows, for the three problems, the mean reconstruction error over the n_μ samples, defined in (5.14), for the three quantities of interests as a function of the number n_s of inducing points. These errors are compared to those obtained by reconstructing the principal components using the centroids as fixed variational hyper-parameters in a new sparse GPR model. We observe that the difference

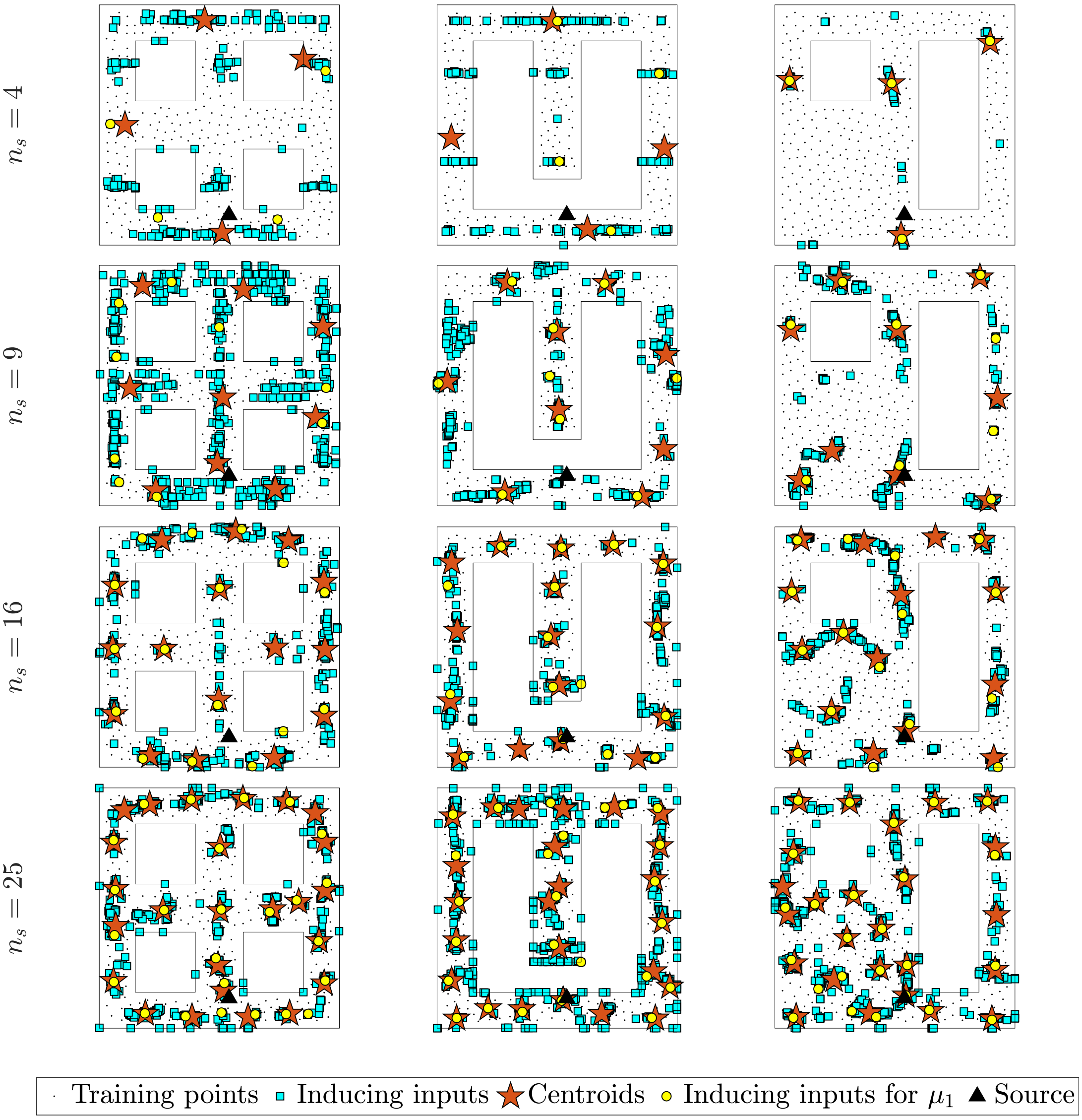


Figure 5.3 – Comparison of the location of inducing points obtained by applying Algorithm 3 for $n_\mu = 100$ samples (*cyan squares*) and the corresponding n_s centroids obtained with Algorithm 4 (*red stars*). The inducing points obtained for one particular sample, i.e., $\boldsymbol{\mu} = [1, 0.33, 2]$, are also shown (*yellow circles*). Each column shows a different geometry while each row shows a fixed number n_s of inducing points, which increases from top to bottom, i.e., $n_s = 4, 9, 16, 25$.

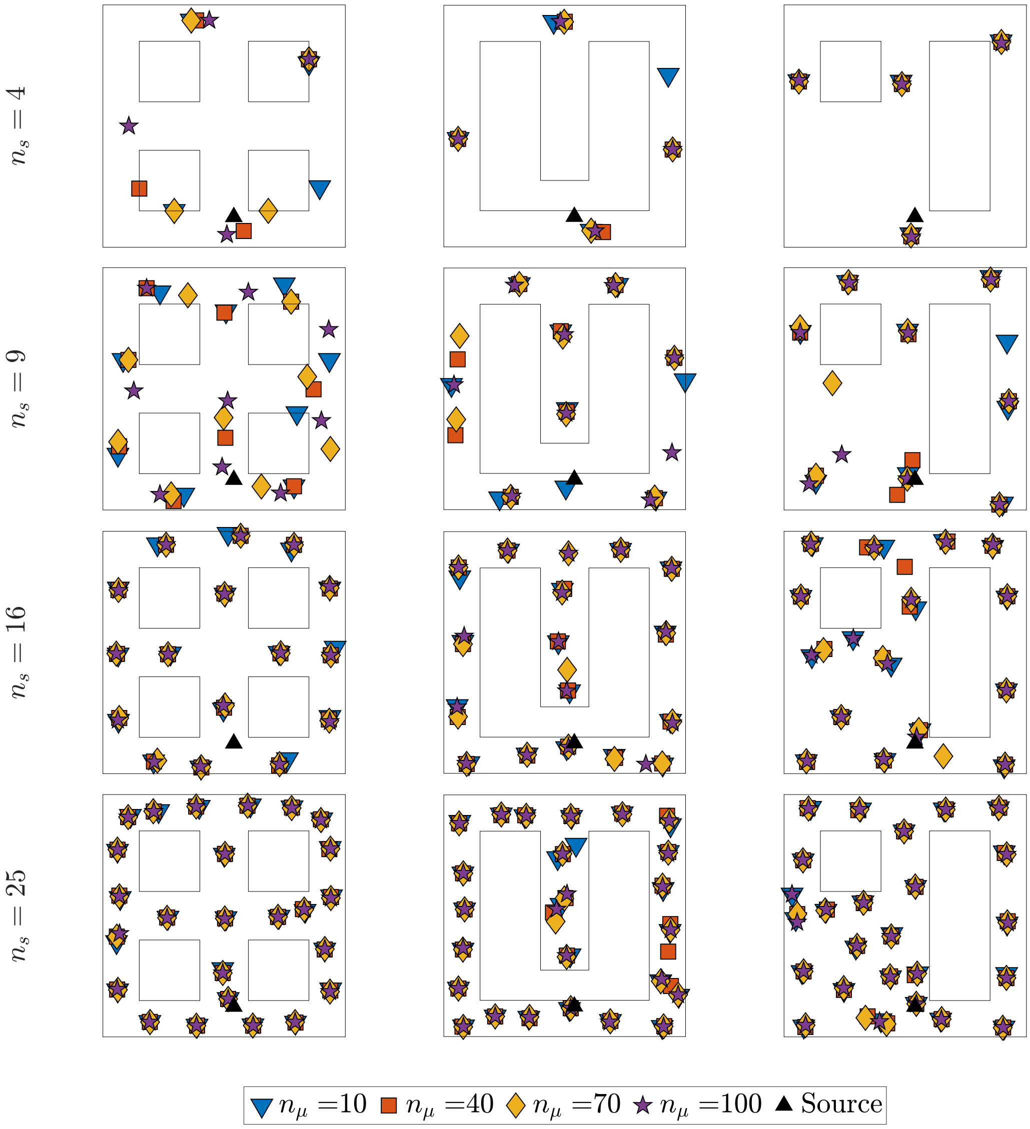


Figure 5.4 – Comparison of the centroids obtained with Algorithm 4 for different number of samples n_μ, namely $n_\mu = 10, 40, 70$ and 100. Each column shows a different geometry while each row shows a fixed number n_s of inducing points, which increases from top to bottom, i.e., $n_s = 4, 9, 16, 25$.

between these two is minimal, which suggest that the centroids are good approximations of the inducing points for sensor placement. Figure 5.9 shows the relative variance reduction (5.15), averaged over the n_μ samples. A variance reduction above 0.7 almost everywhere even for $n_s = 4$ is an indication of good sensor placement. Finally, Figure 5.10 shows the size of the clusters, i.e., the number of samples among the n_μ available data which belong to the n_s clusters. In an optimal scenario we expect each one of the n_s clusters to be composed of exactly n_μ samples. The fact that the clusters in Figure 5.10 all have similar sizes represents an additional validation for (i) the good quality of the K-medoid clusters and (ii) the fact that the clusters are a good summary of the underlying inducing points.

To conclude, Figure 5.11 compares the position of the centroids obtained with Algorithm 4 with the centroids obtained by applying the K-medoids algorithm to the training points $\mathbf{X}$ directly. This strategy is chosen as a proxy to place points *equidistantly* over a complex domain. Although this naive strategy may seem to give almost as good results as the laborious methodology followed to obtain the variational centroids, as shown in Figure 5.12, placing sensors without including physical information does not yield a good result. Indeed, the mean reconstruction accuracy obtained by training a new variational sparse GP model with fixed inducing inputs as the centroids obtained by K-medoids on the training points is not as good as the one obtained with variational centroids.

5.4.2 A three-dimensional example for the guided-wave problem

The sensor placement strategy following the guided-wave monitoring approach can be extended to 3D problems. Let us consider the geometry of a T-beam as shown in Figure 5.13. We consider the acoustic-elastic model (3.1) with zero initial conditions and homogeneous Dirichlet boundary conditions imposed on the surface $z = 0$ together with zero traction on the remaining surfaces. We compute the high fidelity solutions using the FE approximation by $\mathbb{P}_1$ elements over a fine mesh with $N_h = 262'863$ degrees of freedom and for the low fidelity model we obtain $N = 505$ basis. For the time discretization, we consider a shorter time window, i.e., $N_t = 10'000$ and $T = 10$. We consider the same parameter space (4.18) as for the 2D problem, where k is the free parameter of the active source function (4.20), centered at $\bar{S} = [0.7, 1, 2]$. The training dataset corresponds to $n_{\text{dof}} = 4688$ input points of a coarse mesh, restricted to the Neumann surfaces and output of interests of dimension $d_y = 4$, i.e., the first four principal components of the normalized $N_f = 18$ features, extracted from the discrete time signals, as described in Section 4.3.1. We note that the union of the first four principal component accounts for more than 90% of the total variability for all samples. By way of example, the first two components obtained for $\boldsymbol{\mu}_1 = [1, 0.33, 2]$ are shown in the first row of Figure 5.15. After executing Algorithm 4 for $n_\mu = 10$ Sobol's parameters, we obtain the inducing points and the centroids of the K-medoid clusters shown in Figure 5.14. Figure 5.15 also shows the mean reconstruction of the first two output of interest $m^q_{\mathbf{Y}_j(\boldsymbol{\mu}_1)(\mathbf{X})}$, for $j = 1, 2$, over the training set $\mathbf{X}$ for a fixed parameter $\boldsymbol{\mu}_1$ and increasing number of sensors, i.e.,

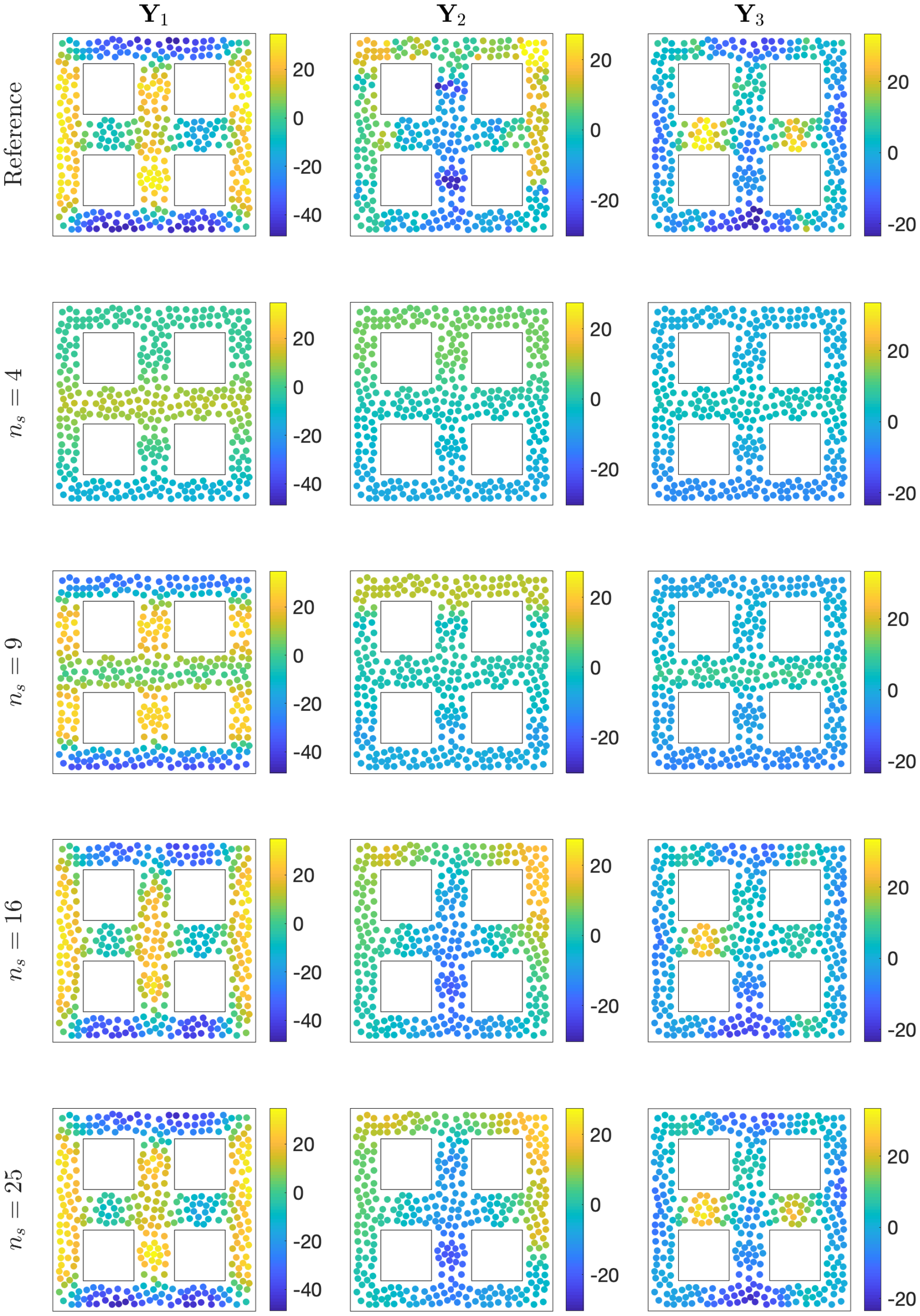


Figure 5.5 – Comparison of the first three principal components obtained for Problem 1*a* either by extracting the features from the time signals and then performing PCA (*first row*) or by sparse GP reconstruction using $n_s = 4, 9, 16$ or 25 inducing points (*second to fifth rows*). As the number of inducing points increases, the output of interests can be better reconstructed. The *reference* principal components correspond to the results obtained for $\boldsymbol{\mu}_1 = [1, 0.33, 2]$. The color scale is the same for the reference and the corresponding reconstructions.

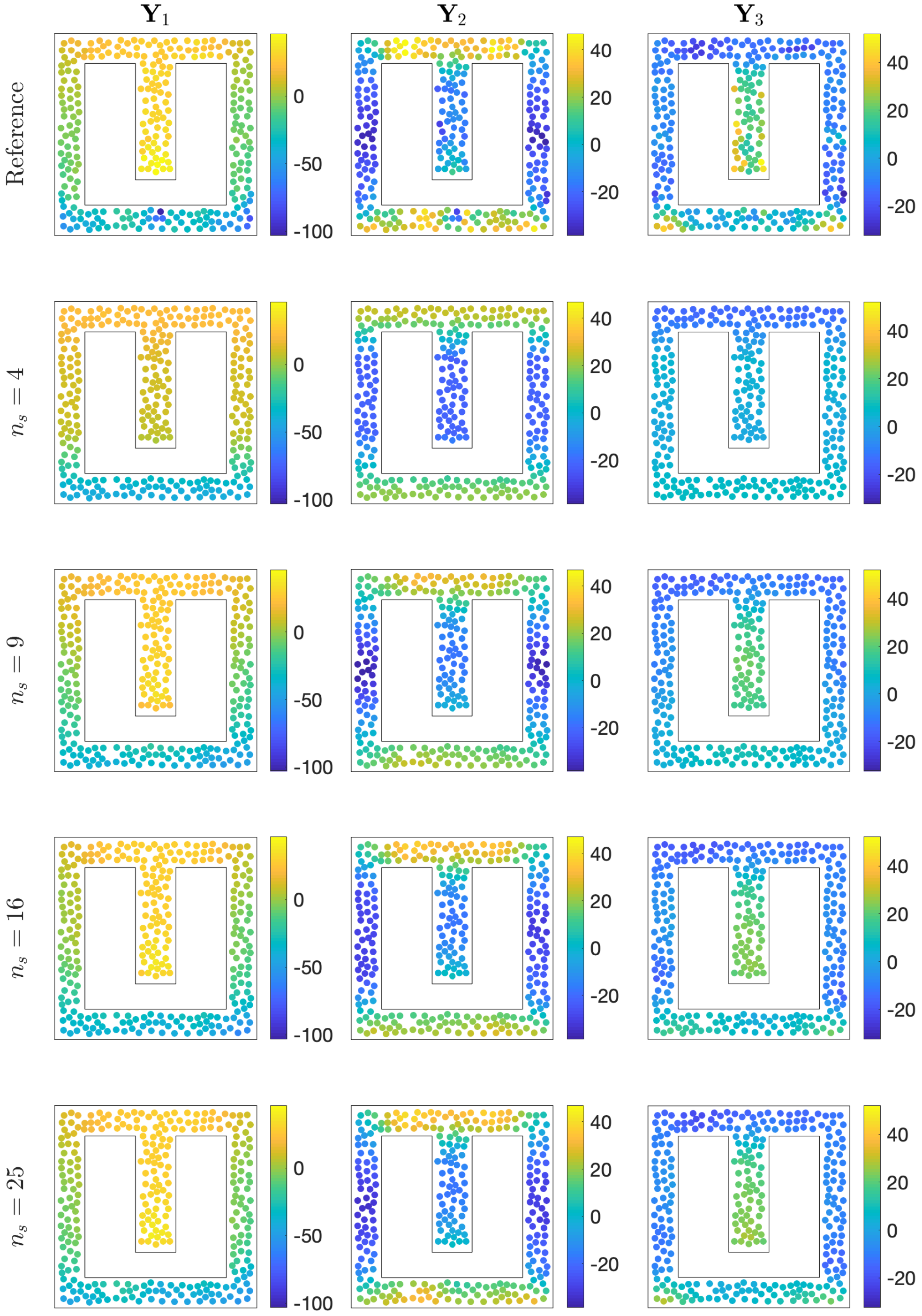


Figure 5.6 – Comparison of the first three principal components obtained for Problem 1*b* either by extracting the features from the time signals and then performing PCA (*first row*) or by sparse GP reconstruction using $n_s = 4, 9, 16$ or 25 inducing points (*second to fifth rows*). As the number of inducing points increases, the output of interests can be better reconstructed. The *reference* principal components correspond to the results obtained for $\boldsymbol{\mu}_1 = [1, 0.33, 2]$. The color scale is the same for the reference and the corresponding reconstructions.

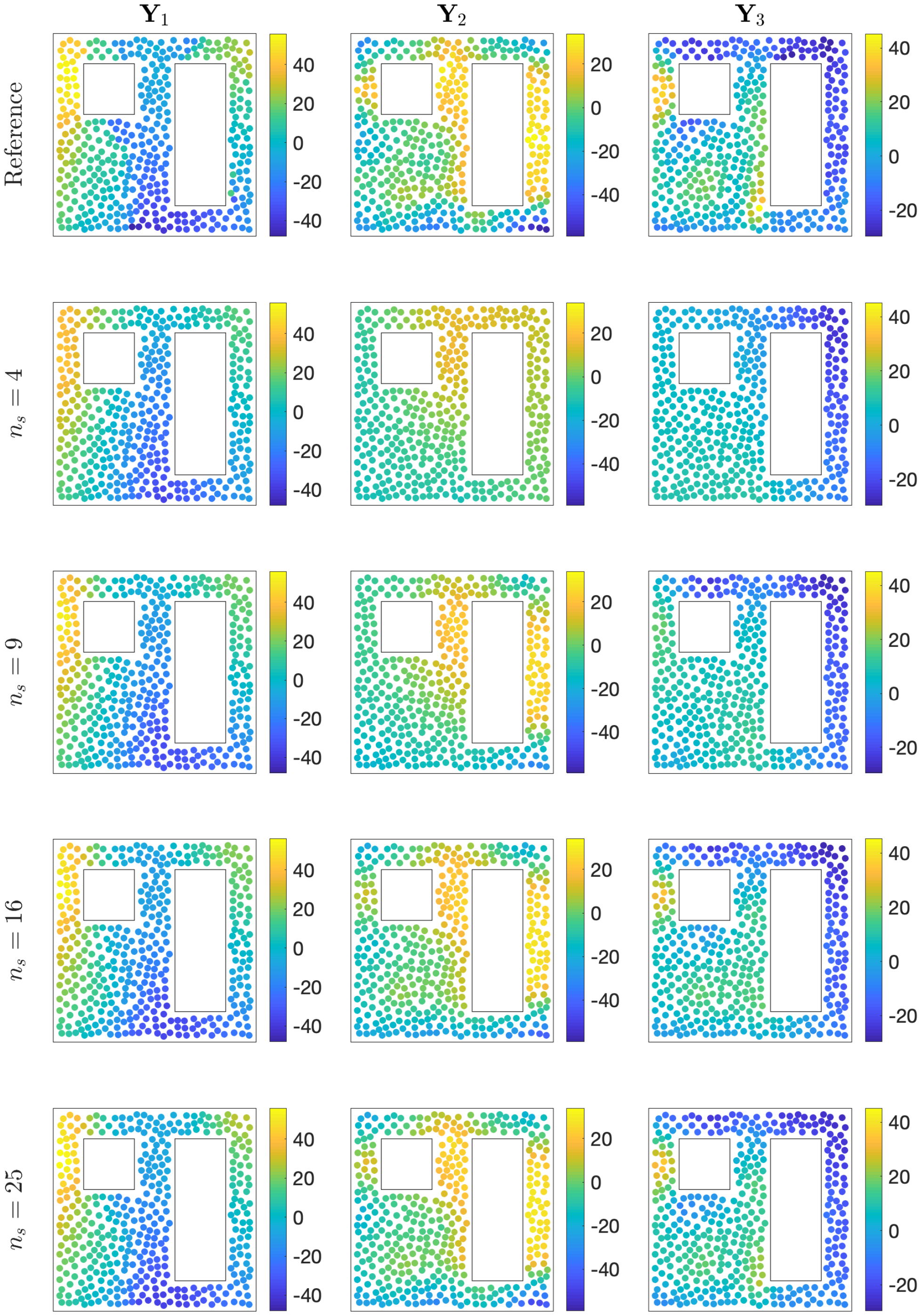


Figure 5.7 – Comparison of the first three principal components obtained for Problem 1*c* either by extracting the features from the time signals and then performing PCA (*first row*) or by sparse GP reconstruction using $n_s = 4, 9, 16$ or 25 inducing points (*second to fifth rows*). As the number of inducing points increases, the output of interests can be better reconstructed. The *reference* principal components correspond to the results obtained for $\boldsymbol{\mu}_1 = [1, 0.33, 2]$. The color scale is the same for the reference and the corresponding reconstructions.

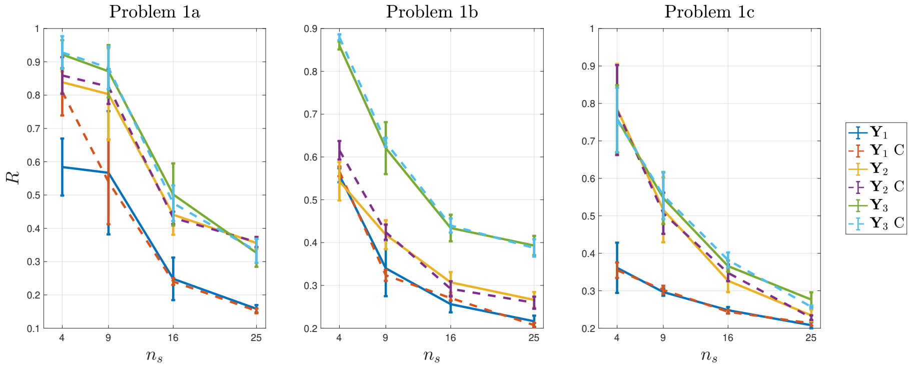


Figure 5.8 – Mean reconstruction errors with error-bars with respect to the number n_s of inducing points for the first, second, and third principal components (*solid lines*) used to train the variational sparse GP model. The corresponding mean reconstruction error, obtained by training a new variational sparse GP model with fixed inducing inputs corresponding to the centroids, is also shown (*dashed lines*). Each plot shows the result for one of the three geometries.

$n_s = 4, 16, 36$. As expected, the different characteristics of the output of interest become more visible in the predictions as the number of sensors increases. Finally, the relative variance reduction (5.15), with respect to the centroids and averaged over n_μ samples, is shown in Figure 5.16 for all training points. An overall relative reduction above 92% is achieved already for $n_s = 4$ sensors.

5.4.3 Application to a realistic geometry of an offshore jacket

We finally consider a real-life engineering example of an offshore jacket, as shown in Figure 5.17. The Static-Condensation Reduced-Basis-Element (SCRBE) solver from Akselos [sof20] is used to establish a component-based synthesis (CMS) and model order reduction framework. The problem consists of 192 components and the discrete full order model has more than four million degrees of freedom. To accelerate the solution process, the degrees of freedom can be drastically reduced by taking a random subset of points within each component as representatives of that component. In this way, the total number of degrees of freedom decreases to $N_h = 4632$. We refer to [HKP13] and to [Hur65; HL10] for more details on the SCRBE method and introduction to CMS, respectively. An example on how to use the SCRBE method to address local non-linearities in complex structures can be found in [ZGH19].

The bottom of the jacket is fixed on the ground and other boundaries are assumed to be free. We introduce two parameters, $\mu_x, \mu_y \in \mathcal{P} = [0.1, 1]$ kPa, representing the surface wind loads on the 64 components in the dark box in Figure 5.17 in the x and y directions, respectively and assume the jacket to be linear elastic with Young's modulus $E = 200$ GPa and Poisson's ratio $\nu = 0.3$. The displacements under different load combinations are

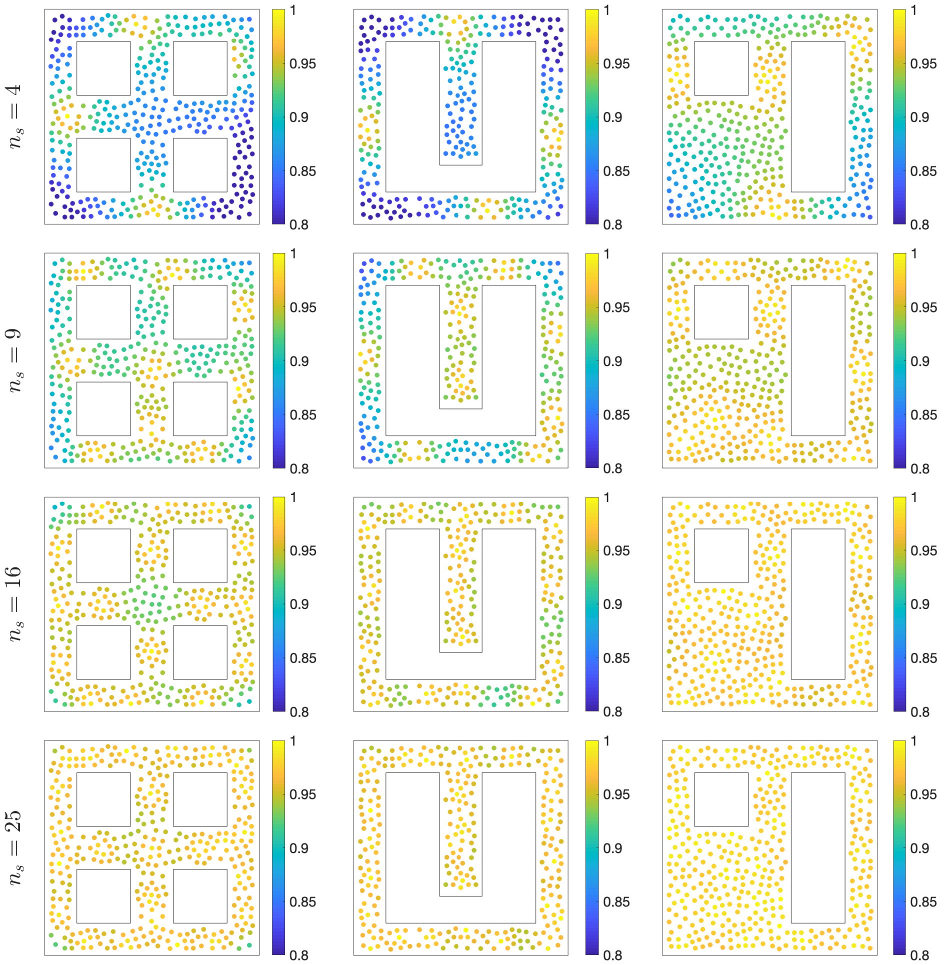


Figure 5.9 – Relative variance reduction (5.15) obtained using n_s centroids and averaged over n_μ samples. Each column shows a different geometry while each row shows a fixed number n_s of inducing points, which increases from top to bottom, i.e., $n_s = 4, 9, 16, 25$. The color scale is the same for all the plots.

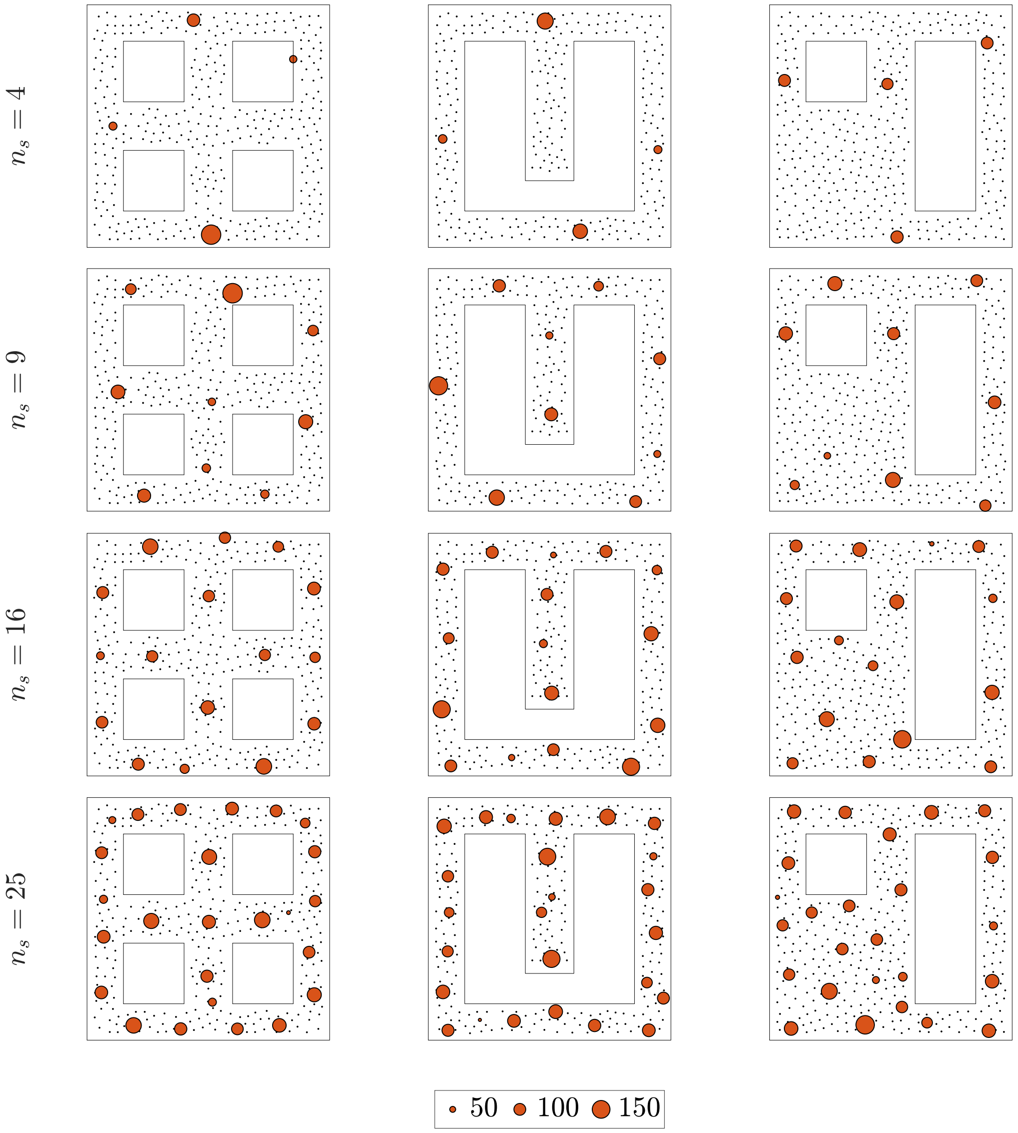


Figure 5.10 – Visualization of the number of sensors belonging to each of the n_s clusters by means of different bubble sizes. Each column shows a different geometry while each row shows a fixed number n_s of inducing points, which increases from top to bottom, i.e., $n_s = 4, 9, 16, 25$.

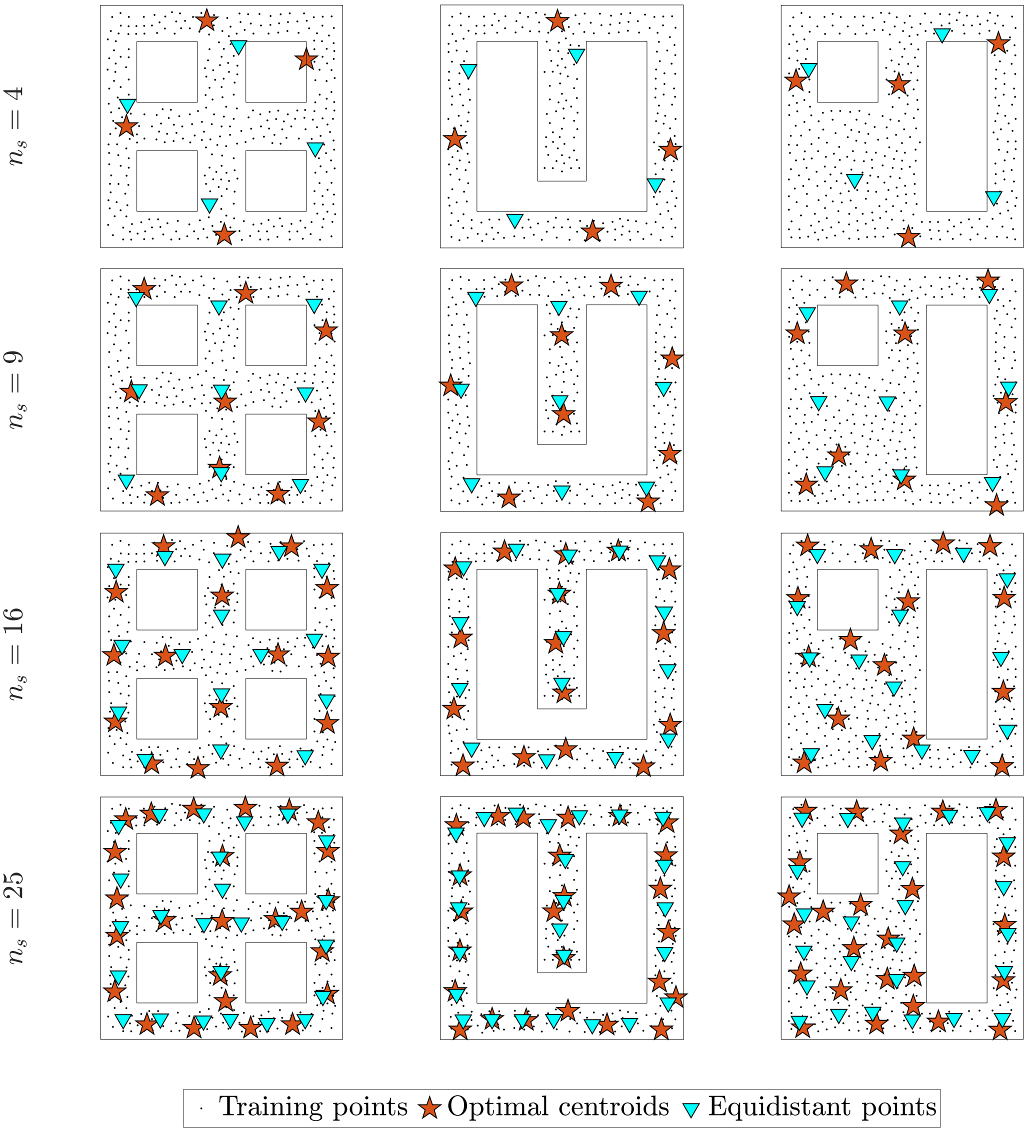


Figure 5.11 – Comparison of centroids obtained using Algorithm 4 (*red stars*) and the naive clustering, referred to as equidistant points (*cyan down-facing triangles*). Each column shows a different geometry while each row shows a fixed number n_s of inducing points, which increases from top to bottom, i.e., $n_s = 4, 9, 16, 25$.

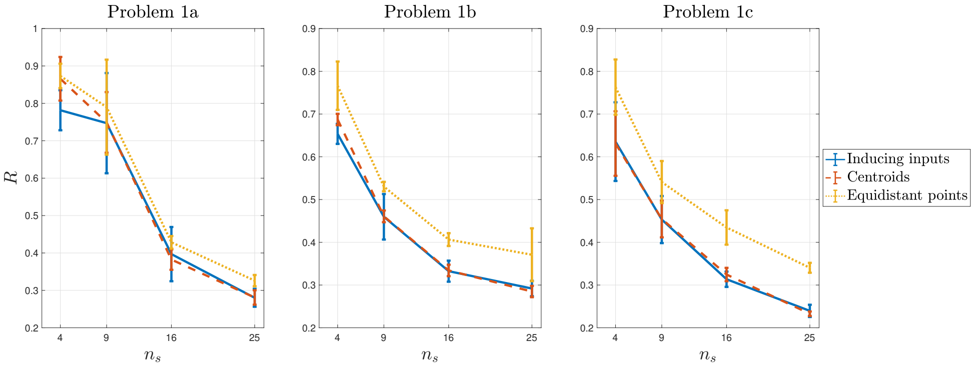


Figure 5.12 – Mean reconstruction errors with error-bars with respect to the number n_s of inducing points for the three quantities of interest jointly (*solid line*) used to train the variational sparse GP model. The corresponding mean reconstruction error, obtained by training a variational sparse GP model with fixed inducing inputs corresponding to the centroids is also shown (*dashed line*) together with the one where the fixed inducing inputs are the naive centroids (*dotted line*). Each plot shows the result for one of the three geometries.

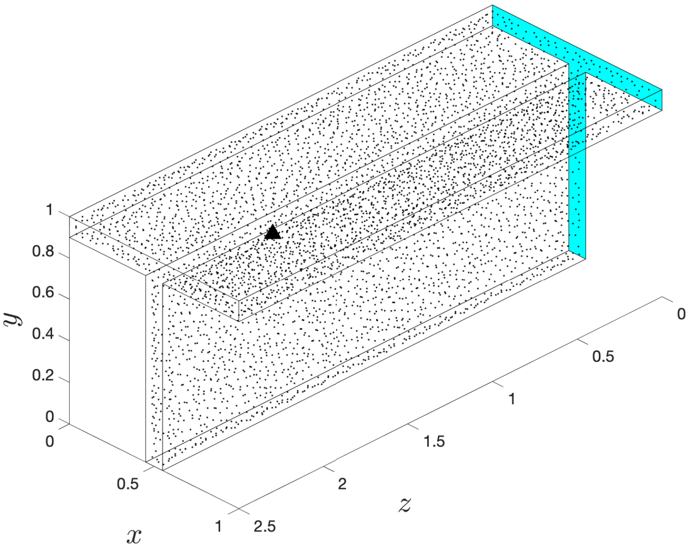


Figure 5.13 – 3D geometry of a T-beam with 4688 training points (*small black dots*), corresponding to the vertices of a coarse mesh over the domain. The location of the center of the active source corresponds to $\bar{S} = [0.7, 1, 2]$ (*black triangle*). The Dirichlet boundary corresponds to the surface at $z = 0$ (*cyan filled surface*).

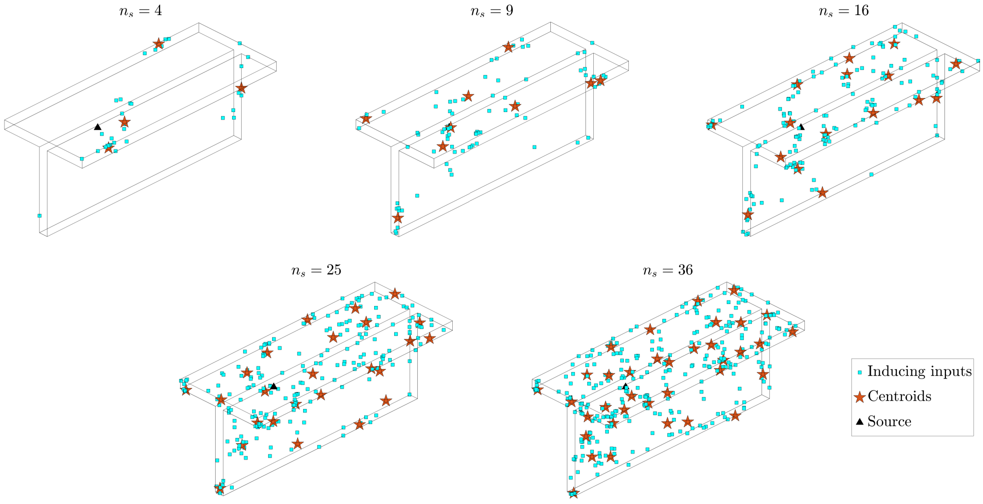


Figure 5.14 – Comparison of the location of inducing points obtained by applying Algorithm 3 for $n_\mu = 10$ samples (*cyan squares*) and the corresponding n_s centroids obtained with Algorithm 4 (*red stars*) for the 3D problem. Each plot shows a different fixed number n_s of inducing points, i.e., $n_s = 4, 9, 16, 25, 36$.

chosen as quantity of interest. Here, we choose to first identify the optimal n_s components and then place one sensor per chosen component instead of computing the exact locations of the sensors directly. We note that this is a practical procedure in real-life engineering where the exact location of a sensor on a chosen component can be decided later, both empirically through engineering experience and practicality. We assume a budget of $n_s = 10$ displacement sensors and, for each one of the 192 components, we fix a sensor location, e.g., the point near the geometric center of that component. Thus, the admissible set Ω_s is such that $|\Omega_s| = 192$. We randomly generate $n_\mu = 40$ samples in $\mathcal{P}$ and apply Algorithm 4 to get the n_s cluster centers as the components for sensor placement, as shown in Figure 5.18.

We note that although the geometry of the jacket structure is complicated, the chosen components are distributed approximately evenly over the whole domain, providing evidence that employing variational inference of sparse GPRs prevents waste of sensed information. To validate this sensor configuration, considering the complexity of the geometry and the large number of degrees of freedom, we return to the anomaly detection strategy introduced in Section 4.3. First, we place $n_s = 10$ displacement sensors on the surface of the optimal components and then train a one-class Support Vector Machine (oc-SVM) classifier for each sensor location, following the procedure presented in Chapter 4, for $N_{tr} = 100$ samples, randomly generated from $\mathcal{P}$. We observe that for real-life engineering problems, to assess the most probable damages, one may include know-how and experience of engineers. For the proposed configuration, we consider an increased wind load, i.e., $\Omega_\mu^{\text{extra}} = [1, 1.5]$ kPa, to represent a source of potential structural damages.

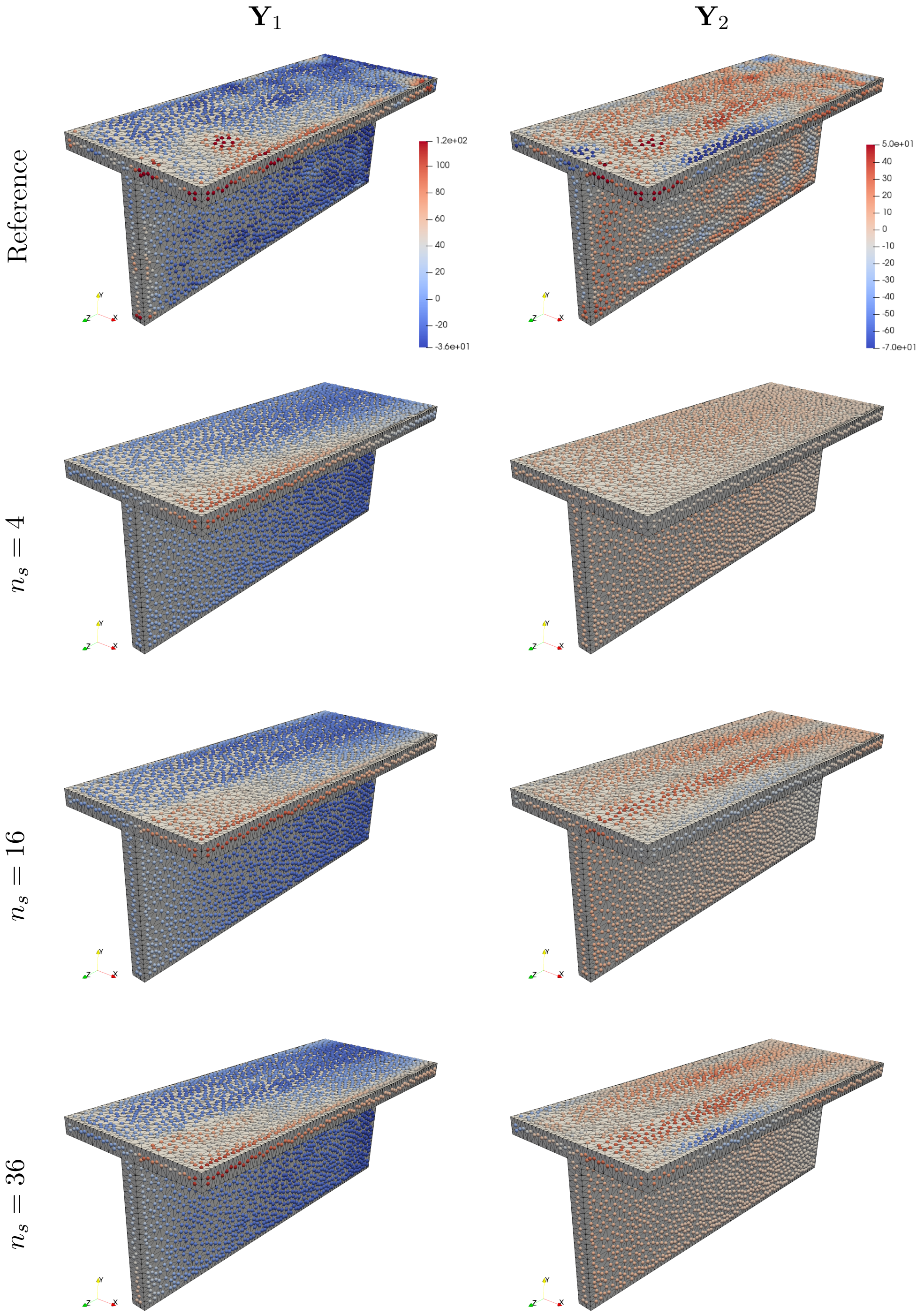


Figure 5.15 – Comparison of the first two principal components obtained either by extracting the features from the time signals and then performing PCA (*first row*) or by sparse GP reconstruction using $n_s = 4, 16$, or 36 inducing points (*second to fourth rows*) for the 3D problem. As the number of inducing points increases, the output of interests can be better reconstructed. The *reference* principal components correspond to the results obtained for $\boldsymbol{\mu}_1 = [1, 0.33, 2]$. The color scale is the same for the reference and the corresponding reconstructions.

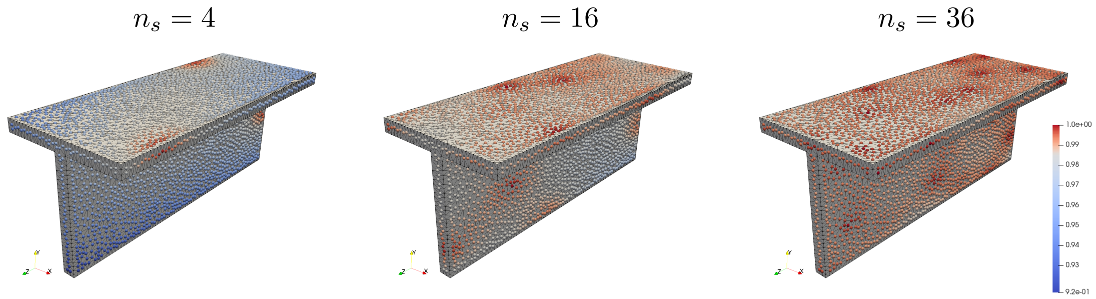


Figure 5.16 – Relative variance reduction (5.15) obtained using n_s centroids and averaged over n_μ samples for the 3D problem. Each plot shows a different fixed number n_s of inducing points, which increases from left to right, i.e. $n_s = 4, 16, 36$. The color scale is the same for the three plots.

We design four test scenarios, depending on the chosen input parameter space, i.e., either the baseline $\mathcal{P}$ or the modified $\mathcal{P}$, and for each case we sample $N_{tr} = 100$ parameters. In particular, case 1 corresponds to the healthy scenario, i.e., μ_x, $\mu_y \in \mathcal{P}$; case 2 and 3 represent scenarios of potential minor damages, i.e., we choose $\mu_x \in \mathcal{P}^{\text{extra}}$ and $\mu_y \in \mathcal{P}$ for case 2 and, the opposite, i.e., $\mu_x \in \mathcal{P}$ and $\mu_y \in \mathcal{P}^{\text{extra}}$ for case 3; lastly, for case 4, the loads in both directions are sampled from the extended parameter space, i.e., μ_x, $\mu_y \in \mathcal{P}^{\text{extra}}$. The classification results for the four test cases, sensor by sensor, are shown in Table 5.1, where the accuracy percentages in correctly classifying $n_\mu = 100$ samples per scenario are provided. Healthy samples are classified as inliers with 98% of success for all the sensor locations. The major damaged case (case 4) is always detected, i.e., for all samples and all sensor locations, the classifier correctly identifies the outliers. However, among all scenarios, we observe that for case 2 we do not obtain as accurate results as compared to other cases. We point out that the test cases are randomly generated and we notice that the false positives in cases 2 and 3 correspond to the situation in which one of the two parameters, i.e., either μ_x or μ_y, sampled from $\mathcal{P}^{\text{extra}}$, is close to the lower bound, i.e., close to the healthy domain $\mathcal{P}$, fooling the classifier. In this case, the accuracy of the classifier can be improved by enlarging the training data set. Finally, we remark that, given the general situation where various types of anomalies in different locations can appear during the life time of a structure, relying on the assumption that we only have access to the simulation data of the healthy structure allows us to present a systematic way to place a designed amount of sensors to encourage the representation of the statistics of the whole domain while preventing sensed information waste.

5.5 Conclusions

A systematic approach to the sensor placement problem in a SHM context, where no prior knowledge on the damages is assumed, is proposed. The examples presented in this

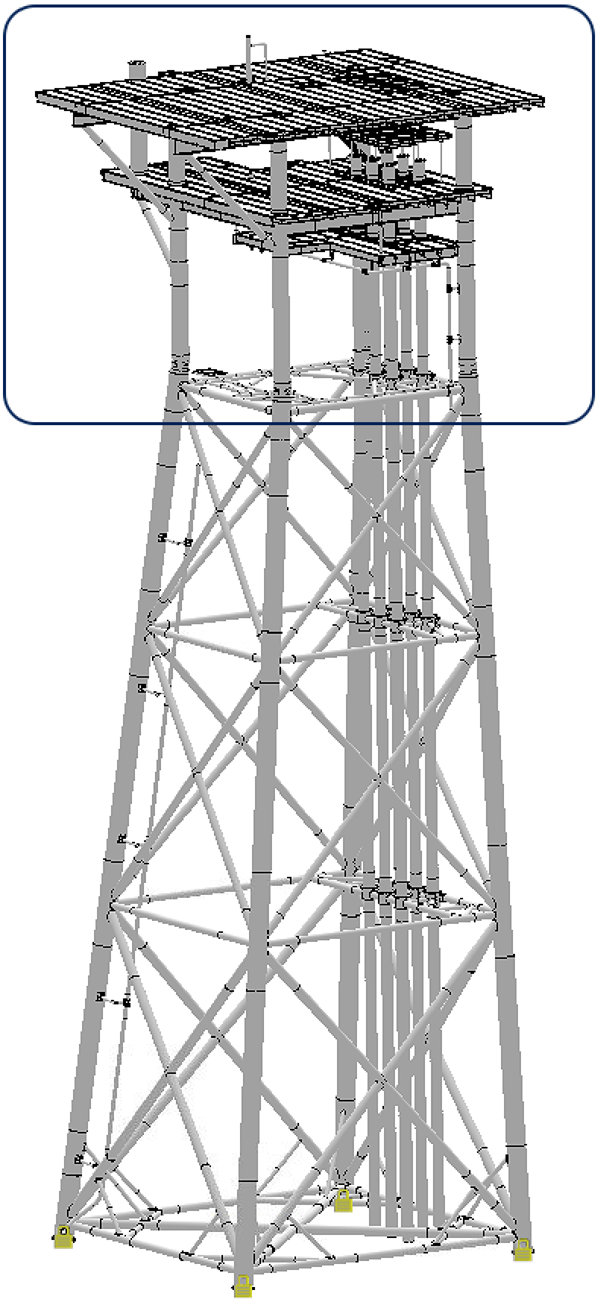


Figure 5.17 – Jacket model: wind loads applied on components in the square.

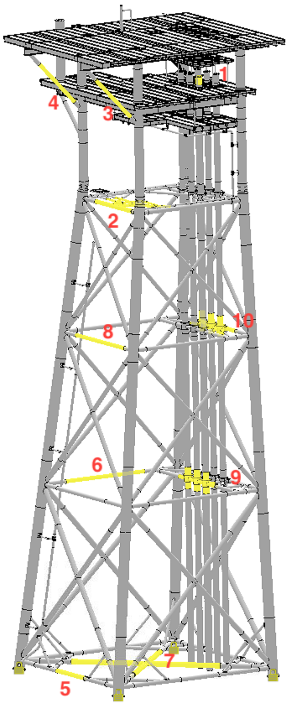


Figure 5.18 – Jacket model: components chosen for sensor placement.

Sensor number	Case 1	Case 2	Case 3	Case 4
1	99	81	93	100
2	100	83	94	100
3	99	77	94	100
4	100	77	94	100
5	98	91	92	100
6	100	81	93	100
7	100	79	90	100
8	99	74	94	100
9	100	83	93	100
10	100	76	93	100

Table 5.1 – Sensor-wise percentages of accuracy for undamaged (case 1), minor damaged (cases 2 and 3) and major damaged (case 4) scenarios.

chapter provide numerical evidence that the variational inference of sparse GPR can be modified to place the sensors on structures characterized by complex geometries. The proposed approach is validated against both 2D and 3D numerical examples to confirm the quality of the sensor placement. We note that one of the novelties of the proposed method is that it does not assume any prior information of the anomalies, hence, it is robust to different type and severity of damages. In this work, the generation of synthetic healthy databases leverages reduced order modeling techniques to efficiently include physical and geometrical parametric dependencies. As a direct consequence, the method is easily extendable to other structures and avoids the high computational costs related to simulating high fidelity models and considering all possible damage combinations. Nevertheless, we highlight that the core of the approach is independent of the method used to generate the output of interests and that any damage-sensitive quantity can be used to guide the placement of sensors in the absence of damaged scenarios.

We finally remark that in real-life engineering, the parameter space describing the natural variations of a large-scale structure is expected to be high dimensional. The procedure explained in this chapter can be extended to many parameters, but it requires a higher computational effort for both the construction of a healthy database and the training of multiple sparse variational GPR models. When the number of parameters is too large, one may rely on methodologies that compress the parameter space by retaining only those few parameters that influence the quantity of interest the most. The variance-based global sensitivity indices (Sobol's indices) [Sob01] and the derivative based global sensitivity measures (DGSM) [SK09] are popular choices. A brief introduction to these techniques and preliminary results towards this direction are provided in Appendix A.

CHAPTER 6

CONCLUSIONS

6.1 Summary and conclusions

This thesis focuses on the integration of mathematical models and observations to enable accurate predictions in the context of structural health monitoring. With the objective of estimating the state of damage of a structure of interest, we integrate methodologies, based on model order reduction (MOR) for systems modeled by parametric PDEs, with data-driven techniques. We propose a novel simulation-based approach for anomaly detection, which combines efficient construction of synthetic datasets with semi-supervised learning techniques.

In this semi-supervised context, we build a training set of sensor measurements solely composed of *healthy* labeled data and we test the obtained classifiers with new sensor measurements to evaluate if they comply with the salient features of the training set or not. This approach provides two main advantages. First, it gives the possibility to consider damage per se, without the need of pre-defining its type, location, or extent. Instead of specifying the expected damages by simulating a few examples, we prefer to construct a robust dataset that describes the baseline healthy variations of the structure in terms of its material properties, geometry, and boundary conditions. Second, the analysis of the classification responses provides a multi-level hierarchical damage identification process, giving insight not only on the presence of damage, but also on its location and severity. To achieve the same level of information with alternative supervised learning methods, many different categories should be defined a priori, which would lead to a bottleneck in terms of the number of combinations of damaged states to simulate. In addition, we exploit pattern recognition and machine learning strategies, well developed in the context of anomaly detection, e.g., to recognize novel objects in images and video streams. We explore four different strategies: the one-class Support Vector Machines, the Isolation Forests, the Local Outlier Factor and the Variational Autoencoders. We also highlight the possibility of using autoencoders to automatically extract the most important features that healthy measurements have in common, thus overcoming the need of designing engineering-based damage-sensitive features.

The absence of a priori assumptions on the anomalous data is also exploited to systematically place a network of sensors on a structure. To guide the optimal placement of sensors, we leverage in an original way sparse Gaussian processes, which are typically used to reduce the computational burden when performing regression tasks on big data. The possibility of systematically placing a fixed number of sensors on a structure of interest is of great importance in order to provide a good quality of system surveillance and to control the costs of sensor deployment. The present strategy also integrates the parametric dependency of the physical model to account for its natural variations.

The physical model is designed to illustrate a guided-wave based monitoring approach, where an active source excites the structure with a known impulse for a limited period in time. From a mathematical standpoint, the Laplace transform provides a powerful tool to connect the time and frequency domains and analyse the transient response of the system. The reduced model is built in frequency domain and, in particular, we design a stable reduced model by means of the cotangent-lift method [PM16] to solve the system associated with the Finite Element model when considering the decoupling in the real and imaginary parts. Weeks method, a numerical inverse Laplace transform strategy, provides a way to recover the time signals at a cost which is independent of the time variable, thus allowing for the generation of discrete time signals.

Finally, we validate the framework of simulation-based anomaly detection and sensor placement with 2D and 3D examples of increasingly complexity.

6.2 Outlook

In this thesis we illustrate the potential of SHM as a decision-making support tool for the implementation of efficient digital twins, which can be crucial to reduce fatalities and economical losses. However, many challenges curb the extensive use of SHM technologies. Primarily, although SHM is a very multidisciplinary topic, there are in general limited interactions between the different communities in structural engineering, sensor networks, and scientific computing. We believe that a strategy to favour a synergetic exchange of expertise towards efficient SHM technologies should be implemented by academic and industry partners at a national and international level. Furthermore, the lack of reliable real case studies often hinders the validation of new methodologies. It is worth considering the definition of a few complex test cases to be used as baseline comparison.

One of the challenges to simulation-based approaches consists in the efficient integration of real experimental data with synthetic data. An effective calibration of the input parameters should be performed to reduce the bias in the model and generate robust classifiers that accurately distinguish healthy configurations from damaged ones even when training data are synthetic and test data are experimental. For this, a Bayesian approach should be used to equip the model with a confident estimation of the sampling distribution for each parameter. These techniques should be combined with the insight gained by performing sensitivity analysis to select the input variables which most influence the output of interest and thus reduce the computational burden associated with large input parameters typical of real-life engineering examples. The availability of real experimental data for model calibration could be further exploited in the construction of the training database and in the tuning of the threshold to distinguish healthy scenarios from damaged ones. We conjecture that constructing a dataset by including a multi-fidelity approach, where real data are combined with weighted synthetic data, would result in a more robust classifier.

Moreover, in this work, a real-valued anomaly score is computed for each new measurement by means of one-class machine learning algorithms. It would be interesting to consider a probabilistic approach to quantify how the uncertainties on the input parameters and the model propagate through the various steps of the SHM system. This is essential to estimate the level of confidence associated with the predictions on the state of damage and therefore better guide the final decisions. Recent attempts on the integration of uncertainty quantification with engineering design and monitoring can be found in [LML$^+$17; BdSG12; JLS15; AP05; ZZCA13].

We also highlight that the problems in this thesis are limited to the linear case. The extension to more complex models, including nonlinearities should be developed to provide more realistic simulated data. Efforts in this direction can be found for example in [SDC20], where the authors develop a reduced order modeling method based on data from nonlinear structural systems to make predictions of time series under arbitrary dynamic forcing. Other examples on non-linear structural analysis combined with reduced modeling can be found in [GH18; ZGH19].

Finally, we note that the proposed method allows to estimate the state of damage at a fixed point in time. However, a full monitoring system should also be able to provide a way to follow the growth of a degradation process, e.g., the expansion of cracks or the advancement of corrosion. In this sense, implementing a data assimilation framework with the repeated integration of model updates, would help to provide a dynamic assessment.

APPENDIX A

SENSITIVITY ANALYSIS AND PARAMETER ESTIMATION

In this thesis we consider the case in which the range and probability distribution of the input parameter is known. In particular, we focus on the approximation of sensor measurements given a parameter vector $\boldsymbol{\mu} \in \mathcal{P}$, characterizing the natural variations of the structure of interest. In particular, we focus on the input-output mapping

$$\boldsymbol{\mu} \to \boldsymbol{u}_h(\boldsymbol{x}, t; \boldsymbol{\mu}) \to \hat{\mathbf{g}}_i(\boldsymbol{\mu}),$$

where $\boldsymbol{u}_h(\boldsymbol{x}, t; \boldsymbol{\mu})$ is the high-fidelity numerical approximation of the displacement, i.e., the solution of (3.8) and $\hat{\mathbf{g}}_i(\boldsymbol{\mu})$ are the output of interests evaluated at the sensor locations $\boldsymbol{x}_i \in \Omega$ for $i = 1, \dots, n_s$. However, a suitable model calibration is necessary to reduce the model errors which may prevent correct classification results especially when the training set is composed of synthetic data and experimental measurements are evaluated during the test phase. While parameter estimation goes beyond the objectives of this thesis, developing an efficient and accurate mathematical framework for the estimation of the model parameters is crucial to develop a correct structural health monitoring (SHM) system. Recently, several attempts have been made to include uncertainty quantification in the SHM technologies, see, e.g., [ZXSO16], where a fast Bayesian method is developed to perform modal identification of a tall building, or [LML$^+$17], where a dynamic Bayesian network is used to track the evolution of variables required to monitor the wing of an aircraft.

Parameter estimation

Parameter estimation and model calibration techniques based on Bayesian inference have been developed in the last decades. We refer to [Stu10; KS06] for detailed descriptions of inverse problems and to [RC15; LSZ15] for a probabilistic overview on data assimilation.

In this appendix we limit ourselves to the introduction of the principal concepts needed to formulate the Bayesian inverse problem and we present a preliminary result on the estimation of a few parameters describing the speed function of the damped wave equation, which can be found in [RBH18].

The goal of inverse problems is to estimate the unknown parameters $\boldsymbol{\mu}$ from noisy observations, which in our context correspond to time-dependent sensor measurements:

$$\mathbf{g}_i^{\text{exp}} = \mathcal{G}_i(\boldsymbol{\mu}) + \varepsilon_{\text{noise}}, \quad \varepsilon_{\text{noise}} \sim \pi_{\text{noise}} := \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \tag{A.1}$$

where the operator $\mathcal{G}_i$ maps the input parameter $\boldsymbol{\mu}$ to the output of interest $\hat{\mathbf{g}}_i(\boldsymbol{\mu})$. We consider here an additive Gaussian model, which mimics the possible effect of many sources of error. However, other possibilities could be considered, e.g., different distributions for the additive noise or multiplicative noise. We additionally assume that some *prior* information on the model parameters is available and we consider the particular case in which the prior distribution is Gaussian, i.e., $\boldsymbol{\mu} \sim \pi_{\text{prior}}(\boldsymbol{\mu}) := \mathcal{N}(\boldsymbol{\mu}_{\text{prior}}, \mathbf{C}_{\text{prior}})$. With a probabilistic approach, the solution of the inverse problem (A.1) is the *posterior* distribution, i.e., the probability distribution of $\boldsymbol{\mu}$ given $\mathbf{g}_i^{\text{exp}}$, which can be estimated thanks to *Bayes' theorem*

$$\pi_{\text{post}}(\boldsymbol{\mu}|\mathbf{g}_i^{\text{exp}}) = \frac{\pi(\mathbf{g}_i^{\text{exp}}|\boldsymbol{\mu})\pi_{\text{prior}}(\boldsymbol{\mu})}{Z},$$

where $\pi(\mathbf{g}_i^{\text{exp}}|\boldsymbol{\mu}) = \pi_{\text{noise}}(\mathbf{g}_i^{\text{exp}} - \mathcal{G}_i(\boldsymbol{\mu}))$ is the *likelihood* and Z is a normalization constant.

In general, the posterior distribution cannot be written in closed form. Instead, it is necessary to sample the probability density function by means of random sampling methods such as the Markov chain Monte Carlo (MCMC) [Gil05] to enable an exploration of the probability distribution. The results presented in [RBH18] focus on the well-known Metropolis-Hasting algorithm [Has70] and consider additional methods such as the Robust Adaptive Metropolis (RAM) [Vih12] and the preconditioned Crank-Nicolson (pCN) algorithm [CRSW13]. The former is introduced to overcome the slow convergence of the Metropolis-Hasting algorithm and guide the choice of the value of the step size used in the random walk, and the latter presents an efficient sampling method for high-dimensional parameter spaces.

Let us conclude by presenting an example where the goal is to estimate the parameters carachterizing the speed $c(x)$ of the dumped wave equation in 2D. We provide the principal details and refer to [RBH18] for a contextualization of the problem. Let $\Omega := [0, 1]^2$ and $T \in \mathbb{R}_+$, the damped wave equation, equipped with homogeneous boundary and initial conditions, is given as

$$\frac{\partial^2 \boldsymbol{u}}{\partial t^2} + \eta \frac{\partial \boldsymbol{u}}{\partial t} - \nabla \cdot (c^2(\boldsymbol{x}; \boldsymbol{\mu}) \nabla \boldsymbol{u}) = h(t) f(\boldsymbol{x}), \quad \text{in } \Omega \times (0, T]. \tag{A.2}$$

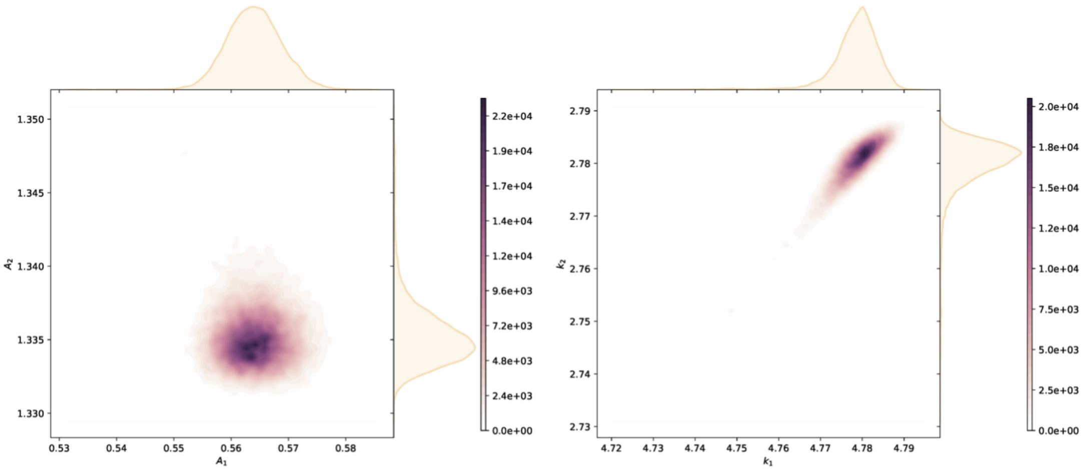


Figure A.1 – Joint posterior density approximations of the parameters characterizing the wave speed. A_1 vs A_2 (*left*) and k_1 vs k_2 (*right*).

Here, $\eta := 0.1$ is the damping coefficient and h and s are the source functions. The space-dependent source function s is defined in (4.20) with $\bar{S} := [0.85, 0.5]$, while h is periodic in time, defined as $h(t) = \sin(\omega t)$. We assume that $c(\boldsymbol{x}; \boldsymbol{\mu})$ has the following expression and it is defined by $d_\mu := 4$ parameters, i.e., $\boldsymbol{\mu} := [A_1, A_2, k_1, k_2]$:

$$c(\boldsymbol{x}; \boldsymbol{\mu}) := \sqrt{0.1 + (A_1 \cos(2\pi k_1 \|\boldsymbol{x}\|) + A_2 \sin(2\pi k_2 \|\boldsymbol{x}\|))^2}.$$

Moreover, let $\pi_{\text{noise}} = \mathcal{N}(0, \sigma^2_{\text{noise}})$ with $\sigma_{\text{noise}} = 0.025$ and $\pi_{\text{prior}} = \mathcal{N}(\boldsymbol{\mu}_{\text{pr}}, \sigma^2_{\text{pr}} \mathbf{I}_2)$, with $\boldsymbol{\mu}_{\text{pr}} := [0.7, 1.5, 5, 3]$, and $\sigma_{\text{pr}} := 0.1$. The observations of the wave speed c correspond to $n_{\text{obs}} = 4$ observations retrieved at $n_s = 9$ sensors located at $\boldsymbol{x} := [x_i, y_j]$ for all the combinations i, j, where $x_i, y_j = [0.25, 0.5, 0.75]$. The observations are obtained numerically by solving (A.2) with a Fourier solver. Finally, the joint posterior distributions for the parameters A_1 and A_2, and the parameters k_1 and k_2, obtained with the aforementioned Bayesian approach and RAM algorithm, are shown in Figure A.1.

Sensitivity analysis

In this appendix, we also present the preliminary results obtained via sensitivity analysis. Sensitivity analysis is a key tool to identify a subset of parameters which most influence a fixed output of interest by associating sensitivity indices to rank the importance of the parameters. These methodologies are particularly important when dealing with real-life engineering systems, which are typically defined by a large number of parameters. Dealing with too many parameters becomes impractical when applying sampling strategies such as the MCMC for parameter estimation. Sensitivity analysis, combined with efficient sampling method such as the pCN and reduced order models strategies, provides a way to overcome the computational bottleneck associated with the exploration of high-dimensional parameter distributions. Different sensitivity indices have been used in the field of uncertainty quantification to rank the input parameters and ultimately the parameter space. The variance based global sensitivity indices (Sobol' indices) [Sob01; STCR04; SRA^{+}08] and the derivative based global sensitivity measures (DGSM) [SK09] are two widely used measures. The former is computationally demanding since a large number of model evaluations are needed to acquire an acceptable convergence. In this work we rely on the DGSM to compress the high dimensional parameter space while retaining a few parameters that entail significant uncertainty with respect to an output of interest. As observed in [SK09], the computational time required for numerical evaluation of DGSM can be much lower than that for estimation of the Sobol' indices although it is problem dependent. In the remainder we briefly introduce the DGSM setting and present a preliminary result.

Let f be a differentiable function and $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{d_\mu}]$ the d_μ-dimensional input. The d_μ local sensitivity measures are are based on partial derivatives, i.e.,

$$v_j := \mathbb{E}\left[\left(\frac{\partial f(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}_j}\right)^2\right].$$

The derivatives are estimated by computing the so-called elementary effect, i.e., the incremental ratio of the j-th input factor

$$EE_j := \frac{f\left([\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{j-1}, \boldsymbol{\mu}_j + \Delta\boldsymbol{\mu}_j, \boldsymbol{\mu}_{j+1}, \dots, \boldsymbol{\mu}_{d_\mu}]\right) - f(\boldsymbol{\mu})}{\Delta\boldsymbol{\mu}_j}.$$

Although these quantities are usually estimated using expensive sampling methods such as Monte Carlo (MC) or quasi-Monte Carlo (QMC), they require fewer model evaluations than the variance-based methods. A connection between DGSM indices v_j and Sobol' indices is given in [SK09], where it is also shown that small DGSMs yield small total sensitivity indices. Hence, parameters with small DGSM are expected to be less important in explaining the output of interest. From a practical perspective, if v_j is almost zero, $\boldsymbol{\mu}_j$ can be fixed to any value in its range without influencing the value of the output of interest.

We consider here a preliminary 2D example with the same healthy geometry and 15 fixed sensor locations as shown in Figure 4.3. Different from the numerical examples in Chapters 4 and 5, in this appendix we take a larger number of input parameters. For this we introduce a conforming domain partition of Ω, i.e.,

$$\Omega := \bigcup_{j=1}^{d_\Omega} \Omega_j,$$

where Ω_i are mutually non-overlapping open subdomains such that $\Omega_i \cap \Omega_j = \emptyset, 1 \leq i < j \leq d_\Omega$. Figures A.2 and A.3 show two partition examples

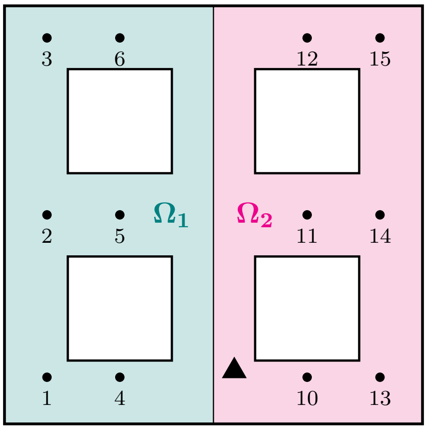


Figure A.2 – 2D geometry with $d_\Omega = 2$ subdomains. Ω_1 and Ω_2 are the left (*green*) and right (*pink*) subdomain, respectively. The location of the center of the active source corresponds to $\bar{S} = [0.54, 0.125]$ (*black triangle*).

with $d_\Omega = 2$ and $d_\Omega = 21$, respectively. We assume that each one of the d_Ω subdomains is characterized by a local value of the parameters representing the material properties, i.e., the Young modulus E and the Poisson ratio ν, which are uniformly sampled from the same parameter range[1]. In particular

$$E_j \sim \mathcal{U}\left([0.95, 1.05]\right), \quad \nu_j \sim \mathcal{U}\left([0.31, 0.35]\right), \quad \text{for } j = 1, \ldots, d_\Omega. \tag{A.3}$$

The parameter characterizing the number of cycles before attenuation of the source impulse is independent of the domain partition, i.e., $k \sim \mathcal{U}\left([1.75, 2.25]\right)$. Hence, for the examples Figures A.2 and A.3, we consider a total of $d_\mu = 5$ and $d_\mu = 43$ parameters, respectively.

This choice provides a simple way to artificially increase the number of parameters. We conjecture that each sensor signal will be mostly influenced by the subset of the parameters which define the material properties in a neighborhood of the sensor location. This implies that some sensor signals will be influenced by a subset of the parameters and other sensor signals by another subset. However, without a global output of interest and with local parameters, it will be difficult to select a common subset of parameters. Unless for some applications only very few sensors are relevant, more realistic scenarios where the parameters have a global effect should be designed.

In this context, we follow [ZGH19] and consider a modified version for the DGSMs

$$\hat{v}_j := \frac{v_j}{\sum_{j=1}^{d_\mu} v_j}, \quad \text{for } j = 1, \ldots, d_\mu, \tag{A.4}$$

---

[1] In this appendix we consider a larger interval for each parameter as compared to the parameter space (4.18), which is used for all the other simulations in this thesis.

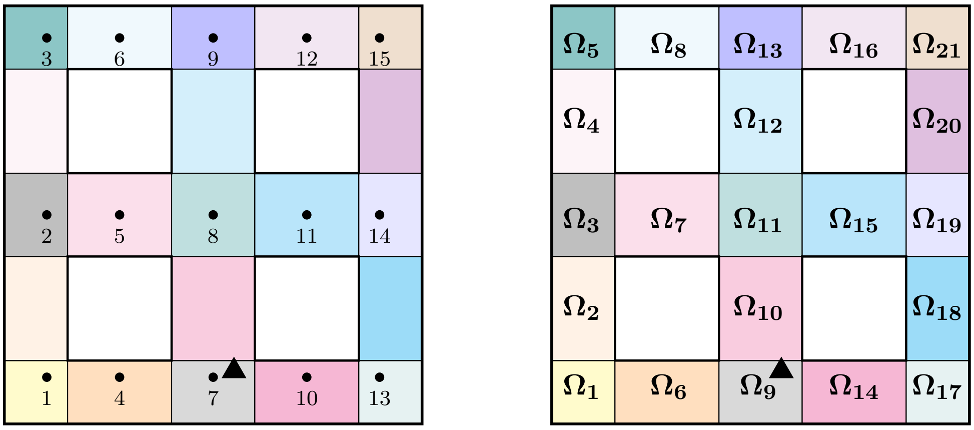


Figure A.3 – 2D geometry with $d_\Omega = 21$ subdomains equipped with 15 sensor locations (*left*). The numbering of the 21 subdomains is also shown (*right*). The location of the center of the active source corresponds to $\bar{S} = [0.54, 0.125]$ (*black triangle*).

where

$$v_j := \frac{1}{N}\sum_{m=1}^{N}\left(\frac{\hat{\mathbf{g}}_i([\boldsymbol{\mu}_1^m, \dots, \boldsymbol{\mu}_{j-1}^m, \boldsymbol{\mu}_j^m + \Delta\boldsymbol{\mu}_j, \boldsymbol{\mu}_{j+1}^m, \dots, \boldsymbol{\mu}_{d_\mu}^m]) - \hat{\mathbf{g}}_i([\boldsymbol{\mu}_1^m, \dots, \boldsymbol{\mu}_{d_\mu}^m])}{\Delta\boldsymbol{\mu}_j}\right)^2. \tag{A.5}$$

Here, the output of interests $\hat{\mathbf{g}}_i(\boldsymbol{\mu}^m)$ are the sensor signals defined in (3.30) and generated following the details provided in Chapters 3 and 4. The methodology is the same except for the affine expansion (3.47), for which additional affine parameters, specific to each subdomain, have to be introduced. In (A.5), we choose $\Delta\boldsymbol{\mu}_j$ to be equal to $\frac{1}{20}$ of the length of the corresponding parameter interval, defined in (A.3). For the case with $d_\mu = 2$, we have $\boldsymbol{\mu}^m := [E_1^m, \nu_1^m, E_2^m, \nu_2^m, k^m]$ for $m = 1, \dots, n_\mu$. We observe that a total of $n_\mu(d_\mu + 1)$ model estimations are required to compute (A.4).

For the problem with $d_\Omega = 2$, the sensitivity indices (A.4), computed using the procedure defined above for the first component of $\hat{\mathbf{g}}_i(\boldsymbol{\mu})$ for $i = 1, \dots, 6, 10, \dots, 15$ are shown in Figure A.4. For a practical reason, for the example with 2 subdomains, we choose to discard the signals retrieved at sensors 7, 8 and 9, which would otherwise lie at the interface between Ω_1 and Ω_2. Each plot shows the sensitivity indices associated with the $d_\mu = 5$ parameters and obtained by averaging over $n_\mu = 30$ random samples. We note that the order of the plots follow the position of the sensors used in Figure A.2. We first observe that for all the sensor locations, the parameter characterizing the active source is the least important. Moreover, the midline sensors, i.e., sensors 2, 5, 11, 13, seem to be more influenced by their neighboring material properties. Indeed, for the signal retrieved at sensors 2 and 5, the parameters E_1 and ν_1 have the highest index value, and, similarly, for the signal retrieved at sensors 11 and 13, E_2 and ν_2 have the highest DGSM value. Figure A.5 shows the effect of the different input parameters on the output of interest at sensors 5 and 11. We observe that the sensor signals at location 5 variate more

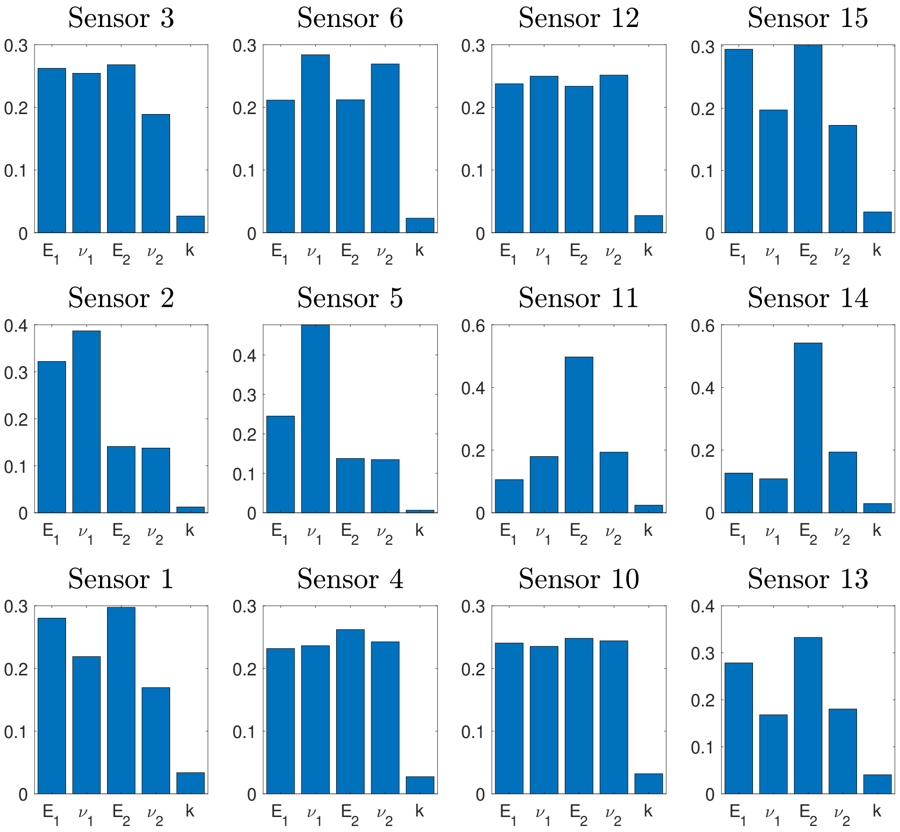


Figure A.4 – DGSM indices with respect to $d_\mu = 5$ parameters for the first component of the sensor signals $\hat{\mathbf{g}}_i(\boldsymbol{\mu})$ for $i = 1, \dots, 6, 10, \dots, 15$.

with respect to E_1 and ν_1. Similarly, the sensor signals at location 11 variate more with respect to E_2 and ν_2. Both signals variate quite consistently with the source parameter.

Figure A.6 shows similar results for the case with $d_\Omega = 21$ subdomains, where the sensitivity indices (A.4), computed for the first component of $\hat{\mathbf{g}}_i(\boldsymbol{\mu})$ for the sensor signals at $\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_{10}, \boldsymbol{x}_{11}, \boldsymbol{x}_{12}$, highlight that the output of interest is mostly influenced by its neighboring parameters. Each plot shows the sensitivity indices associated with the $d_\mu = 43$ parameters and obtained by averaging over $n_\mu = 20$. The signal retrieved at sensor 5 is mostly influenced by ν_3 and ν_{15}, sensor 6 by ν_5, sensor 11 by ν_{15} and E_{17}, and sensor 12 by ν_5 and ν_{17}. Sensors 4 and 10 are mostly influenced by the material properties characterizing Ω_1, Ω_9, and Ω_{10}, and Ω_{14}, and Ω_{17}, which correspond to the subdomains at the bottom of the structure. Finally, the role of k, the parameter characterizing the active source, seems to have a little influence on the sensors on the midline, while it is one of the parameters with higher DGSM value for sensors $4, 6, 10$, and 12.

The preliminary results presented in this appendix, despite not being completely conclusive,

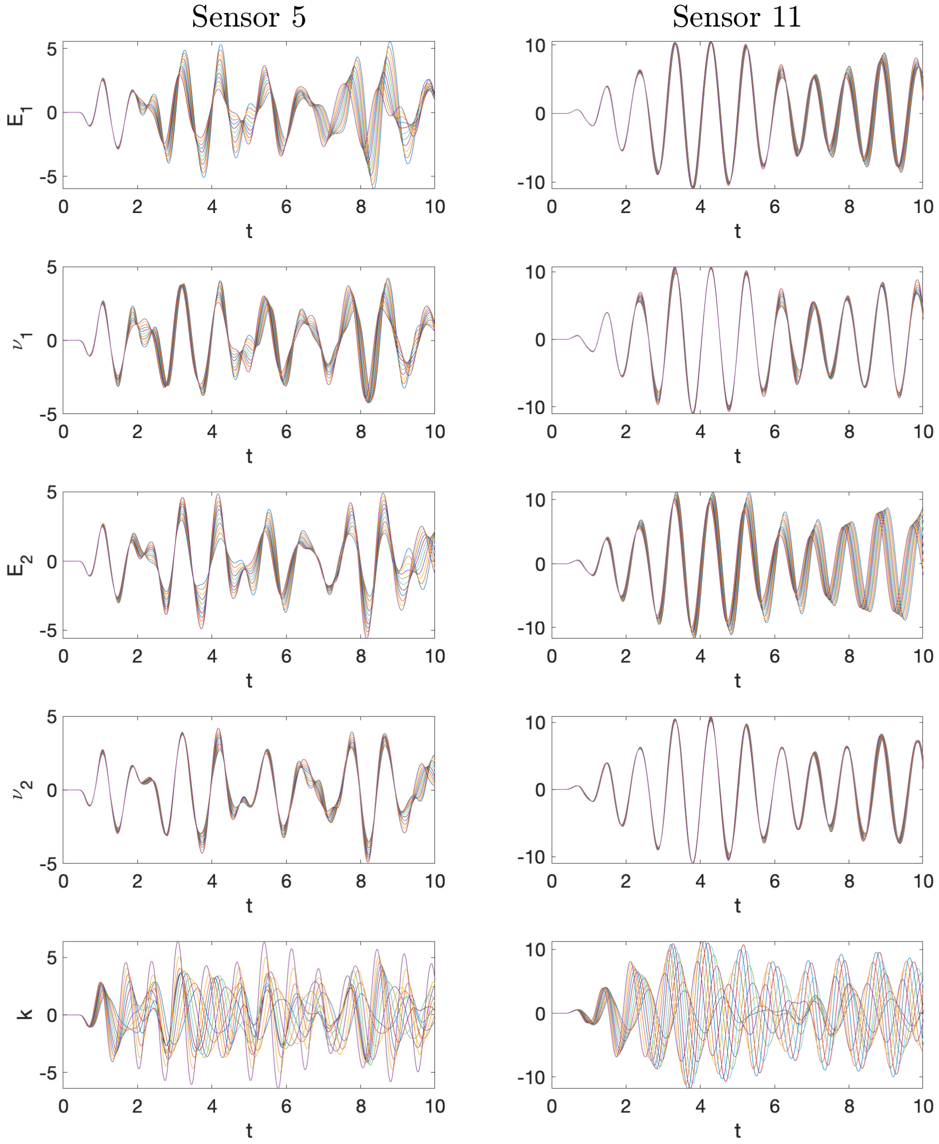


Figure A.5 – Parametric dependency of the signals at sensors 5 and 11 with respect to different values the $d_\mu = 5$ the input parameters. Sensors 5 and 11 belong to the subdomains Ω_1 and Ω_2, respectively. The first two rows show the time-dependent signals for different values of the material properties characterizing Ω_1, i.e., E_1 and ν_1. The third and fourth rows show the time-dependent signals for different values of the material properties characterizing Ω_2, i.e., E_2 and ν_2. The last row shows the time-dependent signals for different values of the parameter characterizing the active source, i.e., k.

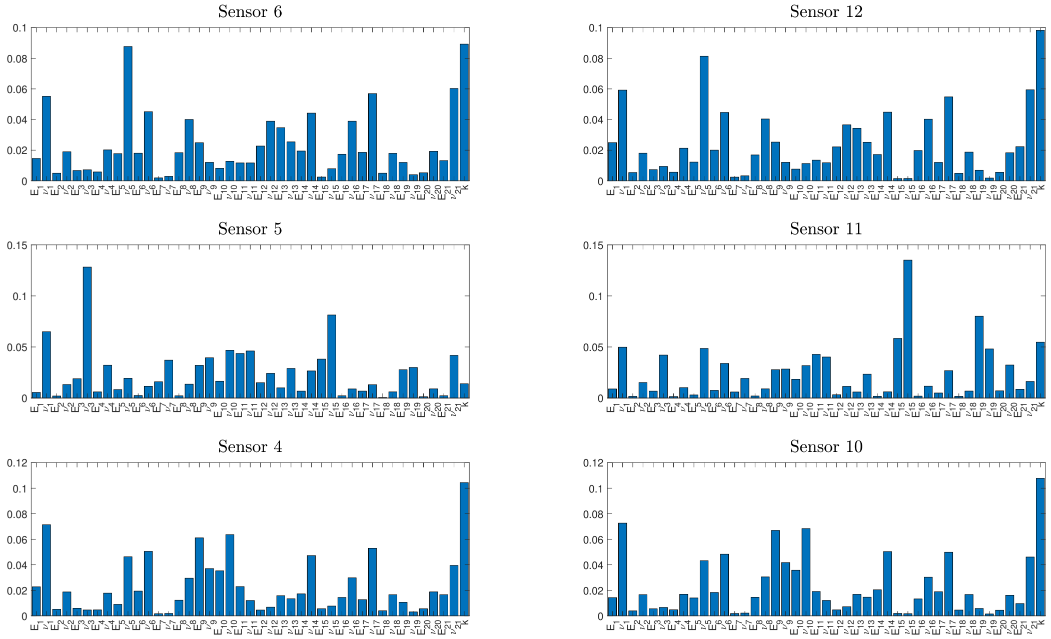


Figure A.6 – DGSM indices with respect to $d_\mu = 43$ parameters for the first component of the sensor signals $\hat{\mathbf{g}}_i(\boldsymbol{\mu})$ for $i = 4, 5, 6, 10, 11, 12$.

provide a starting point to overcome the challenges associated with the task of combining uncertainty quantification and structural health monitoring techniques.

BIBLIOGRAPHY

[ABC19] Konstantinos Agathos, Stephane Bordas, and Eleni Chatzi. Parametrized reduced order modeling for cracked solids. *International Journal for Numerical Methods in Engineering*, 2019.

[ABDC$^+$18] Jean-Philippe Argaud, Bertrand Bouriquet, F. De Caso, Helin Gong, Yvon Maday, and Olga Mula. Sensor placement in nuclear reactors based on the Generalized Empirical Interpolation Method. *Journal of Computational Physics*, 363:354–370, 2018.

[ABH$^+$15] Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

[ABY$^+$16] Guillaume Alain, Yoshua Bengio, Li Yao, Jason Yosinski, Eric Thibodeau-Laufer, Saizheng Zhang, and Pascal Vincent. GSNs: generative stochastic networks. *Information and Inference: A Journal of the IMA*, 5:210–249, 2016.

[Ach03] Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.

[ACZP18] Costas Argyris, Sharmistha Chowdhury, Volkmar Zabel, and Costas Papadimitriou. Bayesian optimal sensor placement for crack identification in structures using strain measurements. *Structural Control and Health Monitoring*, 25(5):e2137, 2018.

[AGA13] Mennatallah Amer, Markus Goldstein, and Slim 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.

[AKM+17] Ali Anaissi, Nguyen Lu Dang Khoa, Samir Mustapha, Mehrisadat Makki Alamdari, Ali Braytee, Yang Wang, and Fang 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.

[AP05] Jason Matthew Aughenbaugh and Christiaan J. J. Paredis. The Value of Using Imprecise Probabilities in Engineering Design. *Journal of Mechanical Design*, 128(4):969–979, 12 2005.

[AV16] Preeti Arora and Shipra Varshney. Analysis of K-means and K-medoids algorithm for big data. *Procedia Computer Science*, 78:507–512, 2016.

[AVCT20] Luis David Avendaño-Valencia, Eleni N. Chatzi, and Dmitri Tcherniak. Gaussian process models for mitigation of operational variability in the structural health monitoring of wind turbines. *Mechanical Systems and Signal Processing*, 142:106686, 2020.

[AW06] Joseph Abate and Ward Whitt. A unified framework for numerically inverting Laplace transforms. *INFORMS Journal on Computing*, 18(4):408–421, 2006.

[BBPK16] Bingni W. Brunton, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Sparse sensor placement optimization for classification. *SIAM Journal on Applied Mathematics*, 76(5):2099–2122, 2016.

[BdSG12] André T. Beck and Wellison J. de Santana Gomes. A comparison of deterministic, reliability-based and risk-based structural optimization under uncertainty. *Probabilistic Engineering Mechanics*, 28:18–29, 2012.

[BGW15] Peter Benner, Serkan Gugercin, and Karen Willcox. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM review*, 57(4):483–531, 2015.

[BH20] Caterina Bigoni and Jan S. Hesthaven. Simulation-based anomaly detection and damage localization: An application to structural health monitoring. *Computer Methods In Applied Mechanics And Engineering*, 363, 2020.

[Bis06] Christopher M. Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

[BKNS00] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: identifying density-based local outliers. In *ACM sigmod record*, pages 93–104, 2000.

[BMS04] Franco Brezzi, L Donatella Marini, and Endre Süli. Discontinuous galerkin methods for first-order hyperbolic problems. *Mathematical models and methods in applied sciences*, 14(12):1893–1903, 2004.

[BSR16] Francesco Ballarin, Alberto Sartori, and Gianluigi Rozza. RBniCS - reduced order modelling in FEniCS. http://mathlab.sissa.it/rbnics, 2016.

[BWW20] Prabhav Borate, Gang Wang, and Yi Wang. Data-driven structural health monitoring approach using guided Lamb wave responses. *Journal of Aerospace Engineering*, 33(4):04020033, 2020.

[BZH20] Caterina Bigoni, Zhenying Zhang, and Jan S. Hesthaven. Systematic sensor placement for structural anomaly detection in the absence of damaged states. *Computer Methods in Applied Mechanics and Engineering*, 371:113315, 2020.

[C$^+$15] François Chollet et al. Keras. https://keras.io, 2015.

[CABK18] Emily Clark, Travis Askham, Steven L. Brunton, and J. Nathan Kutz. Greedy sensor placement with cost constraints. *IEEE Sensors Journal*, 19(7):2642–2656, 2018.

[CC19] Youngsoo Choi and Kevin Carlberg. Space-time least-squares Petrov-Galerkin projection for nonlinear model reduction. *SIAM Journal on Scientific Computing*, 41(7):A26–A58, 2019.

[CCMA17] Giovanni Capellari, Eleni Chatzi, Stefano Mariani, and Saeed Eftekhar Azam. Optimal design of sensor networks for damage detection. *Procedia engineering*, 199:1864–1869, 2017.

[CD06] Alan K. Cline and Inderjit S. Dhillon. *Computation of the singular value decomposition*. Citeseer, 2006.

[CDdMU17] Giovanni Castellazzi, Antonio Maria D'Altri, Stefano de Miranda, and Francesco Ubertini. An innovative numerical modeling strategy for the structural analysis of historical monumental buildings. *Engineering Structures*, 132:229–248, 2017.

[CFL28] Richard Courant, Kurt Friedrichs, and Hans Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische annalen*, 100(1):32–74, 1928.

[CJK$^+$19] Kevin T. Carlberg, Antony Jameson, Mykel J. Kochenderfer, Jeremy Morton, Liqian Peng, and Freddie D. Witherden. Recovering missing CFD data for high-order discretizations using deep neural networks and dynamics learning. *Journal of Computational Physics*, 395, 2019.

[Cle55] Charles William Clenshaw. A note on the summation of Chebyshev series. *Mathematics of Computation*, 9(51):118–120, 1955.

[CO02] Lehel Csató and Manfred Opper. Sparse on-line Gaussian processes. *Neural computation*, 14(3):641–668, 2002.

[Coh07] Alan M. Cohen. *Numerical methods for Laplace transform inversion*, volume 5. Springer Science & Business Media, 2007.

[Cre91] Noel A. C. Cressie. *Statistics for spatial data.* A Wiley-interscience publication. J. Wiley, 1991.

[CRSW13] Simon L Cotter, Gareth O. Roberts, Andrew M. Stuart, and David White. MCMC methods for functions: modifying old algorithms to make them faster. *Statistical Science*, pages 424–446, 2013.

[Cru76] Kenny S. Crump. Numerical inversion of Laplace transforms using a Fourier series approximation. *Journal of the ACM (JACM)*, 23(1):89–96, 1976.

[CS02] Nello Cristianini and Bernhard Scholkopf. Support Vector Machines and Kernel Methods: The New Generation of Learning Machines. *Ai Magazine*, 23(3):31, 2002.

[CW18] Zexun Chen and Bo Wang. How priors of initial hyperparameters affect Gaussian process regression models. *Neurocomputing*, 275:1702–1710, 2018.

[Dav91] Lawrence Davis. *Handbook of genetic algorithms.* CumInCAD, 1991.

[DFPS96] S. W. Doebling, C. R. Farrar, M. B. Prime, and D. W. 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.

[DLBM14] Xuemei Ding, Yuhua Li, Ammar Belatreche, and Liam P. Maguire. An experimental evaluation of novelty detection methods. *Neurocomputing*, 135:313–327, 2014.

[DPW14] Wolfgang Dahmen, Christian Plesken, and Gerrit Welper. Double greedy algorithms: Reduced basis methods for transport dominated problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(3):623–663, 2014.

[DS18] Niccolò Dal Santo. *Multi space reduced basis preconditioners for parametrized partial differential equations.* PhD thesis, EPFL, 2018.

[DSC07] Santanu Das, Ashok N Srivastava, and Aditi Chattopadhyay. Classification of damage signatures in composite plates using one-class SVMs. In *Aerospace Conference, 2007 IEEE*, pages 1–19. IEEE, 2007.

[Duf93] Dean 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.

[Dur74] F. Durbin. Numerical inversion of Laplace transforms: an efficient improvement to Dubner and Abate's method. *The Computer Journal*, 17(4):371–376, 1974.

[EG13] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of Finite Elements*, volume 159. Springer Science & Business Media, 2013.

[FRG$^+$12] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13(Jul):2171–2175, 2012.

[FT10] Eric B. Flynn and Michael D. Todd. A Bayesian approach to optimal sensor placement for Structural Health Monitoring with application to active sensing. *Mechanical Systems and Signal Processing*, 24(4):891–903, 2010.

[FW12] Charles R. Farrar and Keith Worden. *Structural Health Monitoring: a machine learning perspective.* John Wiley & Sons, 2012.

[GB01] Ajith H. Gunatilaka and Brian A. Baertlein. Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):577–589, 2001.

[GB18] Francisco J. Gonzalez and Maciej Balajewicz. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. *arXiv preprint arXiv:1808.01346*, 2018.

[GBCB16] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[GGLM88] Burton S. Garbow, Giulio Giunta, James N. Lyness, and Almerico Murli. Software for an implementation of Weeks' method for the inverse Laplace transform. *ACM Transactions on Mathematical Software (TOMS)*, 14(2):163–170, 1988.

[GH01] David Gottlieb and Jan S. Hesthaven. Spectral methods for hyperbolic problems. *Journal of Computational and Applied Mathematics*, 128(1-2):83–131, 2001.

[GH18] Mengwu Guo and Jan S. Hesthaven. Reduced order modeling for nonlinear structural analysis using Gaussian process regression. *Computer Methods in Applied Mechanics and Engineering*, 341:807–826, 2018.

[Gil05] Walter R. Gilks. Markov chain Monte Carlo. *Wiley Online Library*, 4, 2005.

[GJM13] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE, 2013.

[GLR89] Giulio Giunta, Giuliano Laccetti, and Mariarosaria Rizzardi. More on the Weeks method for the numerical inversion of the Laplace transform. *Numerische Mathematik*, 54(2):193–200, 1989.

[Gol66] Stanford Goldman. *Laplace transform theory and electrical transients.* Dover, 1966.

[GP05] Martin A. Grepl and Anthony T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.

[GPAM$^+$14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[GPC17] Matthieu Gresil, Adisorn Poohsai, and Neha Chandarana. Guided wave propagation and damage detection in composite pipes using piezoelectric sensors. *Procedia Engineering*, 188:148–155, 2017.

[GPy12] GPy. A Gaussian process framework in python, since 2012.

[GR70] G. H. Golub and C. Reinsch. Handbook series linear algebra: Singular value decompositionand least squares solutions. *Numerische Mathematik*, 14:403–420, 1970.

[GU16] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[GZZZ04] H. Y. Guo, L. Zhang, L. L. Zhang, and J. X. Zhou. Optimal placement of sensors for Structural Health Monitoring using improved genetic algorithms. *Smart materials and structures*, 13(3):528, 2004.

[HA03] Jorge E. Hurtado and Diego A. Alvarez. Classification approach for reliability analysis with stochastic finite-element modeling. *Journal of Structural Engineering*, 129(8):1141–1149, 2003.

[Has70] W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 51, 1970.

[Haw80] Douglas M. Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[Hes17] Jan S. Hesthaven. *Numerical methods for conservation laws: From analysis to algorithms.* SIAM, 2017.

[HH84] G. Honig and U. Hirdes. A method for the numerical inversion of Laplace transforms. *Journal of Computational and Applied Mathematics*, 10(1):113–132, 1984.

[HKB19] Sahand Hariri, Matias Carrasco Kind, and Robert J. Brunner. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[HKP13] Dinh Bao Phuong Huynh, David J. Knezevic, and Anthony T. Patera. A static condensation reduced basis element method: approximation and a posteriori error estimation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(1):213–251, 2013.

[HL10] Ulrich L. Hetmaniuk and Richard B. Lehoucq. A special finite element method based on component mode synthesis. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(3):401–420, 2010.

[HMT11] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[HO08] Bernard Haasdonk and Mario Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 42(2):277–302, 2008.

[Hor95] Cornelius O. Horgan. Korn's inequalities and their applications in continuum mechanics. *SIAM review*, 37(4):491–511, 1995.

[HRS15] Jan S. Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations.* SpringerBriefs in Mathematics. Springer International Publishing, 2015.

[Hur65] Walter C. Hurty. Dynamic analysis of structural systems using component modes. *AIAA journal*, 3(4):678–685, 1965.

[IC04] Jeong-Beom Ihn and Fu-Kuo 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.

[JK08] Stephen Joe and Frances Y. Kuo. Constructing Sobol sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing*, 30(5):2635–2654, 2008.

[JLS15] Miche Jansen, Geert Lombaert, and Mattias Schevenels. Robust topology optimization of structures with imperfect geometry based on geometric nonlinear analysis. *Computer Methods in Applied Mechanics and Engineering*, 285:452–467, 2015.

[JMG95] Nathalie Japkowicz, Catherine Myers, and Mark Gluck. A novelty detection approach to classification. In *International Conference on Artificial Intelligence*, volume 1, pages 518–523, 1995.

[KHSM11] Safa Khazai, Saeid Homayouni, Abdolreza Safari, and Barat Mojaradi. Anomaly detection in hyperspectral images based on an adaptive Support Vector method. *IEEE Geoscience and Remote Sensing Letters*, 8(4):646–650, 2011.

[KKAG14] Naimul M. Khan, Riadh Ksantini, Imran S. Ahmad, and Ling Guan. Covariance-guided one-class Support Vector Machine. *Pattern Recognition*, 47(6):2165–2177, 2014.

[KKW20] Michael G. Kapteyn, David J. Knezevic, and Karen Willcox. Toward predictive digital twins via component-based reduced-order models and interpretable machine learning. In *AIAA Scitech 2020 Forum*, page 0418, 2020.

[KM17] M. Komeili and C. Menon. Modelling the dynamic response of a microcantilever excited at its base by an arbitrary thermal input using Laplace transformation. *Applied Mathematical Modelling*, 43:481–497, 2017.

[KN98] Edwin M. Knox and Raymond T. Ng. Algorithms for mining distancebased outliers in large datasets. In *Proceedings of the international conference on very large data bases*, pages 392–403. Citeseer, 1998.

[KRCS03] Jeong-Tae Kim, Yeon-Sun Ryu, Hyun-Man Cho, and Norris Stubbs. Damage identification in beam-type structures: frequency-based method vs mode-shape-based method. *Engineering structures*, 25(1):57–67, 2003.

[KS06] Jari Kaipio and Erkki Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media, 2006.

[KSG08] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008.

[KW13] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[KZW+14] Nguyen L. D. Khoa, Bang Zhang, Yang Wang, Fang Chen, and Samir Mustapha. Robust dimensionality reduction and damage detection approaches in Structural Health Monitoring. *Structural Health Monitoring*, 13(4):406–417, 2014.

[LAW15] Marc Lecerf, Douglas Allaire, and Karen Willcox. Methodology for dynamic data-driven online flight capability estimation. *AIAA Journal*, 53(10):3073–3087, 2015.

[LB14] James Long and Oral Buyukozturk. Automated Structural Damage Detection Using One-Class Machine Learning. In *Dynamics of Civil Structures, Volume 4*, pages 117–128. Springer, 2014.

[LC20] Kookjin Lee and Kevin T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404, 2020.

[LeV92] Randall J. LeVeque. *Numerical methods for conservation laws*, volume 3. Springer, 1992.

[LG86] James N. Lyness and Giulio Giunta. A Modification of the Weeks Method for Numerical Inversion of the Laplace Transform. *Mathematics of Computation*, 47(175):313–322, 1986.

[LHSL02] Shaw-Wen Liu, Jin H Huang, Jen-Chun Sung, and C. C. Lee. Detection of cracks using neural networks and computational mechanics. *Computer methods in applied mechanics and engineering*, 191(25-26):2831–2845, 2002.

[LML+17] Chenzhao Li, Sankaran Mahadevan, You Ling, Sergio Choze, and Liping Wang. Dynamic Bayesian network for aircraft wing health monitoring digital twin. *Aiaa Journal*, 55(3):930–941, 2017.

[LMW12] Anders Logg, Kent-Andre Mardal, and Garth N. Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*. Springer, 2012.

[LS07] B. C. Lee and W. J. Staszewski. Sensor location studies for damage detection with Lamb waves. *Smart materials and structures*, 16(2):399, 2007.

[LSZ15] Kody Law, Andrew Stuart, and Kostas Zygalakis. *Data Assimilation*. Springer, 2015.

[LTZ08] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[LY01] X. Lin and F. G. Yuan. Diagnostic Lamb waves in an integrated piezoelectric sensor/actuator plate: analytical and experimental studies. *Smart Materials and Structures*, 10(5):907, 2001.

[MAC17] Charilaos Mylonas, Imad Abdallah, and Eleni Chatzi. Surrogate modelling for fatigue damage of wind-turbine blades using polynomial chaos expansions and non-negative matrix factorization. In *IABSE Symposium Vancouver 2017: Engineering the Future*, volume 109, pages 801–808. International Association for Bridge and Structural Engineering, 2017.

[MAC20] Charilaos Mylonas, Imad Abdallah, and Eleni Chatzi. Deep unsupervised learning for condition monitoring and prediction of high dimensional data with application on windfarm scada data. In *Model Validation and Uncertainty Quantification, Volume 3*, pages 189–196. Springer, 2020.

[MAT19] MATLAB. *Version 9.5 (R2019a).* The MathWorks Inc., Natick, Massachusetts, 2019.

[MLNM20] Xirui Ma, Yizhou Lin, Zhenhua Nie, and Hongwei Ma. Structural damage identification based on unsupervised feature-extraction via variational autoencoder. *Measurement*, 2020.

[MM07] Jennifer E. Michaels and Thomas E. Michaels. Guided wave signal processing and image fusion for in situ damage localization in plates. *Wave motion*, 44(6):482–492, 2007.

[MMP02] Pabitra Mitra, C. A. Murthy, and Sankar K. Pal. Unsupervised feature selection using feature similarity. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):301–312, 2002.

[Mor19] Maurizio Morgese. *Vulnerability of existing infrastructures: the Morandi bridge and SHM possible prospects.* PhD thesis, Politecnico di Torino, 2019.

[MR90] Almerico Murli and Mariarosaria Rizzardi. Algorithm 682: Talbot's method of the Laplace inversion problems. *ACM Transactions on Mathematical Software (TOMS)*, 16(2):158–168, 1990.

[Mur12] Kevin P. Murphy. *Machine learning: a probabilistic perspective.* MIT press, 2012.

[MVE+15] Erik Marchi, Fabio Vesperini, Florian Eyben, Stefano Squartini, and Björn 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.

[New59] Nathan M. Newmark. A method of computation for structural dynamics. *Journal of the engineering mechanics division*, 85(3):67–94, 1959.

[NJL$^+$18] Pathirage C. S. Nadith, Li Jun, Li Ling, Hao Hong, and Liu Wanquan. Application of deep autoencoder model for structural condition monitoring. *Journal of Systems Engineering and Electronics*, 29(4):873–880, 2018.

[NLL$^+$19] Pathirage C. S. Nadith, Jun Li, Ling Li, Hong Hao, Wanquan Liu, and Ruhua Wang. Development and application of a deep learning–based sparse autoencoder framework for structural damage identification. *Structural Health Monitoring*, 18:103–122, 2019.

[NMA15] Federico Negri, Andrea Manzoni, and David Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *Journal of Computational Physics*, 303:431–454, 2015.

[OCDS17] Yaowen Ou, Eleni N. Chatzi, Vasilis K. Dertimanis, and Minas D. Spiridonakos. Vibration-based experimental damage detection of a small-scale wind turbine blade. *Structural Health Monitoring*, 16(1):79–96, 2017.

[OSM19] Wieslaw Ostachowicz, Rohan Soman, and Pawel Malinowski. Optimization of sensor placement for Structural Health Monitoring: A review. *Structural Health Monitoring*, 18(3):963–988, 2019.

[Pap04] Costas Papadimitriou. Optimal sensor placement methodology for parametric identification of structural systems. *Journal of sound and vibration*, 278(4-5):923–947, 2004.

[PCCT14] Marco A. F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[PJ09] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for K-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009.

[PM71] Robert Piessens and Branders 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.

[PM16] Liqian Peng and Kamran Mohseni. Symplectic model reduction of Hamiltonian systems. *SIAM Journal on Scientific Computing*, 38(1):A1–A27, 2016.

[PMQ18] Stefano Pagani, Andrea Manzoni, and Alfio Quarteroni. Numerical approximation of parametrized problems in cardiac electrophysiology by a local reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 340:530–558, 2018.

[PVG$^+$11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn:

Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[PZÖ+18] Evangelia Pippa, Evangelia I. Zacharaki, Ahmet Turan Özdemir, Billur Barshan, and Vasileios Megalooikonomou. Global vs local classification models for multi-sensor data fusion. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, pages 1–5, 2018.

[QCR05] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

[QLAC+19] Giacomo Quaranta, Elena Lopez, Emmanuelle Abisset-Chavanne, Jean Louis Duval, Antonio Huerta, and Francisco Chinesta. Structural health monitoring by combining machine learning and dimensionality reduction techniques. *Revista internacional de métodos numéricos para cálculo y diseño en ingeniería*, 35(1), 2019.

[QMN15] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2015.

[QS06] Alfio Quarteroni and Fausto Saleri. Scientific Computing with MATLAB and Octave, 2006.

[Qua09] Alfio Quarteroni. *Numerical models for differential problems*, volume 2. Springer, 2009.

[RBH18] Fabien Rochat, Caterina Bigoni, and Jan S. Hesthaven. Bayesian inference in parametric acoustic wave equation. Master's thesis, EPFL-MCSS, 2018.

[RC15] Sebastian Reich and Colin Cotter. *Probabilistic forecasting and Bayesian data assimilation.* Cambridge University Press, 2015.

[RHP07] Gianluigi Rozza, Dinh D. P. Huynh, and Anthony 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.

[RMMC20] Luca Rosafalco, Andrea Manzoni, Stefano Mariani, and Alberto Corigliano. Fully convolutional networks for structural health monitoring through multivariate time series classification. *arXiv preprint arXiv:2002.07032*, 2020.

[SB01] Alex J. Smola and Peter L. Bartlett. Sparse greedy Gaussian process regression. In *Advances in neural information processing systems*, pages 619–625, 2001.

[SBH19] Emeric Sibieude, Caterina Bigoni, and Jan S. Hesthaven. Comparative study of several one-class classification methods for simulation-based anomaly detection and damage localization. Semester project, EPFL - MCSS, 2019.

[SD08] S. N. Sivanandam and S. N. Deepa. Genetic algorithms. In *Introduction to genetic algorithms*, pages 15–37. Springer, 2008.

[SDC20] Thomas Simpson, Nikolaos Dervilis, and Eleni Chatzi. On the use of nonlinear normal modes for nonlinear reduced order modelling. *arXiv preprint arXiv:2007.00466*, 2020.

[SF73] Gilbert Strang and George J. Fix. *An analysis of the finite element method.* Prentice-hall, 1973.

[SFB$^+$06] R. Andrew Swartz, Eric Flynn, Daniel Backman, R. Jason Hundhausen, and Gyuhae Park. Active piezoelectric sensing for damage identification in honeycomb aluminum panels. In *Proceedings of 24th Intl. Modal Analysis Conference*, 2006.

[SG06] Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

[SK09] Il'ya M. Sobol and Sergei Kucherenko. Derivative based global sensitivity measures and their link with global sensitivity indices. *Mathematics and Computers in Simulation*, 79(10):3009–3017, 2009.

[SLZ00] Z. Y. Shi, S. S. Law, and L. M. Zhang. Optimum sensor placement for structuraldamage detection. *Journal of Engineering Mechanics*, 126(11):1173–1179, 2000.

[SM09] Afshin Sadri and Koorosh Mirkhani. Wave Propagation Concrete NDT Techniques for Evaluation of Structures and Materials, 2009.

[SMKO18] R. Soman, Pawel Malinowski, Pawel Kudela, and Wieslaw Ostachowicz. Analytical, numerical and experimental formulation of the sensor placement optimization problem for guided waves. In *Proceedings of the 9th EWSHM Conference, Manchester, UK*, pages 10–13, 2018.

[Sob01] Ilya M. Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001.

[sof20] Akselos software. https://akselos.com, 2020.

[SPST$^+$01] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[SRA$^+$08] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[SS00] Alex J. Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for machine learning, 2000.

[STCR04] Andrea Saltelli, Stefano Tarantola, Francesca Campolongo, and Marco Ratto. *Sensitivity analysis in practice: a guide to assessing scientific models*, volume 1. Wiley Online Library, 2004.

[Stu10] Andrew M Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

[Sud89] E. A. Sudicky. The Laplace transform Galerkin technique: A time-continuous finite element theory and application to mass transport in groundwater. *Water Resources Research*, 25(8):1833–1846, 1989.

[SWL03] Matthias Seeger, Christopher Williams, and Neil Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Proc. Ninth International Workshop on Artificial Intelligence and Statistics, ed. by C. Bishop and B. J. Frey*. Society for Artificial Intelligence and Statistics., 2003.

[SWS$^+$00] Bernhard Schölkopf, Robert C. Williamson, Alex J. Smola, John Shawe-Taylor, and John C. Platt. Support Vector Method for Novelty Detection. In *Advances in neural information processing systems*, pages 582–588, 2000.

[Tal79] Alan Talbot. The accurate numerical inversion of Laplace transforms. *IMA Journal of Applied Mathematics*, 23(1):97–120, 1979.

[TD04] David M. J. Tax and Robert P. W. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[Tit09] Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

[TKA16] M. Thiene, Z. Sharif Khodaei, and M. H. Aliabadi. Optimal sensor placement for maximum area coverage (MAC) for damage localization in composite structures. *Smart materials and structures*, 25(9):095037, 2016.

[TM03] David M. J. Tax and Klaus-R. Müller. Feature extraction for one-class classification. In *Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003*, pages 342–349. Springer, 2003.

[TPYP18] Tommaso Taddei, J. D. Penn, M. Yano, and Anthony T. Patera. Simulation-Based Classification; a Model-Order-Reduction Approach for Structural Health Monitoring. *Archives of Computational Methods in Engineering*, 25(1):23–45, 2018.

[TY20] Tommaso Taddei and Lei Yhang. Space-time least-squares Petrov-Galerkin projection for nonlinear model reduction. *arXiv preprint arXiv:2004.06693*, 2020.

[UCC$^+$17] Filippo Ubertini, Gabriele Comanducci, Nicola Cavalagli, Anna Laura Pisello, Annibale Luigi Materazzi, and Franco Cotana. Environmental effects on natural frequencies of the san pietro bell tower in Perugia, Italy, and their removal for structural performance assessmentietro bell tower in perugia, italy, and their removal for structural performance assessment. *Mechanical Systems and Signal Processing*, 82:307–322, 2017.

[Unn03] Runar Unnthorsson. Model selection in one class ν-SVMs using RBF kernels. *Proc. of 16th Int. Cong. CAMADEM, 2003*, 2003.

[Vap98] Vladimir N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, 1998.

[VGO$^+$20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: Fundamental algorithms for scientific computing in Python. *Nature methods*, pages 1–12, 2020.

[Vih12] Matti Vihola. Robust adaptive Metropolis algorithm with coerced acceptance rate. *Statistics and Computing*, 22(5):997–1008, 2012.

[VPRP03] Karen Veroy, Christophe Prud'Homme, Dimitrios Rovas, and Anthony Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *16th AIAA Computational Fluid Dynamics Conference*, page 3847, 2003.

[Wee66] William T. Weeks. Numerical Inversion of Laplace Transforms Using Laguerre Functions. *Journal of the ACM (JACM)*, 13(3):419–429, 1966.

[Wei99] Jacob 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.

[WR96] Christopher K. I. Williams and Carl Edward Rasmussen. Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520, 1996.

[WR06] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[WRH20] Qian Wang, Nicolò Ripamonti, and Jan S. Hesthaven. Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism. *Journal of Computational Physics*, page 109402, 2020.

[WWBG20] D. J. Wagg, Keith Worden, R. J. Barthorpe, and Paul Gardner. Digital twins: State-of-the-art and future directions for modeling and simulation in engineering dynamics applications. *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, 6(3), 2020.

[WXLZ03] Wenjian Wang, Zongben Xu, Weizhen Lu, and Xiaoyun Zhang. Determination of the spread parameter in the Gaussian kernel for classification and regression. *Neurocomputing*, 55(3-4):643–663, 2003.

[XWX15] Yingchao Xiao, Huangang Wang, and Wenli Xu. Parameter selection of Gaussian kernel for one-class SVM. *IEEE transactions on cybernetics*, 45(5):941–953, 2015.

[ZA96] Kornelija Zgonc and Jan D. Achenbach. A neural network for crack sizing trained by Finite Element calculations. *NDT & E International*, 29(3):147–155, 1996.

[ZBA13] F. Zou, I Benedetti, and M. H. Aliabadi. A Boundary Element Model for Sructural Health Monitoring using piezoelectric transducers. *Smart Materials and Structures*, 23(1):015022, 2013.

[ZGH19] Zhenying Zhang, Mengwu Guo, and Jan S. Hesthaven. Model order reduction for large-scale structures with local nonlinearities. *Computer Methods in Applied Mechanics and Engineering*, 353:491–515, 2019.

[ZT05] Olek C. Zienkiewicz and Robert L. Taylor. *The Finite Element Method for solid and structural mechanics.* Elsevier, 2005.

[ZXSO16] F. L. Zhang, H. B. Xiong, W. X. Shi, and X. Ou. Structural health monitoring of Shanghai Tower during different stages using a Bayesian approach. *Structural Control and Health Monitoring*, 23(11):1366–1384, 2016.

[ZZCA13] Siliang Zhang, Ping Zhu, Wei Chen, and Paul Arendt. Concurrent treatment of parametric uncertainty and metamodeling uncertainty in robust design. *Structural and multidisciplinary optimization*, 47(1):63–76, 2013.

Caterina Bigoni

✉ cate.bigoni@gmail.com · in caterina.bigoni · 🎂 18.09.1991 · ⚑ Italian

EDUCATION

Swiss Federal Institute of Technology, Lausanne (EPFL) 01.2016 - Present
PhD in Computational Mathematics

Oral exam on July 27th, 2020

Swiss Federal Institute of Technology, Lausanne (EPFL) 09.2013 - 07.2015
MSc in Computational Science and Engineering (CSE)

Final Grade: 5.53/6

Politecnico di Milano, Italy 09.2010 - 09.2013
BSc in Mathematical Engineering

Exchange semester at Grenoble Institute of Technology, France (Fall 2012)
Final grade: 102/110

CORE EXPERIENCES

Chair of Computational Mathematics and Simulation Science, EPFL 01.2016 - Present
PhD thesis

Title: "Numerical methods for structural anomaly detection using model order reduction and data-driven techniques" under the supervision of Prof. Jan S. Hesthaven.

Chair of Computational Mathematics and Simulation Science, EPFL 02.2015 - 12.2015
Master thesis and graduate research assistant

Implemented a weighted essentially non-oscillatory (WENO) method based on radial basis functions (RBF) and designed a simple algorithm to choose the shape parameter to control the balance between achievable accuracy and numerical stability. The proposed finite-volume numerical method of arbitrarily high order allows to solve hyperbolic conservation laws with discontinuous solutions and is suitable for non-uniform grids and mesh adaptation.

Mathematics Section, EPFL 09.2014 - 02.2020
Teaching assistant and project supervisor

- Prepared exercise sheets, coached exercise sessions, and occasionally substituted the lecturer of Analysis I, Analysis II, Numerical Analysis and Linear Algebra (in French).
- Co-supervised 4 Master student research projects ranging from machine learning algorithms for outlier detection to Bayesian inference to numerical conservation laws.

ABB, Vittuone (Milan), Italy 07.2014 - 09.2014
Intern in the Computational Electromagnetism R&D unit

Updated an in-house computational tool by improving a piece of software for the numerical approximation of Maxwell equations using Whitney differential forms in 3D.

ADDITIONAL EXPERIENCES

Innovation Forum Lausanne (IFL), Switzerland 12.2016 - 10.2018
Vice President

IFL is a no-profit association, part of a global network, which aims at fostering the next generation of scientists-entrepreneurs. My principal responsibilities included the organization and content creation of several scientific events (300+ guests), the promotion of local startup ecosystem (organized the local phase of an international startup accelerator, wrote several startup success stories), and the co-coordination of a team of 15.

Edition Digilex, Lausanne, Switzerland 12.2013 - 12.2015
Translator

Translated a mathematics textbook for the Swiss professional secondary diploma from French to Italian.

TECHNICAL SKILLS

Programming	Matlab, Python, C++
Scientific frameworks	FEniCS, RBniCS, scikit-learn, GPy, PyTorch
Version control	Git
Others	LaTeX

LANGUAGES

Italian	Native
English	Fluent (C1), IELTS Acedemic certificate with grade 7.5/9 (2011)
French	Fluent (C1)
German	Basic (A1/A2), Attended courses at EPFL Language Center in 2019 and 2020

AWARDS

- SIAM Student Travel Award to attend the SIAM Conference on Uncertainty Quantification (2020)
- Exceptional service prize awarded by EPFL (2018)
- ERASMUS scholarship: six-months exchange program at Grenoble Institute of Technology, France (Fall 2012)

PUBLICATIONS

1. **Bigoni C.**, Zhang Z. and Hesthaven, J. S., (2020). Systematic sensor placement for structural anomaly detection in the absence of damaged states. *Computer Methods in Applied Mechanics and Engineering, 371, 113315.*
2. **Bigoni C.** and Hesthaven, J. S., (2020). Simulation-Based Anomaly Detection and Damage Localization: an Application to Structural Health Monitoring. *Computer Methods in Applied Mechanics and Engineering, 363, 112896.*
3. **Bigoni C.** and Hesthaven, J. S., (2017). Adaptive WENO methods based on radial basis function reconstruction. *Journal of Scientific Computing, 72(3), pp.986-1020.*

CONFERENCES & WORKSHOPS

- SIAM UQ 20 – "Achieving a data-model synergy in UQ" mini-symposium (talk and co-organizer of the MS)
  March 24-27, 2020 Garching Campus of TUM, Munich, Germany (cancelled due to Covid-19)
- UNCECOMP 19 – "UQ in vibration-based monitoring & structural dynamics simulation" mini-symposium (talk)
  June 24-26, 2019 – Hersonissos, Crete, Greece
- MATHICSE Retreat (talk)
  June 11-13, 2019 – Champery, Switzerland
- Swiss Numerics Day (talk)
  May 10, 2019 – Lugano, Switzerland
- Applied Machine Learning Days – "Applied ML for anomaly detection on equipment" workshop (attended)
  January 26, 2019 – EPFL, Switzerland
- Mathworks workshop "Demystifying Deep Learning: A practical approach in Matlab" (attended)
  October 19, 2018 – UNIL, Switzerland
- SIAM UQ18 (poster presentation)
  April 16-19, 2018 – Garden Grove, California USA
- Swiss Numerics Day (attended)
  April 28, 2017 – Basel, Switzerland
- Applied Machine Learning Days (attended)
  January 30-31, 2017 – EPFL, Switzerland
- ESA Earth Observation Summer School: Earth system monitoring & modelling (poster presentation)
  August 1-12, 2016 – Frascati, Italy
- PASC16, Platform for Advanced Scientific Computing (attended and volunteered in the organization)
  June 8-10, 2016 – EPFL, Switzerland

- MATHICSE Retreat (talk)
  June 27-29, 2016 – Leysin, Switzerland

- Oberwolfach Seminar "Data Assimilation: The mathematics of connecting dynamical systems to data" (attended)
  May 15-21, 2016 – Oberwolfach, Germany

- SIAM UQ16 (attended and volunteered in the organization)
  April 5-8, 2016 – EPFL, Switzerland

- Winter School on UQ organized by wesNUM - Numerics of west Switzerland (attended)
  February 18-21, 2016 – Münsingen, Switzerland

TEACHING EXPERIENCE

- Principal teaching assistant of Algèbre Linéaire (Dr. C. Urech) – *Fall 2019*
- Assistant coordinator of Analyse II (MER B. Buffoni) – *Spring 2018 and 2019*
- Principal teaching assistant of Analyse I (Dr. P. Wittwer) – *Fall 2018*
- Principal teaching assistant of Analyse I (Prof. A. Buffa) – *Fall 2016 and 2017*
- Teaching assistant of Analyse Avancée II (Prof. J. Stubbe) – *Spring 2016 and 2017*
- Attended "Presenting & Explaining in Class" workshop organized by CAPE, EPFL – *October 18, 2016*
- Teaching assistant of Analyse Numérique (Prof. S. Deparis) – *Spring 2015*
- Teaching assistant of Analyse I (Prof. A. Quarteroni) and Algèbre Linéaire (Prof. S. Deparis) – *Fall 2014*

CO-SUPERVISED PROJECTS

- Semester Project: Comparative study of several one-class classification methods for simulation-based anomaly detection and damage localization (Emeric Sibieude, Master Program in Mathematics) – *Spring 2019*
- Master Project: Bayesian inference in parametric acoustic wave equation (Fabien Rochat, Master Program in Mathematics) – *Fall 2017*
- Master Project: 2D adaptive WENO methods based on RBF reconstructions (Kilian Thomas, Master Program in CSE) – *Spring 2016*
- Semester Project: Two-dimensional adaptive WENO methods based on radial basis function reconstructions (Stefano Ubbiali, Master Program in CSE) – *Spring 2016*