Probabilistic learning and neural networks for statistical metamodel of liner acoustic impedance

Amritesh Sinha

École doctorale Science, Ingénierie et Environnement
Paris-Est Sup

THÈSE

Pour obtenir le grade de

Docteur de l'Université Gustave Eiffel
Spécialité doctorale "Science de l'Ingénieur"

présentée et soutenue publiquement par

Amritesh SINHA

31 May 2024

Probabilistic learning and neural networks for statistical metamodel of liner acoustic impedance

Jury		
R. Ohayon,	Professeur Emérite, CNAM, Paris	Président
R. Ghanem,	Professeur, University of Southern California	Rapporteur
D. Ryckelynck	Professeur, Mines ParisTech PSL University	Rapporteur
G. Cunha	Docteur, Ingénieur Airbus France	Co-encadrant de thèse
C. Desceliers	Professeur, Université Gustave Eiffel	Directeur de thèse
C. Soize	Professeur Emérite, Université Gustave Eiffel	Co-encadrant de thèse

Laboratoire Modélisation et Simulation Multi-Échelle (MSME)
UMR CNRS 8202 Marne-la-Vallée Cedex 2, France

Acknowledgements

I extend my sincere appreciation to Professor Christophe Desceliers for giving me the opportunity to undertake this thesis. I am especially grateful for his generous investment of time in explaining the intricacies of machine learning and for the many enriching discussions we shared throughout the duration of this thesis.

I am grateful to Professor Christian Soize for his supervision of my work. Throughout this journey, he has been a constant source of support, offering invaluable help, encouragement, and wise advice. Additionally, I sincerely appreciate the significant time and immense patience he dedicated to reviewing and providing corrections for my publications, presentations, and the current manuscript.

My heartfelt gratitude extends to Guilherme Cunha from Propulsion Physics Advanced Methods, Data Integration, & Advanced Methods of Airbus, for supervising my work and for the invaluable advice he provided throughout this endeavor.

I would also like to express my thanks to Professor Roger Ohayon for presiding over the jury, and to Professors Roger Ghanem and David Ryckelynck for agreeing to serve as rapporteurs for this work.

I would like to extend my gratitude to all my colleagues from the MSME Lab for the time we shared together, for the meals we shared, coffee breaks discussions, and the after-work activities.

Furthermore, I would like to extend my gratitude to my numerous friends, who have been helpful at every stage, from the initial search and application process to the completion of this thesis. They have played a pivotal role in helping me manage various aspects of this endeavor and have contributed to the countless unforgettable moments we've shared over the past few years.

I am deeply grateful to my family, especially my father, mother, sister, and brother-in-law for their unwavering support and the invaluable time they dedicated to enabling me to take breaks from my work when needed.

Lastly, I wish to honor the memory of my late grandparents, with a special remembrance for my grandfather, whose early influence instilled in me a lifelong spirit of curiosity and learning.

ABSTRACT - Aircraft noise has emerged as a critical concern within the aviation community due to the certification authorities increasingly stringent requirements, particularly in the realm of green aviation. There are also constraints imposed by the community living near airports. This concern encompasses both external and internal noise associated with the aircraft. In modern turbofan engines, notably those with an Ultra High Bypass Ratio (UHBR), fan noise significantly contributes to overall noise levels, characterized by both broadband and tonal noise components. Acoustic liners, designed to mitigate these components, are crucial for effective noise absorption. To ensure their effectiveness, it is imperative to study these liners under various flight conditions.

This work addresses the attenuation of low-frequency tonal noise using adapted acoustic liners. The design of liners needed for targeting low frequencies must have particular geometries and cannot be based on standard geometry acoustic liners. This means that there is a need to optimize these liners on the basis of high-fidelity simulations. However, simulations can be computationally expensive and not always feasible. Additionally, many numerical simulations simplify assumptions to ease calculations, which lead to overlooking or constant holding of certain parameters. This omission fails to capture the full variability of the system-operating conditions, leading to a partial understanding of its performance range.

The challenge lies in (i) identifying and quantifying all known sources of uncertainties and (ii) developing a model that encompasses both known and unknown variability in operating conditions to ensure robustness against these uncertainties. Generating an extensive database through exhaustive exploration of design parameters via high-fidelity simulations is not practical. Thus, a robust statistical metamodel is developed to model a parameterized aeroacoustic liner impedance as a function of frequency and main control parameters. This metamodel is constructed using a small dataset from computationally expensive aeroacoustic simulations, requiring the use of an adapted learning algorithm that is chosen as Probabilistic Learning on Manifolds (PLoM).

Despite the aeroacoustic-simulation reliance on a large computational model and the introduction of some approximations, model uncertainties are accounted for within the training dataset through a probability model. This model is refined using dimensionless data from experiments available in open literature. Since available data is limited, a novel statistical metamodel has been developed, offering consistent predictions and a confidence region for the statistical metamodel of the uncertain parameterized aeroacoustic liner impedance.

Furthermore, a statistical Artificial Neural Network (ANN)-based metamodel is introduced as another representation, offering greater versatility. It includes a prior conditional probability model for the PCA-based statistical reduced representation of the frequency-sampled vector of log-resistance and reactance. This model imposes statistical constraints, presenting challenges for training the ANN-based model using classical optimization methods. An alternative approach involves constructing a second large dataset using conditional statistics estimated with learned realizations from PLoM.

The development of these two low-computational-cost metamodels addresses low-frequency noise for which only a limited number of simulations were available, marking a significant step forward in the study of aircraft noise reduction.

Keywords: Statistical metamodel; Probabilistic learning on manifolds; Neural Networks; Liner

acoustic impedance; Small data;

RÉSUMÉ - Le bruit des avions est devenu une préoccupation majeure au sein de la communauté aéronautique en raison des exigences de plus en plus strictes des autorités de certification, en particulier dans le domaine de l'aviation verte. Il existe également des contraintes imposées par la communauté vivant à proximité des aéroports. Cette préoccupation englobe à la fois le bruit externe et interne associé à l'avion. Dans les turboréacteurs modernes à double flux, notamment ceux dotés d'un taux de dilution élevé (UHBR), le bruit du fan contribue de manière significative au niveau de bruit global, caractérisé à la fois par des composantes d'un bruit large bande et de bruits tonals. Les revêtements acoustiques, conçus pour atténuer ces bruits, sont essentiels pour une absorption efficace du bruit. Pour s'assurer de leur efficacité, il est impératif de les étudier pour diverses configurations de vol.

Cette thèse porte sur l'atténuation du bruit tonal basse fréquence à l'aide de revêtements acoustiques adaptés. La conception des revêtements acoustiques nécessaires adaptés aux basses fréquences doit utiliser des géométries particulières et ne peut pas être basée sur des géométries standards. Cela signifie qu'il est nécessaire d'optimiser ces revêtements acoustiques sur la bases de simulations numériques haute fidélité. Cependant, les simulations peuvent être coûteuses en calcul et ne peuvent pas toujours être réalisées. De plus, la limitation du nombre de simulations numériques conduit à limiter l'exploration de domaine paramétrique de conception , ce qui conduit à négliger ou à maintenir constants certains paramètres. Ces conditions ne permettent pas de capturer toute la variabilité des conditions de fonctionnement du système, ce qui conduit à une compréhension partielle de sa plage de performances.

Le défi réside dans (i) l'identification et la quantification de toutes les sources connues d'incertitudes et (ii) le développement d'un modèle qui englobe à la fois la variabilité connue et inconnue des conditions d'exploitation pour garantir la robustesse face à ces incertitudes. Générer une base de données étendue grâce à une exploration exhaustive des paramètres de conception via des simulations haute fidélité n'est pas possible. Ainsi, un métamodèle statistique robuste est développé pour modéliser l'impédance du revêtement aéroacoustique paramétré en fonction de la fréquence et des principaux paramètres de conception qui sont les paramètre de contrôle. Ce métamodèle statistique est construit à l'aide d'un petit ensemble de données issues de simulations aéroacoustiques coûteuses en termes de calcul, nécessitant l'utilisation d'un algorithme d'apprentissage, choisi comme l'apprentissage probabiliste sur les variétés (PloM).

Malgré le recours à la simulation aéroacoustique à l'aide d'un grand modèle numérique et l'introduction de certaines simplifications, les incertitudes de modélisation sont prises en compte via un modèle probabiliste. Ce modèle est affiné à l'aide de résultats expérimentaux issues de la littérature ouverte. Un métamodèle statistique est développé, offrant des prédictions cohérentes et une région de confiance pour l'impédance paramétrée du revêtement aéroacoustique en présence d'incertitudes.

De plus, un métamodèle statistique basé sur un réseau de neurones artificiels (ANN) est introduit comme autre représentation. Il comprend un modèle de probabilité conditionnelle a priori d'une représentation statistique réduite basée sur l'ACP du vecteur constitué de la log-résistance et de la réactance échantillonnées en fréquence. Ce modèle impose des contraintes statistiques, présentant des défis pour l'apprentissage du modèle basé sur ANN à l'aide de méthodes d'optimisation classiques. Une approche alternative consiste à construire un deuxième grand ensemble de données en utilisant des statistiques conditionnelles estimées avec les réalisations apprises de PLoM.

Le développement de ces deux métamodèles à faible coût de calcul aborde le bruit basse fréquence pour lequel seul un nombre limité de simulations était disponible, marquant une avancée significative dans l'étude de la réduction du bruit des avions.

Mots Clés: Métamodèle statistique; Apprentissage probabiliste sur les variétés; Réseau des neurones; Impédance des revêtements acoustiques; Petite base de données;

Contents

Chapter 1

General Introduction

Air traffic has significantly grown over the past decade. With ever-increasing environmental concerns, green aviation has come into prominence. In particular, aircraft noise is of significant interest to the aviation community, whether it be the question of external or internal noise relative to the aircraft. This has led to increasingly stringent requirements by the certification authorities. In modern turbofan engines (UHBR - Ultra high bypass ratio), fan noise is one of the main contributors to the overall aircraft noise. Fan noise can be characterized by broadband and tonal noise components. Acoustic liners (acoustic treatments) can be designed to tackle both the components. Tonal noise is mainly attenuated by resonance effects while viscous dissipation acts on both tonal and broadband noise. Noise attenuation by acoustic treatments is tuned for the blade passing frequency (BPF), whereas dissipating as much as possible the broadband component, by modifying the acoustic liner geometry or intrinsic properties. In order to be effective in absorbing fan noise, acoustic liners have to be studied in their operating environment *i.e.* in different flight conditions. Typically, noise attenuation is addressed during the early design phases of engines. However, the addition of acoustic liners after the completion of engine design poses challenges in meeting weight constraints and maintaining aircraft performance, as it increases drag within the nacelle, thereby reducing engine efficiency. Hence, the development and integration of these acoustic liners into the nacelle present significant challenges in themselves. In particular, addressing low-frequency tonal noise requires acoustic liners with deep cavities, which is not feasible in practice. This need comes from the fact that the UHBR-noise signature is in the low-frequency range. The development of complex acoustic liners must account for optimization to the target frequency while adhering to dimensional, weight, and placement constraints. Managing such a task, considering these constraints, is a considerable challenge and necessitates extensive testing, simulations, and multidisciplinary optimizations that exceed the scope of this work. In this thesis, we present the development of low-computational cost metamodels for an acoustic liner configuration, tailored to address low-frequency tonal noise, based on a limited number of simulations.

1.1 Motivations and goals

The design and eventual optimization of acoustic liners necessitates conducting numerous simulations and experiments while varying their design parameters. Simulations can often be computationally expensive or not readily available, primarily due to challenges with numerical convergence at specific design points. Moreover, most numerical simulations make simplifying assumptions to facilitate calculations, resulting in certain parameters being held constant or even disregarded in the computational model. For instance, pressure and velocity fluctuations are typically omitted in computational fluid dynamics simulations during initialization. As a consequence, the variability associated with such parameters is not integrated into the computational model, leading to an incomplete understanding of the system's full operating range and performance. Hence, the challenge is twofold: *(i)* identifying and quantifying all known sources of uncertainties and *(ii)* developing a model capable of incorporating all known and unknown variability in operating conditions, thereby ensuring robustness against these uncertainties. Generating a large database by exhaustively exploring the admissible set of design parameters, using a high-fidelity computational model, is impractical. Therefore, there is a need for a high-fidelity but computationally efficient model capable of accommodating model uncertainties and making repeated calls to the admissible set of design parameters. Finally, this model should be versatile, meaning it should be easy to use by engineers with varying levels of expertise and straightforward to integrate into design optimization loops. This is the role that will be played by the developed statistical metamodel.

1.2 State of the art

Experiments have been carried out for identifying uncertainties related to acoustic liner impedance model (see for instance [2, 3, 1, 4] and [5] for the notion of wall acoustic impedance). High-fidelity computational models have also been developed for predicting the acoustic liner performances (see for instance [8, 10, 9, 6, 7, 11]). The design of acoustic liners is of prime interest and many works have been published on this subject (see for instance [18, 14, 17, 12, 15, 13, 16]). Statistical inference such as Bayesian approaches have recently been used for statistical inverse problems related to acoustic liner impedance (see for instance [19, 20]). The acoustic performance of a liner depends on the quantities that are highly related to the operating conditions, such that velocity, mean pressure, and fluid density. Any external variation directly impacts the environment of the acoustic liners and thus the acoustic performance of the liner system.

This requires to take into account uncertainties in the high-fidelity computational model of the liner system. An uncertain computational model of the liner system is presented in [13], which allows for quantifying uncertainties in aeroacoustic models of liner performance. In the absence of sufficient data, we need to generate a large dataset of additional realizations using a probabilistic learning tool that involves a low computational cost, such as Probabilistic Learning on Manifolds(PLoM) [21, 22, 23, 24] (see appendix A for details). Nonparametric conditional statistics and these learned realizations can then be used to construct a statistical metamodel [26, 25].

The machine learning tools and artificial intelligence [27, 28, 29], such as the probabilistic and statistical learning [31, 34, 32, 33, 30], are used in uncertainty quantification for problems that would require computer resources that are not available with the most usual approaches. Thus, methods have emerged in the field of engineering sciences, such as the learning on manifolds [21, 40, 38, 37, 39, 36, 22, 35, 41, 23] and the physics-informed probabilistic learning [43, 44, 45, 42, 46, 47]. Artificial neural networks [51, 50, 52, 48, 49] can then be used to create a mapping of the hyperparameters, which provides a versatile representation of the metamodel.

Open literature on experiments conducted on acoustic liners can be found in [55, 53, 54] that can be then used to account for model uncertainties in this statistical metamodel.

1.3 Objectives of the thesis

The objectives of this thesis can be summarized as follows:

1. Development of a high-fidelity, high-computational-cost small database for an acoustic liner. This database captures extreme values within the admissible set but may not encompass all intermediate points in this admissible set.

2. Identification of model uncertainties through experiments and their integration into a probability model of the acoustic liner impedance.

3. For Mach number equal to zero ($M = 0$) and for a limited number of design parameters, use of Probabilistic Learning on Manifolds (PLoM) to generate a large dataset of realizations, followed by the construction of nonparametric conditional statistics (using Gaussian Kernel Density Estimation (GKDE)), yielding a PLoM-GKDE-based statistical metamodel.

4. For Mach number equal to zero ($M = 0$) and for limited number of design parameters, construction of an Artificial Neural Network (ANNs)-based statistical metamodel, which serves as another representation. This ANN-based statistical metamodel is the primary deliverable.

5. For positive Mach numbers ($M > 0$) and for a larger number of design parameters, construction of a PLoM-GKDE-based statistical metamodel and its ANN-based representation as the ANN-based statistical metamodel.

6. These two ANN-based statistical metamodels ($M = 0$ and $M > 0$), can also be used as generators of realizations of the random acoustic liner impedance.

1.4 Organization of the thesis

In this thesis, the author has intentionally crafted each chapter to be self-contained, enabling them to be read independently of one another as a deliberate stylistic choice. In Chapter 2, model uncertainties are identified from open literature and integrated into the statistical metamodel, with the Mach number kept

fixed at 0. Chapter 3 introduces another representation of the PLoM-GKDE-based statistical metamodel developed in chapter 2, using an ANN-based statistical metamodel. Comparison of the predictions given by these two statistical metamodels are presented. Chapter 4 focuses on simulations conducted for nonzero Mach number, and showcases both the two statistical metamodels. In Chapter 5, additional insights for the neural network-based model and its training process. Finally, the appendices contain mathematical results relevant to the presented developments, followed by References.

Chapter 2

Statistical metamodel based on a PLoM-GKDE formulation for Mach numbers fixed at zero

2.1 Introduction

The content of this chapter is taken from the paper [56] and is devoted to the PLoM-GKDE-based statistical metamodel. Since this chapter is only devoted to the PLoM-GKDE-based statistical metamodel, we will simply write "statistical metamodel".

2.1.1 Objectives

A liner is characterized by its acoustic absorption, which will be modeled by its (local) acoustic impedance adapted to the low-frequency tonal noise. In this work, the liner acoustic impedance is estimated using an aeroacoustic computational model (ACM) applied to a simplified, but representative, configuration of the liner (see Section 2.2.2). One evaluation with such an ACM is computationally expensive and consequently, the ACM cannot be used many times for constructing a parameterized aeroacoustic liner impedance. This is the reason why we propose to develop a statistical metamodel of the impedance whose parameterization is frequency ω, percentage of open area (POA), and sound pressure level (SPL). The Mach number has been set to zero to minimize the number of parameters that need to be considered for the metamodel. In fact, including a non-zero Mach number would require analyzing how the impedance changes with respect to various boundary layer properties (such as friction velocity, boundary layer thickness, local Reynolds number, bulk Mach number, and so on), which is beyond the scope of this study. Further investigations on this topic are currently underway and will be addressed in future work. Due to the use of a simplified configuration, it is of prime importance to take into account model uncertainties induced by modeling errors in the construction of the impedance model. These uncertainties on the impedance model will be introduced in the statistical metamodel of the impedance and the level of

uncertainties will be identified with experiments. We thus propose to develop a statistical metamodel of the parameterized acoustic liner impedance, which is robust with respect to uncertainties. The objective of the chapter is therefore to present a useful methodology to build a robust statistical metamodel of the acoustic liner impedance as a function of the control parameters. The property of robustness is necessary with respect to uncertainties but also to the small amount of data available. A methodology is thus presented to circumvent the prohibitive computational cost of building a parameterized model of the aeroacoustic liner impedance. Such a statistical metamodel can, for example, be used to carry out a parametric analysis of the acoustic liner impedance, to obtain its acoustic performance, and can also be included in an optimization loop. Such possible applications are beyond the scope of the thesis.

2.1.2 Why POA is chosen as a design parameter

Let t, d, and L be the perforated plate thickness, the hole diameter, and the length from the perforated plate to the rigid backing sheet, respectively. Parameters t, d, and L have been held constant and equal to 1.0 mm, 0.8 mm, and 9.6 mm, respectively. It is important to note that the POA is one of the most important parameter when considering liner design. For the range of liners considered, height L of the cavity mainly influences the reactance, by supposing being far from the antiresonance, and has a negligible impact on the resistance (which is mainly a function of the geometric parameters of the resistive layer). However, it is possible to separate the reactance contribution of the cavity from that of the resistive sheet, by subtracting the cotangent term from the reactance value obtained. This is done in Eqn. (2.3). Note that, the results are not independent of L. We are simply observing the orifice velocity instead, at the cavity height L. This will allow the results to be compared with those of the literature. Regarding thickness t, its value has been set at 1.0 mm. This value is representative of what is used in the aeronautical industry. It should also be noted that the integration constraints make the chosen thickness value a very representative value.

2.1.3 Sources of uncertainties and variabilities, and their consideration in the development of the metamodel

There are two types of uncertainties: the uncertainties on the model parameters called the "model-parameter uncertainties" and the "model uncertainties" induced by the modeling errors. In addition, there are some "variabilities" in the real system, due to manufacturing process and due to small differences in the configurations: an experimental configuration of a complex system differs from the designed system and is never perfectly known. When a computational model of a complex system is developed, model-parameter uncertainties, model uncertainties, and variabilities have to be taken into account. When experiments are available, the probability model of uncertainties has to be identified by solving a statistical inverse problem. If information is not available for the construction of an informative probability model of model-parameter uncertainties (that is the case for the considered problem), model-parameter uncertainties and model uncertainties can simultaneously be taken into account using a nonparametric probability model or the output-predictive-error method as explained in [57]. The latter consists in introducing an

additive or multiplicative noise on the quantities of interest of the computational model and in identifying the hyperparameters that control the probabilistic model of the noise using the experiments. It is the latter method that is proposed in this work.

2.1.4 Proposed methodology

The novelty presented in this chapter is the development of a statistical metamodel of the acoustic liner impedance, based on the use of a probabilistic learning tool and a small training dataset. Experimental data is directly integrated in the proposed statistical metamodel in order to take into account model uncertainties (see Section 2.1.1). Note that there is no direct connection between the experimental values and ACM that is constructed independently of the experimental values. The experimental values are only used to calibrate the level of the noise that is added in the model to take into account uncertainties. The learning tool is the probabilistic learning on manifolds (PLoM) [21, 22, 23]. The training dataset is generated using experimental data [55] and Lavieille's work [8] for numerical predictions of acoustic impedance. It should be noted that the development of the statistical metamodel is not performed by using a proper experimental dataset because such dataset would require carrying out a complete experimental campaign that is currently not available. In addition, it would be a challenge to be able to transfer the statistical fluctuations from different dataset (using different experimental cases) to a given computational model. The main limitation regarding third-party experimental datasets is that we do not have any information about the liners manufacturing process. This makes it difficult to compare directly any experimental data with numerical ones, unless we have quality check information on manufactured liners. For instance, it is known that the nominal geometrical parameters and the ones that are effectively manufactured might greatly differ (by more than 20%, depending solely on the manufacturing process). This means that any direct comparison would be biased, unless it is made via statistical quantities. Experimental means varies from one research/industrial institution to another and, unless we compare numerical data to experimental ones after a careful quality check, it would be rather difficult to make a quantitative comparison of all data. In this context, we have used the available experimental database from [55] for which the experimental results are presented in a dimensionless form, covering a large family of liners. As explained, such a database does not permit carrying out a direct probabilistic inference because the experimental statistical mean values cannot be used, as it does not correspond to the case being analyzed in this chapter. Only the experimental statistical fluctuations around the experimental statistical mean will be used to calibrate the noise that models uncertainties. Therefore, physics-based constraints cannot be used for the learning step. In order to circumvent this difficulty, the experimental information is integrated in the training dataset using the output predictive error model [57].

2.1.5 Organization of the chapter

Section 2.2 deals with the definition of the aeroacoustic liner system and with the aeroacoustic computational model that is used to generate the simulated data. In Section 2.3, the experimental dimensional data are derived from the experimental dimensionless data, extracted from the open literature. A comparison of the aeroacoustic computation model predictions with the dimensional experiments is presented.

Section 2.4 is devoted to the construction of the probability model of the model uncertainties, which is calibrated using the experimental dimensional data. Then the training dataset including model uncertainties is constructed. In Section 2.5 we present the construction of the statistical metamodel of the parameterized random liner impedance and then introduce conditional statistics and the methodology to estimate it on the basis of probabilistic learning. Finally, Section 2.6 deals with the presentation of the results and its discussion.

A summary of the probabilistic learning on manifolds (PLoM) algorithm, its parameterization, and the algebraic expressions of the conditional statistics are given in Appendices A and B.

2.2 Defining the liner system and its aeroacoustic computational model

2.2.1 Definition of the liner system

We consider a perforated liner system whose scheme is shown in Fig. 2.1. It is constituted of a perforated plate, a rigid backing sheet, and a honeycomb constituting drainage holes. As explained above, the parameters used to control the liner system are the POA and the SPL. We then define the control parameter as $\mathbf{w} = (w_1, w_2)$ in which w_1 is POA and w_2 is SPL. The control parameter will be modeled by a $\mathbb{R}^2$-valued random variable $\mathbf{W} = (W_1, W_2)$ whose prior probability distribution will allow for generating samples of $\mathbf{W}$.

2.2.2 Computation of the impedance using an aeroacoustic computational model

The domain decomposition used for the aeroacoustic computational model is the one proposed in [8] and is shown in Fig. 2.2. The left figure displays the scheme of elementary period d_1. The central figure shows the domain that is constituted of three subdomains: the outside domain, the resistive sheet domain, and the resonator domain of elementary period d_1. The right figure represents the reduced resonator domain of period δ_1. In the outside domain, the acoustic field is described by the sum of incident and diffracted plane waves in order to compute sound pressure level (SPL) at the upper face of the resistive sheet that allows an equivalent reduced impedance $z^{\,\mathrm{acm}}(\omega;\mathbf{w}) = \mathrm{r}^{\,\mathrm{acm}}(\omega;\mathbf{w}) + \iota\,\mathrm{v}^{\,\mathrm{acm}}(\omega;\mathbf{w})$ that depends on the frequency ω and on the control parameter $\mathbf{w}$, in which $\iota = \sqrt{-1}$, $\mathrm{r}^{\,\mathrm{acm}}(\omega;\mathbf{w})$ is the resistance, and where $\mathrm{v}^{\,\mathrm{acm}}(\omega;\mathbf{w})$ is the reactance. In the reduced resonator domain, which belongs to the resistive sheet domain, the nonlinear Navier-Stokes equations are solved in order to well capture the viscous effects and the nonlinear phenomena at higher SPL. The period δ_1 is chosen so that the corresponding POA is verified at the resistive sheet level. Then, a bi-periodic condition is imposed. This means that any generated vortex can cross the lateral boundaries and still be correctly taken into account. Consequently, the third domain (see Fig. 2.2) is sufficiently wide to capture this motion. We refer the reader to the paper [8] for the details concerning this aeroacoustic computational model. It should be noted that such a simplified aeroacoustic model yields large computational cost for exploring the flight and design configurations. Nevertheless, it demonstrates the interest of the proposed methodology for constructing a robust statistical metamodel.

More complex aeroacoustic computational models can be found in [58, 59, 60]. For each value of the POA and SPL, and for each sampled frequency $\omega_k \in \mathcal{C}_\omega = \{\omega_1, \ldots, \omega_{n_\omega}\}$, the aeroacoustic computational model computes $\mathrm{r}^{\,\mathrm{acm}}(\omega_k)$, $\mathrm{v}^{\,\mathrm{acm}}(\omega_k)$, and the hole-orifice velocity $\mathrm{v}_{\mathrm{or}}^{\mathrm{acm}}(\omega_k)$. In the computational method used, only the reduced resonator domain is meshed for Navier-Stokes computation. For the considered liner, the mesh is made up of $41\,781$ vertices, $206\,560$ elements, and $278\,514$ degrees of freedom. The mesh in the refinement zone (see Fig. 2.3) is adapted according to the POA value. For each value of the POA and SPL, and for the seven sampled frequencies, the CPU time is 448 hours (using a 64 cores computer). The computation has been done by Airbus using the SANUMO software [8] for $n_\omega = 7$ sampled frequencies and for (POA, SPL) $\in [0.03, 0.1] \times [130, 145]$. Since the resistance is very sensitive to the values of POA, in order to not artificially increase the statistical fluctuations in the conditional statistics that will be constructed using probabilistic learning, interval $[0.03, 0.1]$ is split as $[0.03, 0.05] \cup [0.06, 0.1]$, the first one containing 3 points and the second one 5 points. More precisely, the larger the "diameter" of the support (domain of the values of $\mathbf{w}$) of the probability measure of the control parameter $\mathbf{w}$, the larger the width of the confidence domain of the quantities of interest. Therefore, a compromise has to be made between the choice of the points of the training dataset and its effects on the amplitude of the statistical fluctuations of the quantities of interest, which control the width of the confidence domain. An analysis has been conducted to find the partition that is proposed. Note that the SPL interval contain 6 points. In this chapter, the presented analysis is performed for each POA-subinterval $[a_{\mathrm{POA}}, b_{\mathrm{POA}}]$. We therefore consider the values of $\mathbf{w}$ belonging to the finite set

$$\mathcal{C}_{\mathbf{w}} = \{\mathbf{w}^j, j = 1, \ldots, n_d\} \subset \mathcal{S}_{\mathbf{w}} \quad , \quad \mathbf{w}^j \in \mathbb{R}^{n_{\mathbf{w}}} \, , \tag{2.1}$$

in which $n_{\mathbf{w}} = 2$ and $n_d = 18$ for the first subinterval and $n_d = 30$ for the second one, where the points in $\mathcal{C}_{\mathbf{w}}$ are considered as samples of the random variable $\mathbf{W} = (W_1, W_2)$, which belong to the domain $\mathcal{S}_{\mathbf{w}} = [a_{\mathrm{POA}}, b_{\mathrm{POA}}] \times [130, 145] \subset \mathbb{R}^{n_{\mathbf{w}}}$ ($\mathcal{S}_{\mathbf{w}}$ can be viewed as the support of the prior probability distribution of $\mathbf{W}$). We now define the ACM simulation dataset related to the points of $\mathcal{C}_{\mathbf{w}}$. For $j \in \{1, \ldots, n_d\}$, we introduce the vectors $\mathbf{r}^{\,\mathrm{acm},j} = (\mathrm{r}^{\,\mathrm{acm},j}(\omega_1), \ldots, \mathrm{r}^{\,\mathrm{acm},j}(\omega_{n_\omega})) \in \mathbb{R}^{n_\omega}$ and $\mathbf{v}^{\,\mathrm{acm},j} = (\mathrm{v}^{\,\mathrm{acm},j}(\omega_1), \ldots, \mathrm{v}^{\,\mathrm{acm},j}(\omega_{n_\omega})) \in \mathbb{R}^{n_\omega}$ in which $\mathrm{r}^{\,\mathrm{acm},j}(\omega_k) = \mathrm{r}^{\,\mathrm{acm}}(\omega_k; \mathbf{w}^j)$ and $\mathrm{v}^{\,\mathrm{acm},j}(\omega_k) = \mathrm{v}^{\,\mathrm{acm}}(\omega_k; \mathbf{w}^j)$. Finally, for $j \in \{1, \ldots, N_d\}$, we introduce the vector $\mathbf{q}^{\,\mathrm{acm},j} = (\mathbf{r}^{\,\mathrm{acm},j}, \mathbf{v}^{\,\mathrm{acm},j}) \in \mathbb{R}^{n_{\mathbf{q}}} = \mathbb{R}^{n_\omega} \times \mathbb{R}^{n_\omega}$ with $n_{\mathbf{q}} = 2n_\omega$. The points of the ACM simulation dataset $\{\mathbf{q}^{\,\mathrm{acm},1}, \ldots, \mathbf{q}^{\,\mathrm{acm},n_d}\}$ are the realizations of the $\mathbb{R}^{n_{\mathbf{q}}}$-valued random variable $\mathbf{Q}^{\,\mathrm{acm}}$.

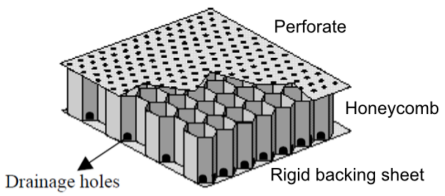


Figure 2.1: Scheme of a perforated liner. (Reprinted from [8] with permission of Maud Lavieille.).

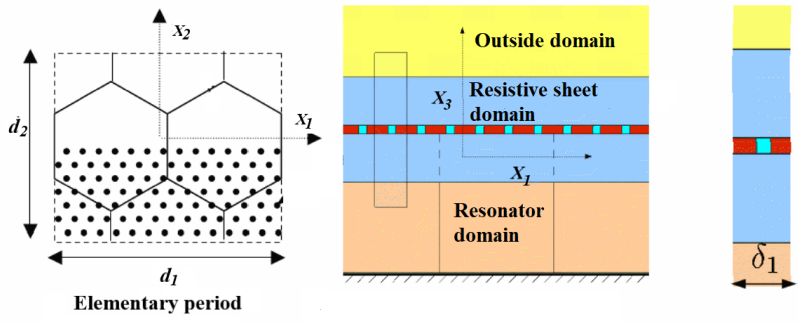


Figure 2.2: Domain decomposition of the aeroacoustic computational model.(Reprinted from [8] with permission of Maud Lavieille.).

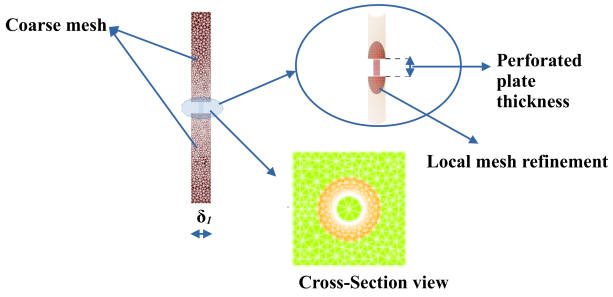


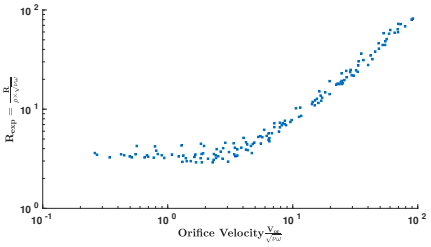
Figure 2.3: Mesh of reduced resonator domain of period: part with mesh refinement and with coarse mesh.

2.3 Experimental dimensional data and comparison with ACM simulations

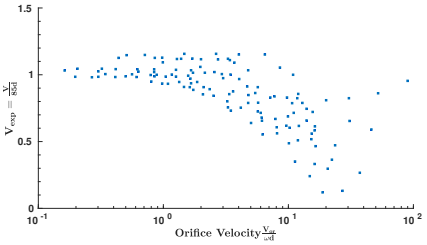
2.3.1 Experimental dimensional data

The experimental data compiled by Panton and Goldman [55] are used to calibrate the level of uncertainties in the constructed statistical metamodel. The experimental dimensionless impedance $z^{\text{exp}}_{\text{dimless}}(\Omega)$ is presented as a function of dimensionless orifice velocity $\Omega = \Omega_{\text{r}} = \text{v}_{\text{or}}(\omega; \mathbf{w})/\sqrt{\nu\omega}$ for dimensionless resistance (see Fig. 2.4a) and $\Omega = \Omega_{\text{v}} = \text{v}_{\text{or}}(\omega; \mathbf{w})/(\omega\, d)$ for dimensionless reactance (see Fig. 2.4b), in which ω is the angular frequency, $\text{v}_{\text{or}}(\omega; \mathbf{w})$ is the experimental frequency-dependent dimensional orifice velocity that depends on control parameter $\mathbf{w}$, ν is the kinematic viscosity, and d is the hole diameter. It should be noted that Ω_{r} and Ω_{v} are defined using the same frequency-dependent dimensional orifice velocity, $\text{v}_{\text{or}}(\omega; \mathbf{w})$. The images shown in Fig. 2.4 are digitized and the datapoints are extracted using Plotdigitizer software [61]. It should be noted that the resistance and reactance are not normalized in the same way in Fig. 2.4 as described at the beginning of Section 2.3.1. Nevertheless, the rescaling was properly performed. For each considered value of Ω_{r} and Ω_{v}, the mean of the closest values of the experimental dimensionless resistance and reactance are extracted, which allows for replacing the cloud of experimental dimensionless points into experimental dimensionless curve. In order to use these

experimental data for constructing the statistical metamodel, these dimensionless data are transformed as a function of dimensionless orifice velocity Ω into dimensional quantities as a function of the frequency, ω. These dimensionless experimental data are transformed into dimensional data that are compatible with



(a) Graph $\Omega_{\mathrm{r}} \mapsto \mathrm{r}^{\mathrm{exp}}_{\mathrm{dimless}}(\Omega_{\mathrm{r}})$ of the dimensionless experimental resistance



(b) Graph $\Omega_{\mathrm{v}} \mapsto \mathrm{v}^{\mathrm{exp}}_{\mathrm{dimless}}(\Omega_{\mathrm{v}})$ of the dimensionless experimental reactance

Figure 2.4: Dimensionless resistance (a) and reactance (b) (vertical axis) as a function of dimensionless orifice velocity (compiled from [55] with permission Copyright 1976, Acoustical Society of America.)

the ACM simulation data. For the experimental dimensional resistance and reactance at sampling points ω_k and for given value of control parameter $\mathbf{w}$, these transformations yield

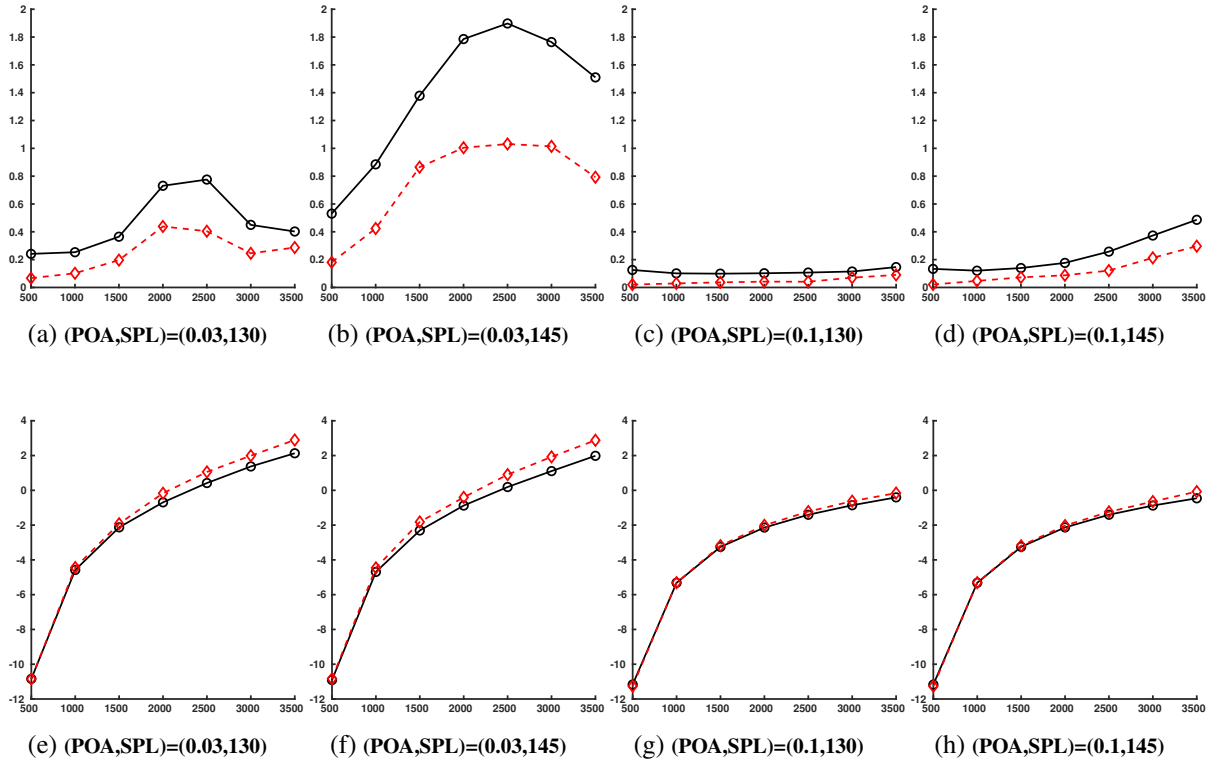
$$\mathrm{r}^{\mathrm{exp}}(\omega_k;\mathbf{w}) = \mathrm{r}^{\mathrm{exp}}_{\mathrm{dimless}}(\Omega_{\mathrm{r},k}(\mathbf{w})) \times \frac{\rho\sqrt{\nu\omega_k}}{c \times w_1} \tag{2.2}$$

$$\mathrm{v}^{\mathrm{exp}}(\omega_k;\mathbf{w}) = \left[\left(\mathrm{v}^{\mathrm{exp}}_{\mathrm{dimless}}(\Omega_{\mathrm{v},k}(\mathbf{w})) \times \frac{8}{3\pi} + \frac{t}{d}\right)\frac{\rho\,\omega_k d}{c \times w_1}\right] - \cot\left(\frac{\omega_k L}{c}\right) \tag{2.3}$$

in which cot is the cotangent, ω_k is a sampled frequency point, ρ is the air density, c is the speed of sound, w_1 is the percentage of open area (POA), t is the perforated plate thickness, and L is length from the perforated plate to the rigid backing sheet. The $\mathbf{w}$-dependent dimensionless frequencies are such that $\Omega_{\mathrm{r},k}(\mathbf{w}) = \mathrm{v}^{\mathrm{acm}}_{\mathrm{or}}(\omega_k;\mathbf{w})/\sqrt{\nu\,\omega_k}$ and $\Omega_{\mathrm{v},k}(\mathbf{w}) = \mathrm{v}^{\mathrm{acm}}_{\mathrm{or}}(\omega_k;\mathbf{w})/(\omega_k d)$, in which $\mathrm{v}^{\mathrm{acm}}_{\mathrm{or}}(\omega_k;\mathbf{w})$ is the orifice velocity computed with ACM for a given ω_k and $\mathbf{w}$. Eqns. (2.2) and (2.3) are used for converting the dimensionless data plotted in Fig. 2.4 to the ones used in this chapter. The dimensionless factors are the ones proposed by Panton in [55]. These parameters happen to have a link with the Crandall model, but as well as other models, such as Poiseuille's, as described in [55]. The dashed curves with diamond markers represent, in Figs. 2.5-(a) to (d), the experimental results for the resistance for the considered values of $\mathbf{w}$ that is the pair (POA, SPL), and in Figs. 2.5-(e) to (h), the reactance for the same values of $\mathbf{w}$.

2.3.2 Comparison of the ACM predictions with experiments

Fig. 2.5 shows the comparison between the ACM simulation dataset and the experimental dimensional dataset for the resistance and for the reactance, for several values of $\mathbf{w}$, the pair (POA, SPL). In Figs. 2.5-(a) to (d), the solid curves with circular markers represent the computational results for the resistance for different values of $\mathbf{w}$ that is the pair (POA, SPL). In Figs. 2.5-(e) to (h), the reactance is plotted for the same values of $\mathbf{w}$. In these figures, the POA and SPL values correspond to its minimum and maximum. These comparisons show that the experimental mean is far from the ACM simulations for the resistance whose good prediction is of prime interest. Consequently, even if the variations are of



(a) **(POA,SPL)=(0.03,130)** (b) **(POA,SPL)=(0.03,145)** (c) **(POA,SPL)=(0.1,130)** (d) **(POA,SPL)=(0.1,145)**

(e) **(POA,SPL)=(0.03,130)** (f) **(POA,SPL)=(0.03,145)** (g) **(POA,SPL)=(0.1,130)** (h) **(POA,SPL)=(0.1,145)**

Figure 2.5: Resistance (figures (a) to (d)) and reactance (figures (e) to (h)) (vertical axis) as a function of frequency (Hz) (horizontal axis), comparison of the ACM simulation data (solid curves with circular markers) with experimental dimensional data (dashed curves with diamond markers) for several values of $\mathbf{w}$ that is the pair (POA, SPL).

the same type for experiments and simulations, these experimental values cannot directly be used for a probabilistic inference. Nevertheless, it will be assumed that the experimental statistical fluctuations around the experimental mean is representative of the level of model uncertainties and consequently, will be used for calibrating the level of uncertainties (see Section 2.4). It should be noted that the experiments underestimate the resistance due to the fact that the diameter of manufactured samples tend to be larger than the ones used for numerical simulations. As a matter of fact, the drilled holes are larger that gives a lower resistance. Also, larger holes imply a higher reactance.

2.4 Probability model of model uncertainties and training dataset

2.4.1 Probability model of model uncertainties

As previously explained, the experimental dimensional dataset only gives consistent information about the statistical fluctuations and not about the mean values. Consequently, a probabilistic inference (Bayesian or likelihood methods, or learning constrained by experiments) cannot be used. Nevertheless, these experimental dimensional data are used for calibrating an additive noise (model uncertainties) to the ACM simulated data yielding the training dataset used by the probabilistic learning. In the proposed construction, the noise level is controlled by a hyperparameter $\delta^{\exp}$ (that will be the coefficient of variation).

This hyperparameter has to be estimated for any value of the control parameter $\mathbf{w}$, using the experimental dimensional dataset. Consequently, conditional statistics have to be used. However, the number of these experimental points is too small to obtain a good convergence of the estimate of δ^{exp}. We then have to resample the probability distribution of the experimental points using PLoM followed by conditional statistics. The steps of the construction are the following.

(i) The first step consists in using the probabilistic learning (the PLoM algorithm summarized in Appendix A) for which we have to define the random vector $\mathbf{X}^{\text{exp}}$ and its training dataset for the learning process. For all $\mathbf{w}$, we define the vectors in $\mathbb{R}^{n_\omega}$ such that $\mathbf{r}^{\text{exp}}(\mathbf{w}) = (\mathrm{r}^{\text{exp}}(\omega_1;\mathbf{w}),\ldots,\mathrm{r}^{\text{exp}}(\omega_{n_\omega};\mathbf{w}))$ and $\mathbf{v}^{\text{exp}} = (\mathrm{v}^{\text{exp}}(\omega_1;\mathbf{w}),\ldots,\mathrm{v}^{\text{exp}}(\omega_{n_\omega};\mathbf{w}))$ related to the experimental dimensional resistance and reactance, in which $\mathrm{r}^{\text{exp}}(\omega_k;\mathbf{w})$ and $\mathrm{v}^{\text{exp}}(\omega_k;\mathbf{w})$ are defined by Eqns. (2.2) and (2.3). Modeling the deterministic vector $\mathbf{w}$ by random variable $\mathbf{W}$ yields the $\mathbb{R}^{n_\omega}$-valued random variables $\mathbf{R}^{\text{exp}} = \mathbf{r}^{\text{exp}}(\mathbf{W})$ and $\mathbf{V}^{\text{exp}} = \mathbf{v}^{\text{exp}}(\mathbf{W})$. Let $\mathbf{Q}^{\text{exp}} = (\mathbf{R}^{\text{exp}}, \mathbf{V}^{\text{exp}})$ be the $\mathbb{R}^{n_\mathbf{q}}$-valued random variable. Let $\mathbb{R}^n = \mathbb{R}^{n_\mathbf{q}} \times \mathbb{R}^{n_\mathbf{w}}$ with $n = n_\mathbf{q} + n_\mathbf{w}$. Let $\mathbf{X}^{\text{exp}} = (\mathbf{Q}^{\text{exp}}, \mathbf{W})$ be the $\mathbb{R}^n$-valued random variable whose independent realizations are $\mathbf{x}^{\text{exp},j} = (\mathbf{q}^{\text{exp},j}, \mathbf{w}^j)$, $j = 1,\ldots,n_d$ with $\mathbf{w}^j \in \mathcal{C}_\mathbf{w}$, where $\mathbf{q}^{\text{exp},j}$ is the realization of $\mathbf{Q}^{\text{exp}}$, which is calculated by $\mathbf{q}^{\text{exp},j} = (\mathbf{r}^{\text{exp}}(\mathbf{w}^j), \mathbf{v}^{\text{exp}}(\mathbf{w}^j))$. The training dataset for $\mathbf{X}^{\text{exp}}$ is represented by the matrix $[x_d^{\text{exp}}] = [\mathbf{x}^{\text{exp},1}\ldots\mathbf{x}^{\text{exp},n_d}] \in \mathbb{M}_{n,n_d}$. For generating $n_{\text{ar}} = n_d \times n_{\text{MC}}$ learned realizations $\mathbf{x}_{\text{ar}}^{\text{exp},\ell}$, $\ell = 1,\ldots,n_{\text{ar}}$ of $\mathbf{X}^{\text{exp}}$ with $n_{\text{MC}} \gg 1$, the PLoM algorithm is used. We then deduced the n_{ar} learned realizations $(\mathbf{q}_{\text{ar}}^{\text{exp},\ell}, \mathbf{w}_{\text{ar}}^{\ell}) = \mathbf{x}_{\text{ar}}^{\text{exp},\ell}$ for all ℓ in $\{1,\ldots,n_{\text{ar}}\}$. As explained in Subsection 2.2.2, the database is split in two parts, one corresponding to $n_d = 18$ and the other to $n_d = 30$, referred to as $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. After the experimental dataset has been extracted using the methodology described in Subsection 2.3.1 a total of $18 + 30$ datapoints is obtained corresponding to control parameters of the finite set defined by Eqn. (2.1). This is not sufficient to have a good convergence of the vector-valued hyperparameter $\boldsymbol{\delta}^{\text{exp}}(\mathbf{w}_o)$ that has to be estimated using conditional statistics. Consequently, a resampling of the probability distribution of each experimental dataset $\mathcal{D}_1$ and $\mathcal{D}_2$ has been carried out using PLoM and adequate convergence has been obtained for $n_{\text{ar}} = 20\,000$ and $n_{\text{ar}} = 40\,000$, respectively.

(ii) For i in $\{1,\ldots,n_\mathbf{q}\}$, the experimental conditional coefficient of variation $\delta_i^{\text{exp}}(\mathbf{w}_o)$ of the component Q_i^{exp} of $\mathbf{Q}^{\text{exp}}$, given $\mathbf{W} = \mathbf{w}_o$ in $\mathcal{S}_\mathbf{w}$, is defined by

$$\delta_i^{\text{exp}}(\mathbf{w}_o) = \frac{\sigma_{Q_i^{\text{exp}}}(\mathbf{w}_o)}{m_{Q_i^{\text{exp}}}(\mathbf{w}_o)}\,, \tag{2.4}$$

in which $m_{Q_i^{\text{exp}}}(\mathbf{w}_o) = E\{Q_i^{\text{exp}}|\mathbf{W} = \mathbf{w}_o\}$ and $\sigma^2_{Q_i^{\text{exp}}}(\mathbf{w}_o) = E\{(Q_i^{\text{exp}})^2|\mathbf{W} = \mathbf{w}_o\} - m^2_{Q_i^{\text{exp}}}(\mathbf{w}_o)$ in which the conditional mathematical expectation are estimated using the learned realizations $\{(\mathbf{q}_{\text{ar}}^{\text{exp},\ell}, \mathbf{w}_{\text{ar}}^{\ell}), \ell = 1,\ldots,n_{\text{ar}}\}$ and the Gaussian Kernel Density Estimation (KDE) method (the integration being explicitly calculated, see Appendix B). We introduce the vector $\boldsymbol{\delta}^{\text{exp}}(\mathbf{w}_o) = (\delta_1^{\text{exp}},\ldots,\delta_{n_\mathbf{q}}^{\text{exp}})$.

(iii) The random vector $\mathbf{Q}^{\text{acm}}$ is transformed into a random vector $\mathbf{Q}$ whose components $\{Q_i, i = 1,\ldots,n_\mathbf{q}\}$ are written as

$$Q_i = (1 + B_i)Q_i^{\text{acm}} \quad , \quad i \in \{1,\ldots,n_\mathbf{q}\}\,, \tag{2.5}$$

in which $1 + B_i$ is a multiplicative random noise that allows model uncertainties to be taken into account.

Vector $\mathbf{Q}$, which represents the $\mathbf{W}$-parameterized and frequency-dependent random liner impedance, will be named the random quantity of interest (QoI). In order to construct the random variables $B_1, \ldots, B_{n_\mathbf{q}}$, we introduce a vector $\boldsymbol{\delta}(\mathbf{w}_o) = (\delta_1(\mathbf{w}_o), \ldots, \delta_{n_\mathbf{q}}(\mathbf{w}_o))$ in which $\delta_i(\mathbf{w}_o)$ is the coefficient of variation of the conditional random variable $1 + B_i$ given $\mathbf{W} = \mathbf{w}_o$. For the experiments, the parameters presenting variability are d/Δ, t/d, v_{or}, ρ/c, and L, in which $\Delta = \sqrt{\nu/\omega}$. It is assumed that the model uncertainties in the ACM are due to variability and uncertainties generated by the same parameters, which justifies the use of the experiments for generating the modeling errors. It should be noted that the components of vectors $\boldsymbol{\delta}(\mathbf{w}_o)$ and $\boldsymbol{\delta}^{\exp}(\mathbf{w}_o)$ represent the variations with respect to frequency ω for the resistance and reactance. We then choose for the frequency variations of $\boldsymbol{\delta}(\mathbf{w}_o)$, the frequency variations of $\boldsymbol{\delta}^{\exp}(\mathbf{w}_o)$. If the level of uncertainties were chosen equal for the experiments and for the model, then we would have $\boldsymbol{\delta}(\mathbf{w}_o) = \boldsymbol{\delta}^{\exp}(\mathbf{w}_o)$. However, we are interested in performing a sensitivity analysis with respect to the level of model uncertainties. We thus introduce a global parameter $a_{\mathrm{unc}} \in [0, 1]$ to quantify the level of uncertainties, and consequently, we write $\boldsymbol{\delta}(\mathbf{w}_o) = a_{\mathrm{unc}}\, \boldsymbol{\delta}^{\exp}(\mathbf{w}_o)$. Three values of a_{unc} will be considered: small uncertainties a_{unc} = 0.2, medium uncertainties a_{unc} = 0.5, and large uncertainties a_{unc} = 1.0. For all $i \in \{1, \ldots, n_\mathbf{q}\}$, the real-valued random variable B_i is thus defined by

$$B_i = \delta_i(\mathbf{W})\, 2\sqrt{3}\, (U_i - \frac{1}{2})\,, \tag{2.6}$$

in which $U_1, \ldots, U_{n_\mathbf{q}}$ are $n_\mathbf{q}$ independent real-valued random variable uniformly distributed on $[0, 1]$ and independent of $\mathbf{W}$. It can then be shown that B_i is a centered random variable and that coefficient of variation of the conditional random variable $1 + B_i$ given $\mathbf{W} = \mathbf{w}_o$ is $\delta_i(\mathbf{w}_o)$.

2.4.2 Training dataset including model uncertainties

To well represent the statistical fluctuations induced by model uncertainties in the training dataset, for each realization of $\mathbf{W}$ we will assign M_d realizations of the random variable $\mathbf{B}$. Consequently, for all i in $\{1, \ldots, n_\mathbf{q}\}$, the $N_d = n_d \times M_d$ realizations $\{\{q_i^{1,m}, \ldots, q_i^{n_d,m}\}, m = 1, \ldots, M_d\}$ of the random variable Q_i defined by Eqn. (2.5) are computed by the equation

$$q_i^{j,m} = (1 + b_i^{j,m})\, q_i^{\mathrm{acm},j}\,, \tag{2.7}$$

in which $b_i^{j,m}$ is the realization of B_i defined by Eqn. (2.6) and such that

$$b_i^{j,m} = \delta_i(\mathbf{w}^j)\, 2\sqrt{3}\, (u_i^{j,m} - \frac{1}{2})\,, \tag{2.8}$$

in which $\{\{u_i^{1,m}, \ldots, u_i^{n_d,m}\}, m = 1, \ldots, M_d\}$ are $N_d = n_d \times M_d$ independent realizations of random variable U_i. It should be noted that for m fixed $\{b_i^{j,m}, j = 1, \ldots, n_d\}$ are n_d independent realizations of B_i, but the realizations in the couple (j, m) are dependent. We then define the random vector $\mathbf{X} = (\mathbf{Q}, \mathbf{W})$ with values in $\mathbb{R}^n$, in which $\mathbf{Q}$ is defined by Eqn. (2.5), whose (j, m)-th realization is $\mathbf{x}^{j,m} = (\mathbf{q}^{j,m}, \mathbf{w}^{j,m})$ in which $\mathbf{q}^{j,m}$ is given by Eqn. (2.7) and where $\mathbf{w}^{j,m} = \mathbf{w}^j$ for all (j, m). For the probabilistic learning, the training dataset is represented by the matrix $[x_d] = [\mathbf{x}^{1,1} \ldots \mathbf{x}^{n_d,1}, \ldots, \mathbf{x}^{1,M_d} \ldots \mathbf{x}^{n_d,M_d}] \in \mathbb{M}_{n,N_d}$.

2.5 Construction of the statistical metamodel of the parameterized random liner impedance

2.5.1 Random manifold associated with the parameterized random liner impedance

The random QoI $\mathbf{Q}$ defined in Section 2.4.2 is written as $\mathbf{Q} = (\mathbf{R}, \mathbf{V})$ in which $\mathbf{R} = (R_1, \ldots, R_{n_\omega})$ and $\mathbf{V} = (V_1, \ldots, V_{n_\omega})$ are the $\mathbb{R}^{n_\omega}$-valued random variables representing the frequency-dependent resistance and reactance. This $\mathbb{R}^{n_\mathbf{q}}$-valued random variable $\mathbf{Q}$ can be written as $\mathbf{Q} = \mathbf{f}(\mathbf{W}, \mathbf{U})$ in which $\mathbf{W}$ is the random control parameter with values in $\mathbb{R}^{n_\mathbf{w}}$, $\mathbf{U}$ is the random uncontrolled parameter with values in $\mathbb{R}^{n_u}$ with $n_u = n_\mathbf{q}$ (see Eqn. (2.8)), and where $(\mathbf{w}, \mathbf{u}) \mapsto \mathbf{f}(\mathbf{w}, \mathbf{u})$ is a deterministic implicit mapping from $\mathbb{R}^{n_\mathbf{w}} \times \mathbb{R}^{n_u}$ into $\mathbb{R}^{n_\mathbf{q}}$. We then define the random mapping $\mathbf{w} \mapsto \mathbf{F}(\mathbf{w})$ such that for all $\mathbf{w}$ in $\mathbb{R}^{n_\mathbf{w}}$, $\mathbf{F}(\mathbf{w}) = \mathbf{f}(\mathbf{w}, \mathbf{U})$. The graph $\{(\mathbf{F}(\mathbf{w}), \mathbf{w}), \mathbf{w} \in \mathcal{S}_\mathbf{w} \subset \mathbb{R}^{n_\mathbf{w}}\}$ defines a random manifold in $\mathbb{R}^n = \mathbb{R}^{n_\mathbf{q}} \times \mathbb{R}^{n_\mathbf{w}}$, in which $n = n_\mathbf{q} + n_\mathbf{w}$. The random variable $\mathbf{X} = (\mathbf{Q}, \mathbf{W})$ with values in $\mathbb{R}^n$ is related to this random manifold because $\mathbf{X}$ can also be rewritten as $\mathbf{X} = (\mathbf{F}(\mathbf{W}), \mathbf{W})$. In Section 2.4.2, we have constructed the training set of $\mathbf{X}$ made up of N_d points $\mathbf{x}^j = (\mathbf{q}^j, \mathbf{w}^j) \in \mathbb{R}^n$, represented by matrix $[x_d] \in \mathbb{M}_{n,N_d}$.

2.5.2 Statistical metamodel

Taking into account the definition of random QoI $\mathbf{Q}$ defined in Section 2.5.1, the objective of the statistical metamodel of the parameterized random linear impedance is to construct the conditional probability distribution $P_{\mathbf{Q}|\mathbf{W}}(d\mathbf{q}|\mathbf{w}_o) = p_{\mathbf{Q}|\mathbf{W}}(\mathbf{q}|\mathbf{w}_o)\, d\mathbf{q}$ of $\mathbf{Q}$ given $\mathbf{W} = \mathbf{w}_o$ for any $\mathbf{w}_o$ in $\mathcal{S}_\mathbf{w} \subset \mathbb{R}^{n_\mathbf{w}}$, in which $p_{\mathbf{Q}|\mathbf{W}}$ is the conditional probability density function on $\mathbb{R}^{n_\mathbf{q}}$ given $\mathbf{W}$. This conditional probability distribution completely defines the statistical metamodel. However, we can only estimate this conditional probability distribution using the nonparametric statistics and sufficiently large set of realizations of $\mathbf{X} = (\mathbf{Q}, \mathbf{W})$. Since the training dataset of $\mathbf{X}$, represented by matrix $[\mathbf{x}_d]$, is constituted of a small number N_d of realizations, we will perform a resampling of $\mathbf{X}$ using a probabilistic learning on manifolds in order to generate a large number $N_{\text{ar}} \gg N_d$ of learned realizations of $\mathbf{X}$ (see Section 2.5.3). The connection between the probability density function of $\mathbf{X} = (\mathbf{Q}, \mathbf{W})$ and the conditional probability density function of interest is the following

$$p_{\mathbf{Q}|\mathbf{W}}(\mathbf{q}|\mathbf{w}_o) = \frac{1}{p_\mathbf{W}(\mathbf{w}_o)}\, p_{\mathbf{Q},\mathbf{W}}(\mathbf{q}, \mathbf{w}_o)\,, \tag{2.9}$$

in which $p_{\mathbf{Q},\mathbf{W}}$ is the joint probability density function on $\mathbb{R}^{n_\mathbf{q}} \times \mathbb{R}^{n_\mathbf{w}}$ of random variables $\mathbf{Q}$ and $\mathbf{W}$, and where $p_\mathbf{W}(\mathbf{w}_o) = \int_{\mathbb{R}^{n_\mathbf{q}}} p_{\mathbf{Q},\mathbf{W}}(\mathbf{q}, \mathbf{w}_o)\, d\mathbf{q}$ is the probability density function of $\mathbf{W}$ at point $\mathbf{w}_o \in \mathcal{S}_\mathbf{w}$.

From an engineering point of view, we are interested in deriving conditional statistics from conditional probability distribution $P_{\mathbf{Q}|\mathbf{W}}$, such as the conditional mean values and the conditional confidence regions. The results presented in Section 2.6 will be the conditional mathematical expectation,

$$E\{\mathbf{Q}|\mathbf{W} = \mathbf{w}_o\} = \int_{\mathbb{R}_{n_\mathbf{q}}} \mathbf{q}\, p_{\mathbf{Q}|\mathbf{W}}(\mathbf{q}|\mathbf{w}_o)\, d\mathbf{q}\,, \tag{2.10}$$

for any $\mathbf{w}_o$ in $\mathcal{S}_{\mathbf{W}}$. We will also present the conditional confidence region of the frequency-dependent random resistance $R_1, \ldots, R_{n_\omega}$ and reactance $V_1, \ldots, V_{n_\omega}$. For k fixed in $\{1, \ldots, n_\omega\}$, let Q_i be the component of $\mathbf{Q}$ representing either resistance R_k or reactance V_k. Then the lower bound q_i^- and the upper bound q_i^+ of the conditional confidence interval of Q_i given $\mathbf{W} = \mathbf{W}_o$ for a probability level p_c are defined by

$$q_i^+ : \mathrm{Proba}\{\mathrm{Q_i} \leq \mathrm{q_i^+} \,|\, \mathbf{W} = \mathbf{w_o}\} = \mathrm{p_c}\,, \tag{2.11}$$

$$q_i^- : \mathrm{Proba}\{\mathrm{Q_i^-} \leq \mathrm{q_i^-} \,|\, \mathbf{W} = \mathbf{w_o}\} = 1 - \mathrm{p_c}\,, \tag{2.12}$$

and where the probability in Eqns. (2.11) and (2.12) are calculated with the conditional cumulative distribution function,

$$\mathrm{Proba}\{\mathrm{Q_i} \leq \mathrm{q_i^*} \,|\, \mathbf{W} = \mathbf{w_o}\} = \int_{-\infty}^{\mathrm{q_i^*}} \mathrm{p_{Q_i|W}}(\mathrm{q_i^*} \,|\, \mathbf{w_o})\, \mathrm{dq_i}\,, \tag{2.13}$$

in which the conditional pdf $p_{Q_i|\mathbf{W}}$ is derived from conditional pdf $p_{\mathbf{Q}|\mathbf{W}}$ by an integration on $\mathbb{R}^{n_\mathbf{q}-1}$.

2.5.3 Generation of learned realizations to estimate the statistical metamodel

In order to estimate the conditional statistics defined by Eqns. (2.10) to (2.13) of the statistical metamodel, we need to generate the learned dataset constituted of a large number N_{ar} of learned realizations $(\mathbf{q}_{\mathrm{ar}}^\ell, \mathbf{w}_{\mathrm{ar}}^\ell)$ of random variable $(\mathbf{Q}, \mathbf{W})$ using the available information defined by the training dataset, represented by matrix $[x_d]$, for which the columns are the N_d points $x^j \in \mathbb{R}^n$. As previously explained, we need a probabilistic learning algorithm, and we propose to use the probabilistic learning on manifolds (PLoM) [21, 22, 23] for which the algorithm is summarized in the Appendix A. Once the learned realizations have been generated, the joint probability density function $p_{\mathbf{Q},\mathbf{W}}$ of $\mathbf{Q}$ and $\mathbf{W}$ is estimated using the multivariate Gaussian Kernel Density Estimation method. The resulting explicit expression allows for performing exact multiple integration with respect to coordinate vector $\mathbf{Q}$. We then obtain explicit algebraic expression for the estimate of Eqns. (2.10) to (2.13), which only depends on the learned dataset. These algebraic expressions are given in the Appendix B.

2.6 Results and discussion

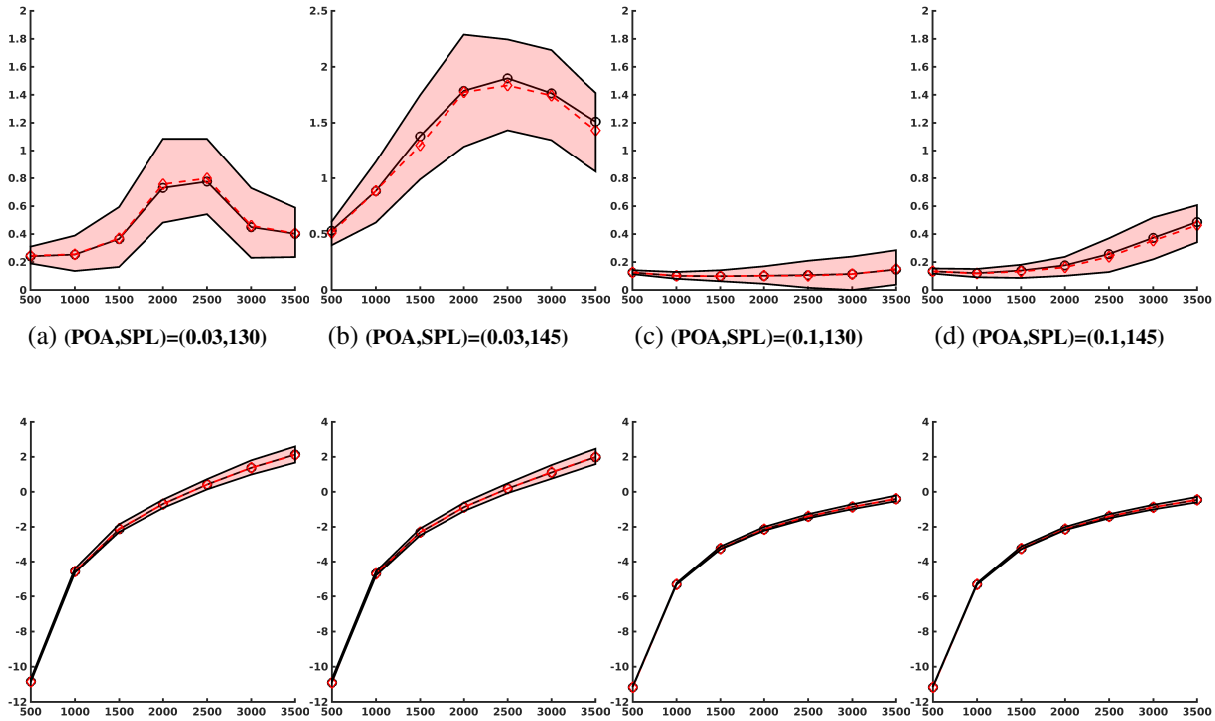
The conditional statistics defined in Section 2.5.2 are estimated using the learned dataset generated as explained in Section 2.5.3 with the following values of the parameters: n_d defined in Section 2.2.2, $M_d = 15$, $N_d = n_d \times M_d$, $N_{\mathrm{ar}} = N_d \times n_{\mathrm{MC}}$ with $n_{\mathrm{MC}} = 6000$, and $a_{\mathrm{unc}} = 0.5$. Two types of results are shown.

(i) The first one is related to the conditional statistics for which the control parameter $\mathbf{w}$ belongs to $\mathcal{C}_{\mathbf{W}}$ (the training dataset is based on these values).

(ii) For the second presented analysis, the control parameter does not belong to $\mathcal{C}_{\mathbf{w}}$ (and consequently, does not belong to the training dataset). Recalling that in the training dataset, the n_d points represent the values of resistance and the reactance as a function of the frequency and the control-parameters samples. Since n_d is very small, all these points have been kept to construct the training set, and it was not possible to keep a part of these points to perform a quality assessment (or cross-validation). So this second analysis has to be seen as predictions performed by the statistical metamodel, for which quality can only be evaluated by coherence.

2.6.1 Predictions of the statistical metamodel for which the control parameter belongs to the training dataset

For several values of the control parameters (POA, SPL) that belong to the training dataset, Fig. 2.6 shows the resistance (figures (a) to (d)) and the reactance (figures (e) to (h)) as a function of frequency. In each plot, it can be seen the curve corresponding to the ACM simulation data. On the other hand, using the training dataset that includes model uncertainties, the curve of the learning-based conditional mathematical expectation is plotted and the learning-based conditional confidence region for a probability level $p_c = 0.98$ is plotted.



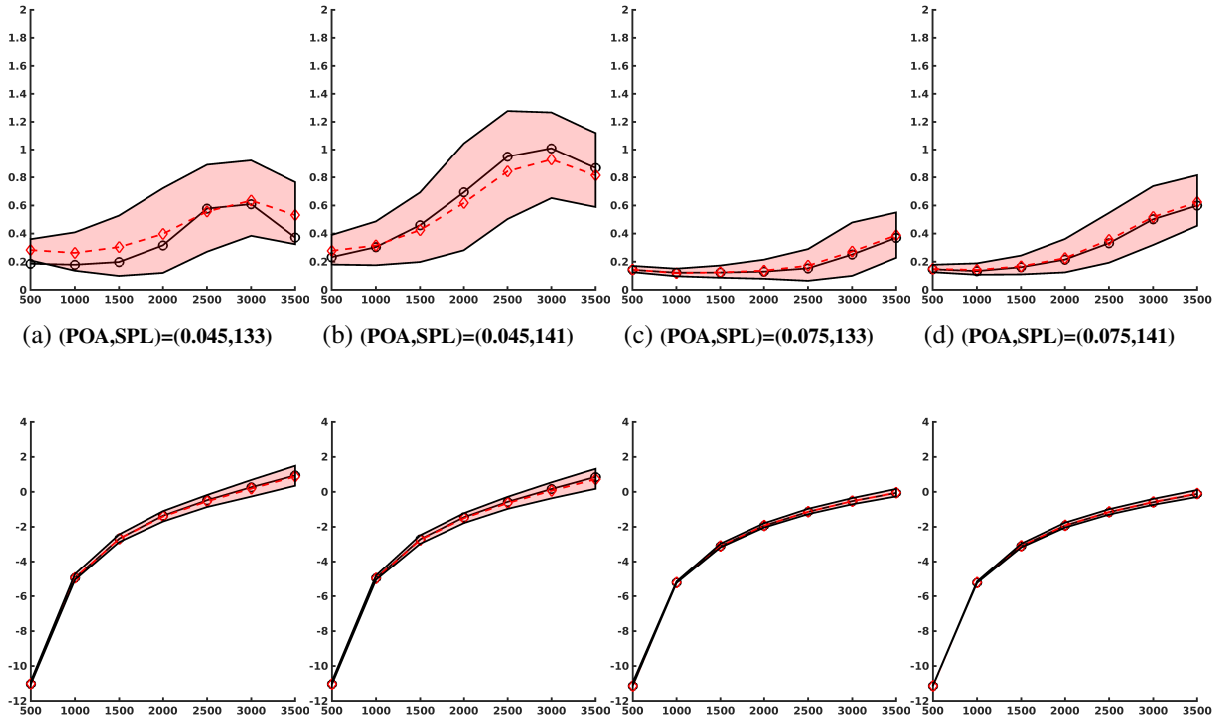
(a) **(POA,SPL)=(0.03,130)** (b) **(POA,SPL)=(0.03,145)** (c) **(POA,SPL)=(0.1,130)** (d) **(POA,SPL)=(0.1,145)**

(e) **(POA,SPL)=(0.03,130)** (f) **(POA,SPL)=(0.03,145)** (g) **(POA,SPL)=(0.1,130)** (h) **(POA,SPL)=(0.1,145)**

Figure 2.6: For several values of control parameters (POA, SPL) that belong to the training dataset, resistance (figures (a) to (d)) and reactance (figures (e) to (h)) (vertical axis) as a function of frequency (Hz) (horizontal axis). ACM simulation data (solid curves with circular markers). Using the training set including model uncertainties, learning-based conditional mathematical expectation (dashed curves with diamond markers) and 98% learning-based conditional confidence region (shaded domain).

2.6.2 Predictions of the statistical metamodel for which the control parameter does not belong to the training dataset

For this case, the nature of the presented results are the same as those presented in Subsection 2.6.1 but the control parameters (POA, SPL) do not belong to the training dataset. Fig. 2.7 shows the resistance (figures (a) to (d)) and the reactance (figures (e) to (h)) as a function of frequency.



(a) **(POA,SPL)=(0.045,133)** (b) **(POA,SPL)=(0.045,141)** (c) **(POA,SPL)=(0.075,133)** (d) **(POA,SPL)=(0.075,141)**

(e) **(POA,SPL)=(0.045,133)** (f) **(POA,SPL)=(0.045,141)** (g) **(POA,SPL)=(0.075,133)** (h) **(POA,SPL)=(0.075,141)**

Figure 2.7: For several values of control parameters (POA, SPL) that do not belong to the training dataset, resistance (figures (a) to (d)) and reactance (figures (e) to (h)) (vertical axis) as a function of frequency (Hz) (horizontal axis).

2.6.3 Discussion

It can be seen that the dispersion of the resistance is larger than the reactance. This means that the resistance is more sensitive to statistical fluctuations than the reactance. This dispersion is due to two factors. The first one is directly correlated to the contents of the training dataset without model uncertainties (the ACM simulation data), which contributes to the dispersion of the resistance. The second one is due to the model uncertainties that have been included in the training dataset. The analyses performed have shown that the contributions of the statistical fluctuations in the ACM simulation data is more dominant than those induced by the model uncertainties. For the considered values of the control parameters the dispersion of the reactance always stays small, while, the dispersion of the resistance strongly depends on the control-parameters values. It should be noted that, for each given value of the control parameter, the estimated conditional expectation of the resistance and reactance is close to the ACM simulation data.

This proximity is due to the fact the random variable **B** related to the model uncertainties is centered and, in addition, in the probabilistic learning process, the mean value corresponds to the ACM simulation data. Finally, Fig. 2.7 related to the second analysis, shows that the predictions performed by the statistical metamodel are coherent with respect to those shown in Fig. 2.6.

2.7 Conclusion

We have presented a methodology for constructing a robust PLoM-GKDE-based statistical metamodel of the acoustic liner impedance as a function of the frequency and on the control parameters, the POA and the SPL. In this work the Mach number has been chosen to be zero. It corresponds to an important configuration that has to be studied. For positive Mach number, the frequency evolutions and the amplitudes of the acoustic impedance are different and are analyzed in Chapter 4, using the presented methodology and also the methodology that will be presented in Chapter 3. The presented construction has been based on computationally expensive aeroacoustic model to generate simulated data that have yielded a small training dataset. Therefore, a probabilistic learning process has been used, and we have chosen the PLoM algorithm. Although the aeroacoustic simulation is conducted with a large aeroacoustic computational model, some approximations have been introduced, generating model errors. A probability model of these model errors has been developed to construct the training dataset. In order to calibrate the model errors, we have used dimensionless experiments available from the open literature. In addition, we have also introduced a sensitivity parameter to the level of model uncertainties. Despite the fact that we have very small amount of data, we have succeeded in proposing a robust statistical metamodel that is novel and whose predictions are consistent. This statistical framework has allowed for exhibiting a confidence region, which gives an information about the level of uncertainties about the acoustic liner impedance as a function of the frequency and the control parameters.

Chapter 3

Statistical metamodel based on an ANN formulation for Mach number fixed at zero

3.1 Introduction

The content of this chapter is taken from the paper [62] that has been submitted for publication and is devoted to the ANN-based statistical metamodel. Since this chapter is only devoted to the ANN-based statistical metamodel, we will often simply write "statistical metamodel" if there is no ambiguity.

3.1.1 Objectives

The objective of this chapter is to construct a statistical metamodel for which the outputs are the frequency-sampled impedance of the acoustic liner and the inputs are the control parameters that are the design parameters. The statistical metamodel must have a low-computational cost to enable its usage in its design optimization loop. In addition, the gradients of the statistical metamodel with respect to its inputs (the control parameters) must also have a low-computational cost.

In this chapter, we construct the ANN-based statistical metamodel using a dataset, referred to as the ACM dataset, which includes samples of control parameters and the corresponding acoustic impedance, numerically simulated by ACM. Consequently, the dataset is small due to the prohibitive computational cost of ACM, which prevents the construction of a large dataset.

As only the components of control parameter are inputs to the statistical metamodel, all other ACM parameters are unobserved (thus uncontrolled) and should be treated as random latent variables. Therefore, the acoustic liner impedance at any frequency should also be modeled as a random variable. A first novelty contribution presented in this chapter is the methodology for constructing such a statistical metamodel that is driven by the physics contained in the small ACM dataset.

3.1.2 Proposed methodology

Similar to the construction presented in chapter 2, the statistical metamodel is defined by the conditional probability distribution of the vector-valued random acoustic impedance, which is sampled in frequency, given the control parameter. The novelty presented in this chapter is the development of an ANN-based statistical metamodel in the following framework. The MaxEnt principle, applied with the available information, is used to construct an informative *prior* model of this conditional probability distribution. It should be noted that the hyperparameters of the probabilistic model depend on the values of the control parameter and are modeled using fully connected feedforward neural networks yielding a statistical ANN-based metamodel. Such a statistical metamodel is fitted on an *ad hoc* training dataset using the maximum likelihood principle.

A principal component analysis (PCA) is conducted on the outputs (frequency-sampled acoustic impedance) of the statistical metamodel. This is not only for potential statistical reduction but also because decorrelation and centering of outputs enhance numerical conditioning, thereby facilitating the optimization process for fitting the ANN to the training dataset. However, such a statistical decorrelation and centering of the outputs introduce constraints on the hyperparameters of the statistical metamodel and consequently, on the parameters the ANN. Therefore, given that these hyperparameters are modeled by fully connected feedforward networks, some deterministic constraints must be considered, yielding the development of constrained training algorithm for such networks. Such a constrained training algorithm is challenging and complex when dealing with mini-batches, and might rely on techniques such as Lagrange multipliers, penalization approaches, correction formulations, etc. We therefore present an unconstrained formulation for the ANN that takes into account the statistical constraints arising from the PCA-based reduction. This is achieved by generating an *ad hoc* training dataset using the PLoM-GKDE-based statistical metamodel presented in Chapter 2. Finally, the statistical metamodel can be used to generate additional realizations of the frequency-sampled acoustic impedance vector, thereby mitigating missing data in the set of control parameters.

3.1.3 Organization of the chapter

The chapter is organized as follows. Section 3.2 briefly defines the control parameters and the ACM for calculating the frequency-sampled vector of acoustic impedance, which is to be used for constructing the ACM dataset (small size). Section 3.3 is dedicated to the parametric probabilistic modeling of the conditional probability distribution of the frequency-sampled vector of acoustic impedance given the control parameter. Section 3.4 focuses on the statistical ANN-based metamodel given the control parameter. Section 3.5 presents a numerical example, along with a discussion of the results.

3.2 Control parameters and ACM dataset

A part of this section is duplicated from Section 2.2 in order to facilitate the reading.

In this chapter, the considered system is an acoustic liner consisting of a perforated plate, a honeycomb

structure, and a rigid backing plate, as depicted in Fig. 2.1. The Mach number is assumed to be equal to zero, that is relevant for the ground configuration. Incorporating non-zero Mach numbers require additional analyses beyond the scope of this chapter. A reduced domain, as described in [8], is used for the computational model (see Fig. 2.2). In this chapter and for sake of simplicity, the liner system is parameterized by $n_{\mathbf{w}} = 2$ parameters (the control parameters) that are the Percentage of Open Area (POA) and the Sound Pressure Level (SPL), represented by $\mathbf{w} = (w_1, w_2)$, where w_1 corresponds to POA and w_2 to SPL.

For such a reduced domain, the frequency-dependent acoustic impedance is denoted as $\omega \mapsto \mathbf{z}^{\,\mathrm{acm}}(\omega; \mathbf{w})$, where ω is the frequency (rad/s). Specifically, $\mathbf{z}^{\,\mathrm{acm}}(\omega; \mathbf{w}) = r^{\,\mathrm{acm}}(\omega; \mathbf{w}) + \imath\, v^{\,\mathrm{acm}}(\omega; \mathbf{w})$, in which $r^{\,\mathrm{acm}}(\omega; \mathbf{w})$ represents the resistance that is positive, $v^{\,\mathrm{acm}}(\omega; \mathbf{w})$ the reactance that is real, $\imath = \sqrt{-1}$. Control parameter $\mathbf{w}$ belongs to an admissible set. The computational domain is centered around the resonator, within which the Navier-Stokes equations are solved. The computational model consists of 278 514 degrees of freedom. For given control parameter $\mathbf{w}$, and for each sampled frequency $\omega_k = (k-1)\,\Delta\omega$ with $k = 1, \ldots, n_{\mathbf{w}}$ where $n_\omega = 7$, the ACM computes the resistance $r^{\,\mathrm{acm}}(\omega_k; \mathbf{w})$ and the reactance $v^{\,\mathrm{acm}}(\omega_k; \mathbf{w})$. The Navier-Stokes equations are solved using the numerical method presented in [8] for $\mathbf{w} = (w_1, w_2) \in [0.03, 0.1] \times [130, 145]$. As discussed in Chapter 2, ACM simulations are performed for $n_d = 48$ values $\mathbf{w}^1, \ldots, \mathbf{w}^{n_d}$ of $\mathbf{w}$, which constitute the set $\mathcal{D}_{\mathbf{w}}$. These points in $\mathcal{D}_{\mathbf{w}}$ are considered as realizations of a random vector $\mathbf{W}$ whose probability density function, $p_{\mathbf{W}}$, is unknown.

The ACM dataset $\mathcal{D}_{\mathrm{acm}}$ is then defined as the set of points $(\mathbf{w}^j, \mathbf{r}^{\,\mathrm{acm},j}, \mathbf{v}^{\,\mathrm{acm},j})$ in $\mathbb{R}^{n_{\mathbf{w}} \times n_\omega \times n_\omega}$ for $j = 1, \ldots, n_d$, in which $\mathbf{r}^{\,\mathrm{acm},j} = (r^{\,\mathrm{acm}}(\omega_1;\, \mathbf{w}^j), \ldots, r^{\,\mathrm{acm}}(\omega_{n_\omega};\, \mathbf{w}^j))$ and $\mathbf{v}^{\,\mathrm{acm},j} = (v^{\,\mathrm{acm}}(\omega_1;\, \mathbf{w}^j), \ldots, v^{\,\mathrm{acm}}(\omega_{n_\omega};\, \mathbf{w}^j))$. For each $j = 1, \ldots, n_d$, we introduce the $2\,n_\omega$-dimensional vector of frequency-sampled impedance $\mathbf{q}^{\,\mathrm{acm},j} = (\log \mathbf{r}^{\,\mathrm{acm},j},\, \mathbf{v}^{\,\mathrm{acm},j})$ in which $\log \mathbf{r}^{\,\mathrm{acm},j} = (\log r^{\,\mathrm{acm}}(\omega_1;\, \mathbf{w}^j), \ldots, \log r^{\,\mathrm{acm}}(\omega_{n_\omega};\, \mathbf{w}^j))$.

The model uncertainties are due to random latent parameters that, consequently, cannot be defined as control parameters. In order to take into account these model uncertainties, the random vectors $\mathbf{Q}, \mathbf{R}$, and $\mathbf{V}$ are introduced whose conditional probability density functions, given $\mathbf{W} = \mathbf{w}^j$, are denoted by $\mathbf{q} \mapsto p_{\mathbf{Q}}(\mathbf{q}|\, \mathbf{w}^j)$, $p_{\mathbf{R}}(\mathbf{r}|\, \mathbf{w}^j)$, and $p_{\mathbf{V}}(\mathbf{v}|\, \mathbf{w}^j)$. These conditional probability density functions are constructed as explained in Chapter 2. For instance, the conditional mean value of $\mathbf{Q}$ given $\mathbf{W} = \mathbf{w}^j$ is chosen as $\mathbf{q}^{\,\mathrm{acm},j}$ for $j = 1, \ldots, n_d$; the conditional dispersion coefficient given $\mathbf{W} = \mathbf{w}^j$ has been identified using experimental data. We then generate $m_d = 15$ statistically independent realizations $\mathbf{q}^{j,1}, \ldots, \mathbf{q}^{j,m_d}$ from $p_{\mathbf{Q}|\mathbf{W}}$ given $\mathbf{W} = \mathbf{w}^j$. Hence, training dataset $\mathcal{D}^*_{\mathrm{acm}}$ is made up of a total of $n_{\,\mathrm{acm}} = n_d \times m_d$ realizations $(\mathbf{w}^{j,k}, \mathbf{q}^{j,k})$ with $j = 1, \ldots, n_d$ and $k = 1, \ldots, m_d$, in which $\mathbf{w}^{j,k}$ is a rewriting of $\mathbf{w}^j$ that is independent of k (we introduce a repetition). For the sake of simplicity, all the realizations are rewritten as $(\mathbf{w}^j,\, \mathbf{q}^j)$ with $j = 1, \ldots, n_{\,\mathrm{acm}}$.

In Chapter 2, the PLoM (Probabilistic Learning on Manifold) is carried out to learn the joint probability density function $p_{\mathbf{Q},\,\mathbf{W}}$ of random vectors $\mathbf{Q}$ and $\mathbf{W}$ using $\mathcal{D}^*_{\mathrm{acm}}$ as a training dataset. PLoM also allows n_{ar} additional statistically independent realizations $\{(\mathbf{w}^\ell_{\mathrm{ar}},\, \mathbf{q}^\ell_{\mathrm{ar}})\,,\, \ell = 1, \ldots, n_{\mathrm{ar}}\}$ to be generated, which constitute the learned dataset $\mathcal{D}^*_{\mathrm{ar}}$.

3.3 Prior probabilistic model of the frequency-sampled impedance vector

In Chapter 2, the conditional probability density function $p_{\mathbf{Q}|\mathbf{W}}$ is estimated with the Gaussian Kernel Density Estimation (GKDE) using additional realizations (generated by PLoM) of $\mathbf{Q}$ and $\mathbf{W}$. In the context of the construction of a statistical metamodel based on a neural network, we need to introduce an algebraic representation of the conditional probability distribution of $\mathbf{Q}$ given $\mathbf{W}$, depending on hyperparameters. In this chapter we have chosen a Gaussian model for which the hyperparameters are conditional mean value and the conditional covariance matrix of $\mathbf{Q}$ given $\mathbf{W}$. The neural network will be used for predicting these conditional hyperparameters. In this section, we then present the construction of the algebraic *prior* probabilistic model of $p_{\mathbf{Q}|\mathbf{W}}$. Nevertheless, since $\mathbf{Q}$ is in high dimension, we will introduce a statistical reduction $\mathbf{H}$ of $\mathbf{Q}$ using a PCA. Within Section 3.3.2, the conditional hyperparameters associated with such a *prior* probabilistic model are represented as functions of $\mathbf{w}$. The modeling of these functions is carried out using fully connected feedforward networks that are trained to map control parameter $\mathbf{w}$ onto a corresponding set of hyperparameters of the *prior* probabilistic model.

3.3.1 PCA-based statistical reduction H of Q

A PCA is used to construct the statistical reduction of $\mathbf{Q}$, yielding a normalized random vector $\mathbf{H}$ (centered with identity covariance matrix). Random vector $\mathbf{H}$ is therefore written as $\mathbf{H} = [\lambda]^{-1/2}\,[\phi]^T\,(\mathbf{Q} - \underline{\mathbf{q}})$, in which $\underline{\mathbf{q}} = (1/n_{\text{ar}})\sum_{\ell=1}^{n_{\text{ar}}} \mathbf{q}_{\text{ar}}^{\ell}$ is the empirical mean value of random vector $\mathbf{Q}$, $[\lambda]$ is a $(m \times m)$ diagonal matrix, and $[\phi]$ is a $(n_{\mathbf{q}} \times m)$ matrix whose columns are orthonormal vectors, and are such that $[C_{\mathbf{Q}}][\phi] = [\phi][\lambda]$. The estimate of the $(n_{\mathbf{q}} \times n_{\mathbf{q}})$ covariance matrix of $\mathbf{Q}$ is $[C_{\mathbf{Q}}] = (n_{\text{ar}}-1)^{-1}\sum_{\ell=1}^{n_{\text{ar}}}(\mathbf{q}_{\text{ar}}^{\ell}-\underline{\mathbf{q}})\,(\mathbf{q}_{\text{ar}}^{\ell}-\underline{\mathbf{q}})^T$. The diagonal entries of $[\lambda]$ are the m-largest eigenvalues of $[C_{\mathbf{Q}}]$. By construction, the $\mathbb{R}^m$-valued random variable $\mathbf{H}$ is such

$$E\{\mathbf{H}\} = 0_m \quad , \quad E\{\mathbf{H} \otimes \mathbf{H}\} = [\mathrm{I}_m]\,. \tag{3.1}$$

3.3.2 Prior conditional probabilistic density function of H given W

Let $\boldsymbol{\eta} \mapsto p_{\mathbf{H}|\mathbf{W}}(\boldsymbol{\eta}|\mathbf{w})$ be the conditional probability density function of $\mathbf{H}$ given $\mathbf{W}$. The *prior* conditional probability density function $p_{\mathbf{H}|\mathbf{W}}$ is constructed using the MaxEnt principle (see for instance [57]) with the following available information: (1) the support of $\boldsymbol{\eta} \mapsto p_{\mathbf{H}|\mathbf{W}}(\boldsymbol{\eta}|\mathbf{w})$ is $\mathbb{R}^m$, (2) the conditional mean value and the conditional covariance matrix of $\mathbf{H}$ given $\mathbf{W} = \mathbf{w}$ are the vector $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$ and the matrix $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$, which are estimated using the training dataset $\mathcal{D}_{\text{ar}}^{*}$ for each given value of $\mathbf{w}$. Therefore, $\mathbf{H}$ given $\mathbf{W} = \mathbf{w}$ is a multivariate Gaussian random variable with mean value $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$ and covariance matrix $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$.

For any given $\mathbf{w}$ in its admissible set, the estimate of hyperparameters $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$ and $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$, constructed using GKDE from nonparametric statistics and dataset $\mathcal{D}_{\text{ar}}^{*}$, are written as

$$\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}) = \frac{\sum_{\ell=1}^{n_{\text{ar}}} \boldsymbol{\eta}_{\text{ar}}^{\ell} \exp\left(-\frac{1}{2s^2}(\mathbf{w}-\mathbf{w}_{\text{ar}}^{\ell})^T\,[C_{\mathbf{W}}]^{-1}\,(\mathbf{w}-\mathbf{w}_{\text{ar}}^{\ell})\right)}{\sum_{\ell=1}^{n_{\text{ar}}} \exp\left(-\frac{1}{2s^2}(\mathbf{w}-\mathbf{w}_{\text{ar}}^{\ell})^T\,[C_{\mathbf{W}}]^{-1}\,(\mathbf{w}-\mathbf{w}_{\text{ar}}^{\ell})\right)}\,, \tag{3.2}$$

$$[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})] = \frac{\sum_{\ell=1}^{n_{\text{ar}}} \widetilde{\boldsymbol{\eta}}_{\text{ar}}^{\ell}(\mathbf{w}) \, (\widetilde{\boldsymbol{\eta}}_{\text{ar}}^{\ell}(\mathbf{w}))^T \exp\left(-\frac{1}{2s^2}(\mathbf{w}-\mathbf{w}_{\text{ar}}^{\ell})^T \, [C_{\mathbf{W}}]^{-1} \, (\mathbf{w}-\mathbf{w}_{\text{ar}}^{\ell})\right)}{\sum_{\ell=1}^{n_{\text{ar}}} \exp\left(-\frac{1}{2s^2}(\mathbf{w}-\mathbf{w}_{\text{ar}}^{\ell})^T \, [C_{\mathbf{W}}]^{-1} \, (\mathbf{w}-\mathbf{w}_{\text{ar}}^{\ell})\right)} \, , \tag{3.3}$$

where (1) the $(n_{\mathbf{w}} \times n_{\mathbf{w}})$ matrix $[C_{\mathbf{W}}] = (n_{\text{ar}}-1)^{-1} \sum_{\ell=1}^{n_{\text{ar}}} (\mathbf{w}_{\text{ar}}^{\ell} - \underline{\mathbf{w}}) \, (\mathbf{w}_{\text{ar}}^{\ell} - \underline{\mathbf{w}})^T$ is the estimate of the covariance matrix of $\mathbf{W}$ in which $\underline{\mathbf{w}} = (1/n_{\text{ar}}) \sum_{\ell=1}^{n_{\text{ar}}} \mathbf{w}_{\text{ar}}^{\ell}$ is the estimate of the mean value of $\mathbf{W}$; (2) for all $\ell = 1, \ldots, n_{\text{ar}}$, we have $\boldsymbol{\eta}_{\text{ar}}^{\ell} = [\lambda]^{-1/2}[\phi]^T(\mathbf{q}_{\text{ar}}^{\ell} - \underline{\mathbf{q}})$ and $\widetilde{\boldsymbol{\eta}}_{\text{ar}}^{\ell}(\mathbf{w}) = \boldsymbol{\eta}_{\text{ar}}^{\ell} - \boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$; (3) s is the Silverman bandwidth given by

$$s = \left\{ \frac{4}{n_{\text{ar}}(2+n)} \right\}^{1/(n+4)} \quad , \quad n = m + n_{\mathbf{w}} \tag{3.4}$$

Due to Eqn (3.1), $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$ and $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$ have to satisfy the following equations,

$$E\{\mu_{\mathbf{H}|\mathbf{W}}(\mathbf{W})\} = 0_m \quad , \tag{3.5}$$

$$E\left\{ [C_{\mathbf{H}|\mathbf{W}}(\mathbf{W})] + \boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{W}) \boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{W})^T \right\} = [I_m] \quad . \tag{3.6}$$

With the proposed methodology, these two equations will automatically be satisfied.

3.3.3 Statistically independent realizations of R and V given W

For given $\mathbf{w}$, let $\mathbf{R}(\mathbf{w})$ and $\mathbf{V}(\mathbf{w})$ be the $\mathbb{R}^{n_\omega}$-valued random variables defined in Section 3.2. For any given $\mathbf{w}$ in its admissible set, N statistically independent realizations $\boldsymbol{\eta}^1(\mathbf{w}), \ldots, \boldsymbol{\eta}^N(\mathbf{w})$ of $\mathbf{H}$ given $\mathbf{W} = \mathbf{w}$ are generated using the multivariate Gaussian random distribution whose mean value is $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$ and covariance matrix is $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$. We then deduce N statistically independent realizations $\mathbf{q}^1(\mathbf{w}), \ldots, \mathbf{q}^N(\mathbf{w})$ of random vector $\mathbf{Q}$ given $\mathbf{W} = \mathbf{w}$ such that, for $j = 1, \ldots, N$, $\mathbf{q}^j(\mathbf{w}) = \underline{\mathbf{q}} + [\Phi] \, [\lambda]^{1/2} \, \boldsymbol{\eta}^j(\mathbf{w})$. For $j = 1, \ldots, N$, the block decomposition of vector $\mathbf{q}^j(\mathbf{w})$ is written as $(\mathbf{q}_{\mathbf{R}}^j(\mathbf{w}), \, \mathbf{v}^j(\mathbf{w}))$ with $\mathbf{q}_{\mathbf{R}}^j$ and $\mathbf{v}^j(\mathbf{w})$ being two n_ω dimensional vectors. Obviously, $\mathbf{v}^1(\mathbf{w}), \ldots, \mathbf{v}^N(\mathbf{w})$ are statistically independent realizations of $\mathbf{V}(\mathbf{w})$ and the statistically independent realizations $\mathbf{r}^1(\mathbf{w}), \ldots, \mathbf{r}^N(\mathbf{w})$ of $\mathbf{R}(\mathbf{w})$ are such that $\mathbf{q}_{\mathbf{R}}^j(\mathbf{w}) = \log(\mathbf{r}^j(\mathbf{w}))$. Note that, for given $\mathbf{w}$, conditional mean vectors $\boldsymbol{\mu}_{\mathbf{R}|\mathbf{W}}(\mathbf{w})$ and $\boldsymbol{\mu}_{\mathbf{V}|\mathbf{W}}(\mathbf{w})$, and the conditional covariance matrices $[C_{\mathbf{R}|\mathbf{W}}(\mathbf{w})]$ and $[C_{\mathbf{V}|\mathbf{W}}(\mathbf{w})]$ are estimated using the statistically independent realizations $\mathbf{r}^j(\mathbf{w}), \ldots, \mathbf{r}^N(\mathbf{w})$ and $\mathbf{v}^j(\mathbf{w}), \ldots, \mathbf{v}^N(\mathbf{w})$. Note that vector $\boldsymbol{\mu}_{\mathbf{R}|\mathbf{W}}(\mathbf{w})$ and matrix $[C_{\mathbf{R}|\mathbf{W}}(\mathbf{w})]$ can also be written as

$$\boldsymbol{\mu}_{\mathbf{R}|\mathbf{W}}(\mathbf{w}) = \exp\left(\boldsymbol{\mu}_{\mathbf{Q}_{\mathbf{R}}|\mathbf{W}}(\mathbf{w}) + \frac{1}{2} \operatorname{diag}[C_{\mathbf{Q}_{\mathbf{R}}|\mathbf{W}}(\mathbf{w})] \right) \quad , \tag{3.7}$$

$$[C_{\mathbf{R}|\mathbf{W}}(\mathbf{w})] = \exp\left([C_{\mathbf{Q}_{\mathbf{R}}|\mathbf{W}}(\mathbf{w})] - 1 \right) \odot \left(\boldsymbol{\mu}_{\mathbf{R}|\mathbf{W}}(\mathbf{w}) \, \boldsymbol{\mu}_{\mathbf{R}|\mathbf{W}}(\mathbf{w})^T \right) , \tag{3.8}$$

where $[A] \odot [B]$ stands for Hadamard product of matrices $[A]$ and $[B]$; $\exp([A])$ is the element-wise exponential and $\operatorname{diag}[A]$ is the vector made up of the diagonal entries of a given matrix $[A]$. The block decomposition of $\boldsymbol{\mu}_{\mathbf{Q}|\mathbf{W}}(\mathbf{w}) = \underline{\mathbf{q}} + [\Phi] \, [\lambda]^{1/2} \, \boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$ into n_ω dimensional vectors $\boldsymbol{\mu}_{\mathbf{Q}_{\mathbf{R}}|\mathbf{W}}(\mathbf{w})$ and $\boldsymbol{\mu}_{\mathbf{V}|\mathbf{W}}(\mathbf{w})$, is written as

$$\boldsymbol{\mu}_{\mathbf{Q}|\mathbf{W}}(\mathbf{w}) = (\boldsymbol{\mu}_{\mathbf{Q}_{\mathbf{R}}|\mathbf{W}}(\mathbf{w}), \, \boldsymbol{\mu}_{\mathbf{V}|\mathbf{W}}(\mathbf{w})) \, . \tag{3.9}$$

The block decomposition of matrix $[C_{\mathbf{Q}|\mathbf{W}}(\mathbf{w})] = [\Phi]\,[\lambda]^{1/2}\,[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]\,[\lambda]^{1/2}\,[\Phi]^T$ into $(n_\omega \times n_\omega)$ matrices $[C_{\mathbf{Q_R}|\mathbf{W}}(\mathbf{w})]$, $[C(\mathbf{w})]$, and $[C_{\mathbf{V}|\mathbf{W}}(\mathbf{w})]$, is written as

$$[C_{\mathbf{Q}|\mathbf{W}}(\mathbf{w})] = \begin{pmatrix} [C_{\mathbf{Q_R}|\mathbf{W}}(\mathbf{w})] & [C(\mathbf{w})] \\ \\ [C(\mathbf{w})] & [C_{\mathbf{V}|\mathbf{W}}(\mathbf{w})] \end{pmatrix}. \tag{3.10}$$

Concerning the random frequency-sampled vector $\mathbf{V}(\mathbf{w})$, vector $\boldsymbol{\mu}_{\mathbf{V}|\mathbf{W}}(\mathbf{w})$ and matrix $[C_{\mathbf{V}|\mathbf{W}}(\mathbf{w})]$ are directly obtained from block decomposition of vector $\boldsymbol{\mu}_{\mathbf{Q}|\mathbf{W}}(\mathbf{w})$ and matrix $[C_{\mathbf{Q}|\mathbf{W}}(\mathbf{w})]$ given by Eqns. (3.9) and (3.10).

3.4 ANN-based statistical metamodel

3.4.1 Fully connected feedforward neural network

Deterministic mappings $\mathbf{w} \mapsto \mu_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$ and $\mathbf{w} \mapsto [C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$ may have a complex behavior, not only because their supports are multidimensional, but also because the underlying physical process is complex. In such a case, fully connected feedforward neural network is well adapted to represent such deterministic mappings. We then consider a fully connected feedforward neural network $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}\ \boldsymbol{\theta}_1)$ with parameter $\boldsymbol{\theta}_1$, which is constructed in order to model deterministic mapping $\mathbf{w} \mapsto \mu_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$. However, a representation of $\mathbf{w} \mapsto [C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$ by a fully connected feedforward network is not straightforward because, for each given $\mathbf{w}$, the output has to be a positive-definite matrix. To circumvent this apparent difficulty, the matrix logarithm of the $(m \times m)$ symmetric covariance matrix $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$ is calculated for each given $\mathbf{w}$, which yields a $(m \times m)$ symmetric matrix $[\log C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$. If all the $m(m+1)/2$ entries of the upper triangular block of matrix $[\log C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$ are collected into the $m(m+1)/2$ dimensional vector $\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$ then, a fully connected feedforward network $\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}\ \boldsymbol{\theta}_2)$ with parameter $\boldsymbol{\theta}_2$ is constructed for modeling the deterministic mapping $\mathbf{w} \mapsto \boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$. Then, for a given $\mathbf{w}$, the output of $\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\ \boldsymbol{\theta}_2)$ with parameter $\boldsymbol{\theta}_2$ is used in order to assemble matrix $[\log C_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\ \boldsymbol{\theta}_2)]$. Then, matrix $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\ \boldsymbol{\theta}_2)]$ is calculated as the matrix exponential of $[\log C_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\ \boldsymbol{\theta}_2)]$. We then defined as the statistical ANN-based metamodel, the probabilistic model in Section 3.3 where $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\ \boldsymbol{\theta}_1)$ and $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\ \boldsymbol{\theta}_2)]$ are used for modelling $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w})$ and $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$. Consequently, the statistical ANN-based model is defined as the probability density function $\boldsymbol{\eta} \mapsto p_{\mathbf{H}|\mathbf{W}}(\boldsymbol{\eta}|\mathbf{w};\ \boldsymbol{\theta}_1,\ \boldsymbol{\theta}_2)$ that is a multivariate Gaussian probability density function with mean value $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\ \boldsymbol{\theta}_1)$ and covariance matrix $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\ \boldsymbol{\theta}_2)]$.

3.4.2 ANN for regression with the learned dataset $\mathcal{D}^*_{\text{ar}}$

As it is usually the case for most regression problems, parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are adjusted by fitting the ANN to a suitable training dataset. In this chapter, such fitting is carried out in minimizing with respect to $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ the negative-log-likelihood $\mathcal{L}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ of the ANN. Obviously, $\mathcal{D}^*_{\text{acm}}$ and $\mathcal{D}_{\text{acm}}$ are not suitable as training datasets for such fitting process since their sizes are small (less than one hundred elements). It is well known that, in the framework of ANN modeling (as well for deterministic ANN modeling), small

training datasets yields *overfitting* models that are unable to predict their targets well enough for unseen values of their inputs. On the other hand, since the size n_{ar} of learned dataset $\mathcal{D}^*_{\text{ar}}$ is as large as needed, such dataset is completely suitable as a training dataset. Using learned dataset $\mathcal{D}^*_{\text{ar}}$, the negative-log-likelihood to be minimized is written as

$$\mathcal{L}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = -\sum_{\ell=1}^{n_{\text{ar}}} \log\, p_{\mathbf{H},\,\mathbf{W}}(\boldsymbol{\eta}^{\ell}_{\text{ar}},\, \mathbf{w}^{\ell}_{\text{ar}};\, \boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2) \quad , \tag{3.11}$$

$$= -\sum_{\ell=1}^{n_{\text{ar}}} \log\, \Big(p_{\mathbf{H}|\mathbf{W}}(\boldsymbol{\eta}^{\ell}_{\text{ar}}|\mathbf{w}^{\ell}_{\text{ar}};\, \boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2)\, p_{\mathbf{W}}(\mathbf{w}^{\ell}_{\text{ar}}) \Big) \quad , \tag{3.12}$$

$$= -\sum_{\ell=1}^{n_{\text{ar}}} \log\, p_{\mathbf{H}|\mathbf{W}}(\boldsymbol{\eta}^{\ell}_{\text{ar}}|\mathbf{w}^{\ell}_{\text{ar}};\, \boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2) - \sum_{\ell=1}^{n_{\text{ar}}} \log p_{\mathbf{W}}(\mathbf{w}^{\ell}_{\text{ar}}) \quad . \tag{3.13}$$

Minimizing $\mathcal{L}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ with respect to $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ is equivalent to minimizing the cost function $J(\boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2)$ defined by

$$J(\boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2) = \frac{1}{2}\sum_{\ell=1}^{n_{\text{ar}}} \log\, \Big(\det\, [C_{\mathbf{H}|\mathbf{W}}(\boldsymbol{\eta}^{\ell}_{\text{ar}};\, \boldsymbol{\theta}_2)] \Big) + \frac{1}{2}\sum_{\ell=1}^{n_{\text{ar}}} \big(\boldsymbol{\eta}^{\ell}_{\text{ar}} - \boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}^{\ell}_{\text{ar}};\, \boldsymbol{\theta}_1)\big)^T [C_{\mathbf{H}|\mathbf{W}}(\mathbf{w}^{\ell}_{\text{ar}};\, \boldsymbol{\theta}_2)]^{-1} \big(\boldsymbol{\eta}^{\ell}_{\text{ar}} - \boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}^{\ell}_{\text{ar}};\, \boldsymbol{\theta}_1)\big) \,. \tag{3.14}$$

Note that such minimization should be performed under the constraints defined by Eqns. (3.5) and (3.6). Classically, such constraints would be taken into account introducing Lagrange multipliers [63, 64, 65], augmented Lagrangian [64, 65], penalty methods [63, 64, 65], barrier methods [65], projected gradient methods [66] or Sequential Quadratic Programming [65]. It is not straightforward to implement such methods in the framework of fully connected feedforward neural network training when mini-batch are required due to constraints on RAM availability due to CPU or GPU limitations. In this chapter and as explained in Section 3.4.3, we take advantage of the PLoM in order to fit the probabilistic model by adjusting $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ such that constraints defined by Eqns. (3.5) and (3.6) are automatically satisfied.

3.4.3 ANN for regression with a learned GKDE-based estimates dataset

It should be noted that minimizing $J(\boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2)$ with respect to $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ under constraints defined by Eqns. (3.5) and (3.6) is equivalent to construct the likelihood-based statistical estimators of conditional mean and conditional covariance matrix of $\mathbf{H}$ given $\mathbf{W} = \mathbf{w}$. Such statistical estimators are different from the GKDE-based statistical estimators defined by Eqns. (3.2) and (3.3). Therefore, in the context of constructing $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\, \boldsymbol{\theta}_1)$ and $\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\, \boldsymbol{\theta}_2)$, an alternative strategy to the typical approach of minimizing the negative-log-likelihood $\mathcal{L}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ in regression problems, as presented in Section 3.4.2, is proposed. This alternative entails generating a GKDE-based estimates dataset $\mathcal{D}^*_{\mathbf{H}|\mathbf{W}}$ made up of pre-calculated estimates of conditional mean and conditional covariance of $\mathbf{H}$ given $\mathbf{W}$, using GKDE-based estimators as defined by Eqns. (3.2) and (3.3). Then, parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are adjusted such that the ANN-based statistical metamodel defined by $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\, \boldsymbol{\theta}_1)$ and $\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\, \boldsymbol{\theta}_2)$ fits dataset $\mathcal{D}^*_{\mathbf{H}|\mathbf{W}}$. Such a strategy requires a very large dataset $\mathcal{D}^*_{\text{ar}}$ in order to construct a large enough dataset $\mathcal{D}^*_{\mathbf{H}|\mathbf{W}}$ using GKDE-based

estimators. Such an adapted very large dataset $\mathcal{D}^*_{\text{ar}}$ can easily be constructed by PLoM. Therefore, dataset $\mathcal{D}^*_{\mathbf{H}|\mathbf{W}}$ consists of $\nu < n_{\text{ar}}$ elements $(\boldsymbol{\mu}^1, \boldsymbol{\zeta}^1), \ldots, (\boldsymbol{\mu}^\nu, \boldsymbol{\zeta}^\nu)$ defined as follows. For all $j = 1, \ldots, \nu$,

$$\boldsymbol{\mu}^j = \frac{\sum_{\ell=\nu+1}^{n_{\text{ar}}} \boldsymbol{\eta}^\ell_{\text{ar}} \exp\left(-\frac{1}{2s^2}(\mathbf{w}^j_{\text{ar}} - \mathbf{w}^\ell_{\text{ar}})^T [C_{\mathbf{W}}]^{-1} (\mathbf{w}^j_{\text{ar}} - \mathbf{w}^\ell_{\text{ar}})\right)}{\sum_{\ell=\nu+1}^{n_{\text{ar}}} \exp\left(-\frac{1}{2s^2}(\mathbf{w}^j_{\text{ar}} - \mathbf{w}^\ell_{\text{ar}})^T [C_{\mathbf{W}}]^{-1} (\mathbf{w}^j_{\text{ar}} - \mathbf{w}^\ell_{\text{ar}})\right)} \quad , \tag{3.15}$$

and the $m(m+1)/2$ dimensional vector $\boldsymbol{\zeta}^j$ collects all the entries of the upper triangular part of the matrix logarithm $[\log C^j]$ of the matrix $[C^j]$, defined as

$$[C^j] = \frac{\sum_{\ell=\nu+1}^{n_{\text{ar}}} \widetilde{\boldsymbol{\eta}}^\ell_{\text{ar}}(\mathbf{w}^j_{\text{ar}}) \, (\widetilde{\boldsymbol{\eta}}^\ell_{\text{ar}}(\mathbf{w}^j_{\text{ar}}))^T \exp\left(-\frac{1}{2s^2}(\mathbf{w}^j_{\text{ar}} - \mathbf{w}^\ell_{\text{ar}})^T [C_{\mathbf{W}}]^{-1} (\mathbf{w}^j_{\text{ar}} - \mathbf{w}^\ell_{\text{ar}})\right)}{\sum_{\ell=\nu+1}^{n_{\text{ar}}} \exp\left(-\frac{1}{2s^2}(\mathbf{w}^j_{\text{ar}} - \mathbf{w}^\ell_{\text{ar}})^T [C_{\mathbf{W}}]^{-1} (\mathbf{w}^j_{\text{ar}} - \mathbf{w}^\ell_{\text{ar}})\right)} \quad . \tag{3.16}$$

Note that $n_{\text{ar}} - \nu$ is the number of realizations used for the GKDE-based estimates of conditional mean and conditional covariance of $\mathbf{H}$ given $\mathbf{W} = \mathbf{w}^j_{\text{ar}}$. Hence, the least-square estimation of parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ is obtained as the parameters that minimize the cost function $\mathcal{J}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ that is written as

$$\mathcal{J}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \frac{1}{2}\left(\sum_{j=1}^{\nu} \|\boldsymbol{\mu}^j - \boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}^j_{\text{ar}}; \boldsymbol{\theta}_1)\|^2 + \sum_{j=1}^{\nu} \|\boldsymbol{\zeta}^j - \boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}^j_{\text{ar}}; \boldsymbol{\theta}_2)\|^2\right) \quad , \tag{3.17}$$

This optimization problem is solved using ADAM (ADAptive Moment estimation) algorithm [67] with a learning rate γ scheduler (see Chapter 5) that adjusts the learning rate over the course of training as

$$\gamma(k) = \max(\gamma_o \, \alpha^{(k-1)/\Delta}, \gamma_{\text{min}}) \, , \tag{3.18}$$

where k is the epoch, γ_o is the initial learning rate (default value is $\gamma_o = 0.001$), $\alpha = 0.95$ is the decay factor, $\Delta = 5$ is the decay period, and $\gamma_{\text{min}} = 1 \times 10^{-7}$ is the minimum leaning rate. For the ADAM algorithm, the parameters are fixed as $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 1 \times 10^{-8}$.

3.5 Numerical applications

3.5.1 Architecture of the ANN

Concerning the architecture of the ANN, rather than constructing two multi-outputs fully connected feedforward networks (one for $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_1)$ and one for $\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_2)$), we choose to construct m single output fully connected feedforward networks $\{\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_{1,1})\}_1, \ldots, \{\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_{1,m})\}_m$ and $d = m(m+1)/2$ single output fully connected feedforward networks $\{\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_{2,1})\}_1, \ldots, \{\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_{2,d})\}_d$, where $\boldsymbol{\theta}_1$ is rewritten as $\boldsymbol{\theta}_1 = (\boldsymbol{\theta}_{1,1}, \ldots, \boldsymbol{\theta}_{1,m})$ and $\boldsymbol{\theta}_2$ is rewritten as $\boldsymbol{\theta}_2 = (\boldsymbol{\theta}_{2,1}, \ldots, \boldsymbol{\theta}_{2,d})$. In addition, each of these parameters $\boldsymbol{\theta}_{i,j}$ are fine-tuned for each POA partition introduced in Chapter 2. It means that we have calculated one $\boldsymbol{\theta}_{i,j}$ for POA in $[0.03, 0.05]$ and another $\boldsymbol{\theta}_{i,j}$ for POA in $[0.06, 0.10]$. The training is carried out in parallel on a cluster of 3 *Tesla V100 - 32 GB* GPUs. For each fully connected feedforward network, there are four hidden layers, the number of units is 20, 250, 75 and 25 respectively.

This architecture implies a total number of 52 628 parameters (biases and weights) for the two partitions of POA. For the first and the fourth layer, Glorot [68] initialization is used and for the second and third layer, He [69] initialization is used. Rectified linear unit (ReLU) activation functions is used for each of the four hidden layers.

3.5.2 Statistical convergence analysis for the learned GKDE-based estimates dataset

A statistical convergence analysis is carried out with respect to number ν (see section 3.4.3). Figure 3.1 shows the graph of $n_{\text{ar}} \mapsto (n_{\text{ar}} - \nu + 1)^{-1} \sum_{j=\nu+1}^{n_{\text{ar}}} \| [C_{\mathbf{H}|\mathbf{W}}(\mathbf{w}_{\text{ar}}^j)] \|^2$ in which $\| \cdot \|$ is the Frobenius norm. The horizontal axis of Fig. 3.1 is the number $n_{\text{ar}} - \nu$ statistically independent realizations used in order to construct the GKDE-based estimates of conditional covariance of $\mathbf{H}$ given $\mathbf{W} = \mathbf{w}$ with $\nu = 60\,000$. It can be shown that convergence is reached for $n_{\text{ar}} - \nu = 200\,000$, that is to say $n_{\text{ar}} = 260\,000$.

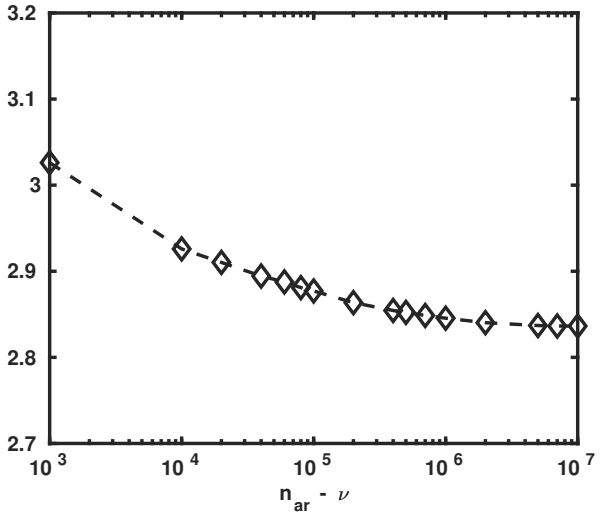
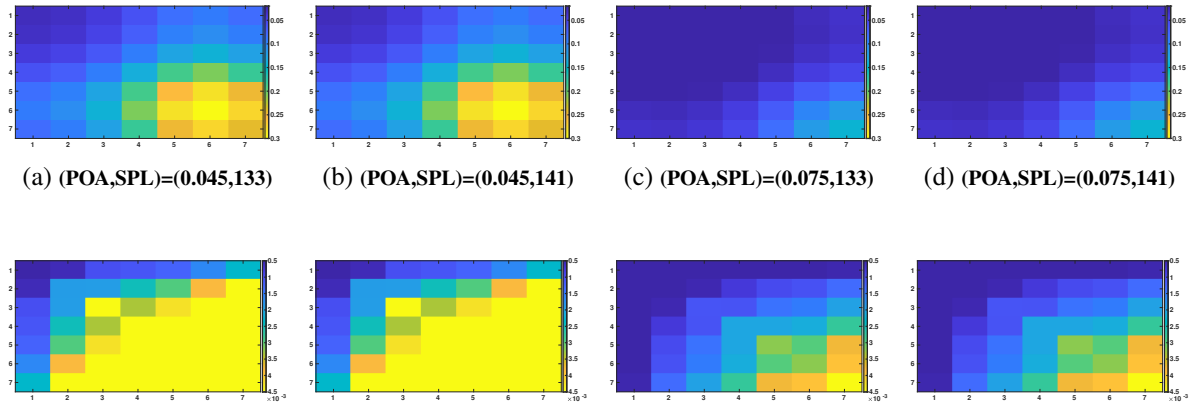


Figure 3.1: Statistical convergence analysis of the GKDE estimator for the conditional covariance matrix $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$ of $\mathbf{H}$ given $\mathbf{W} = \mathbf{w}$. Graph of $n_{\text{ar}} \mapsto (n_{\text{ar}} - \nu + 1)^{-1} \sum_{j=\nu+1}^{n_{\text{ar}}} \| [C_{\mathbf{H}|\mathbf{W}}(\mathbf{w}_{\text{ar}}^j)] \|^2$. Horizontal axis: $n_{\text{ar}} - \nu$.

3.5.3 Conditional covariance matrices of R and V given W

Using the database $\mathcal{D}_{\text{ar}}^*$ and conditional statistics based on GKDE estimation, the $(m \times m)$ conditional covariance matrix $[C_{\mathbf{Q}|\mathbf{W}}(\mathbf{w})]$ can be calculated. The expression is not included in this chapter, but its complete derivation can be found in Appendices A and B. Using Eqns. (3.7) to (3.10) allows for calculating conditional covariance matrices of resistance $[C_{\mathbf{R}|\mathbf{W}}(\mathbf{w})]$ and reactance $[C_{\mathbf{V}|\mathbf{W}}(\mathbf{w})]$ for given $\mathbf{W} = \mathbf{w}$. Furthermore, as explained in Section 3.3.3 and given models $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_1)$ and $\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_2)$, conditional covariance matrix $[C_{\mathbf{R}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)]$ (resp. $[C_{\mathbf{V}|\mathbf{W}}(\mathbf{w}, \boldsymbol{\theta}_2)]$) of $\mathbf{R}(\mathbf{w})$ (resp. $\mathbf{V}(\mathbf{w})$) can be constructed for given $\mathbf{W} = \mathbf{w}$. The entries of matrices $[C_{\mathbf{R}|\mathbf{W}}(\mathbf{w})]$ and $[C_{\mathbf{V}|\mathbf{W}}(\mathbf{w})]$ (calculated by GKDE) are displayed in Fig. 3.2. The entries of matrices $[C_{\mathbf{R}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)]$ and $[C_{\mathbf{V}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_2)]$ (calculated using

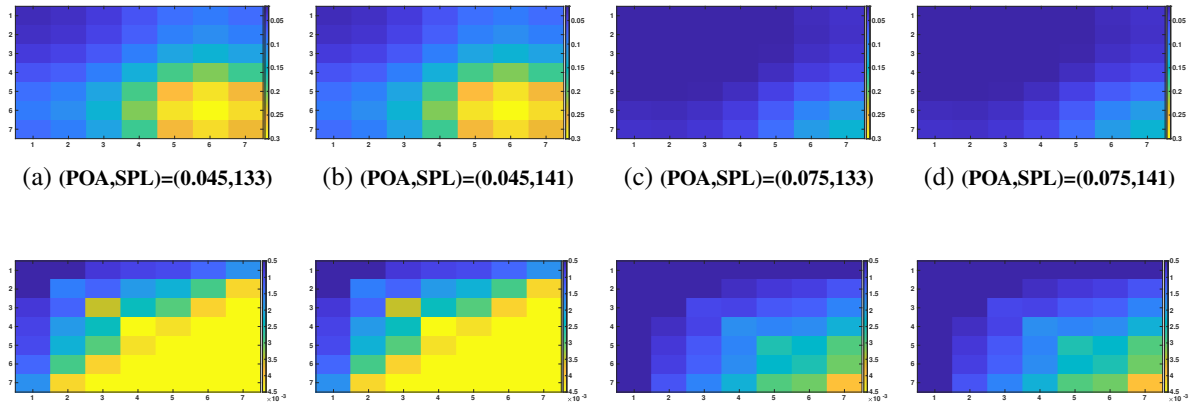
the statistical ANN-based metamodel) are displayed in Fig. 3.3. The frequency-sampled resistance is displayed in Figs 3.2a to 3.2d and in 3.3a to 3.3d. The frequency-sampled reactance is displayed in Figs 3.2e to 3.2h and 3.3e to 3.3h. It should be noted that the considered values of the control parameters are indicated in each sub-figure. It can be seen that the matrices are not diagonal and that a correlation exits between the different frequency points of the frequency-sampled impedance. It can also be seen that GKDE-based estimate in Fig. 3.2 and statistical ANN-based metamodel estimate in Fig. 3.3 of the covariance matrices are quantitatively the same.



(a) **(POA,SPL)=(0.045,133)** (b) **(POA,SPL)=(0.045,141)** (c) **(POA,SPL)=(0.075,133)** (d) **(POA,SPL)=(0.075,141)**

(e) **(POA,SPL)=(0.045,133)** (f) **(POA,SPL)=(0.045,141)** (g) **(POA,SPL)=(0.075,133)** (h) **(POA,SPL)=(0.075,141)**

Figure 3.2: Conditional covariance matrices with GKDE-based estimation of the conditional covariance of resistance (Figs.3.2a to 3.2d) and reactance (Figs. 3.2e to 3.2h) given four different values of $\mathbf{w}$



(a) **(POA,SPL)=(0.045,133)** (b) **(POA,SPL)=(0.045,141)** (c) **(POA,SPL)=(0.075,133)** (d) **(POA,SPL)=(0.075,141)**

(e) **(POA,SPL)=(0.045,133)** (f) **(POA,SPL)=(0.045,141)** (g) **(POA,SPL)=(0.075,133)** (h) **(POA,SPL)=(0.075,141)**

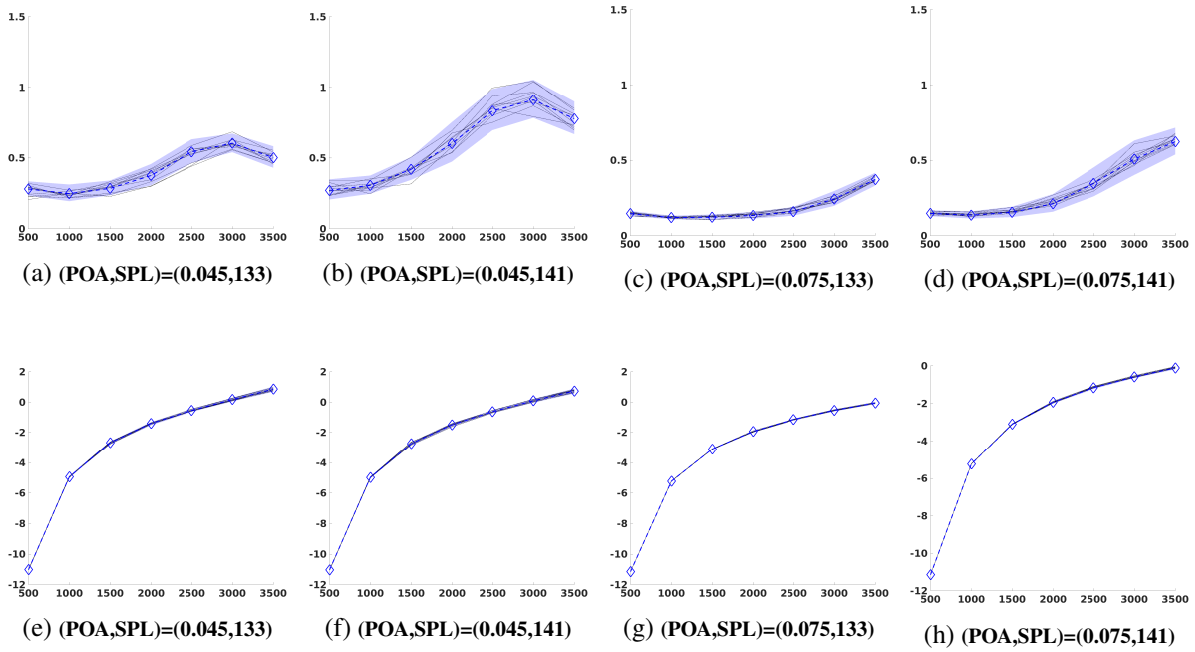
Figure 3.3: Conditional covariance matrices with statistical ANN-based metamodel estimation of the conditional covariance of resistance (Figs.3.3a to 3.3d) and reactance (Figs. 3.3e to 3.3h) given four different values of $\mathbf{w}$

3.5.4 Frequency-sampled acoustic impedance using the ANN-based statistical metamodel

For any value of $\mathbf{w}$ in its admissible set, the conditional statistics (mean values and confidence regions) of $\mathbf{R}(\mathbf{w})$ and $\mathbf{V}(\mathbf{w})$ are estimated using the statistically independent realizations $\mathbf{r}^1(\mathbf{w}; \boldsymbol{\theta}_1, \boldsymbol{\theta}_2), \ldots,$

$\mathbf{r}^N(\mathbf{w};\, \boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2)$ and $\mathbf{v}^1(\mathbf{w};\, \boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2), \ldots, \mathbf{v}^N(\mathbf{w};\, \boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2)$ as presented in Section 3.3.3 and obtained using $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\, \boldsymbol{\theta}_1)$ and $\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\, \boldsymbol{\theta}_2)$ as explained in Section 3.4.1.

Figures 3.4 shows 10 realizations of frequency-sampled resistances (Figs 3.4a to 3.4d) and 10 corresponding realizations of frequency-sampled reactance (the values of the control parameters are indicated in each sub-figure). The dashed blue line (resp. the blue domain) is the conditional mean value (resp. the conditional confidence region) of the frequency-sampled resistance (Figs. 3.4a to 3.4d) and reactance (Figs. 3.4a to 3.4d).



(a) **(POA,SPL)=(0.045,133)** (b) **(POA,SPL)=(0.045,141)** (c) **(POA,SPL)=(0.075,133)** (d) **(POA,SPL)=(0.075,141)**

(e) **(POA,SPL)=(0.045,133)** (f) **(POA,SPL)=(0.045,141)** (g) **(POA,SPL)=(0.075,133)** (h) **(POA,SPL)=(0.075,141)**

Figure 3.4: Random generations of frequency-sampled impedance using the ANN-based statistical metamodel. Conditional mean value (dashed blue line) and conditional confidence region (blue region) with a probability level $P_c = 98\%$ for the resistance ((a) to (d)) and reactance ((e) to (h)). The 10 thin black lines represent realizations of the frequency-sampled impedance generated by the ANN-based statistical metamodel for given $\mathbf{w}$. Horizontal axis is the frequency in Hz

3.5.5 Comparison of confidence regions estimated by the PLoM-GKDE-based statistical metamodel and the ANN-based statistical metamodel

In this section, we focus on the comparison of the confidence regions estimated by nonparametric statistics and the statistical ANN-based metamodel developed in this chapter. It is well known that, for convergence, nonparametric statistical estimates require a much larger number of realizations. Figure 3.5 illustrates the resistance (figures a to d) and reactance (panels e to h), showcasing the graphs of conditional mean and conditional confidence regions for a given $\mathbf{w}$. These are obtained using conditional statistics using PLoM-GKDE-based statistical metamodel and are compared with the ANN-based statistical metamodel for the frequency-sampled impedance. The blue dashed curve and the blue zone represent the conditional

mean and $p_c = 98\%$-conditional confidence region for the ANN-based statistical metamodel. The red dashed curve represents the conditional mean values estimated using Eqn. (2.10). For the estimation of the conditional confidence region (refer to Eqns. (2.11) to (2.13)), the number of independent realizations N used for the convergence of the PLoM-GKDE-based statistical estimator is varied between $N = 10^5$ to $N = 10^8$, which is indicated by red zone and a different border (as shown in the legend of Fig. 3.5). For values close to $N = 10^7$, the conditional confidence region of the PLoM-GKDE-based statistical metamodel given $\mathbf{w}$, converges towards the predictions of the ANN-based statistical metamodel in terms of estimating the associated uncertainty level. In order to understand why such a high value of N is

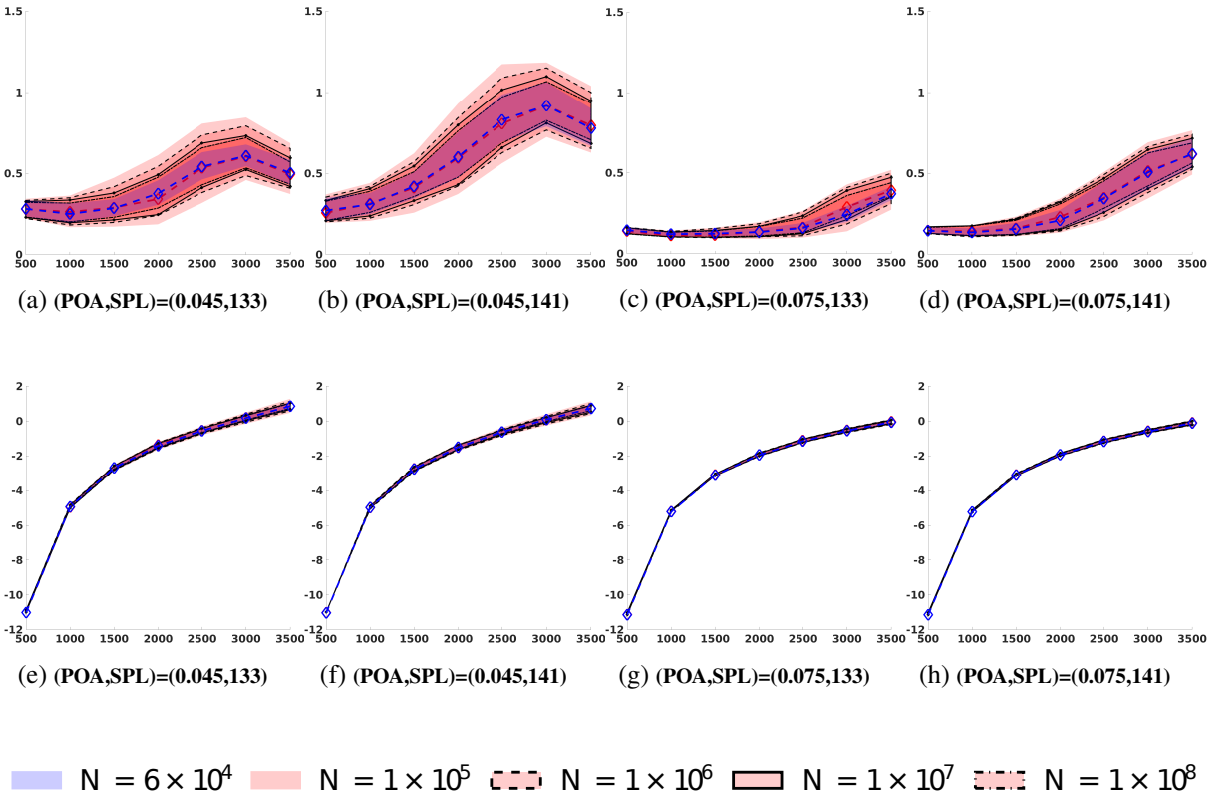
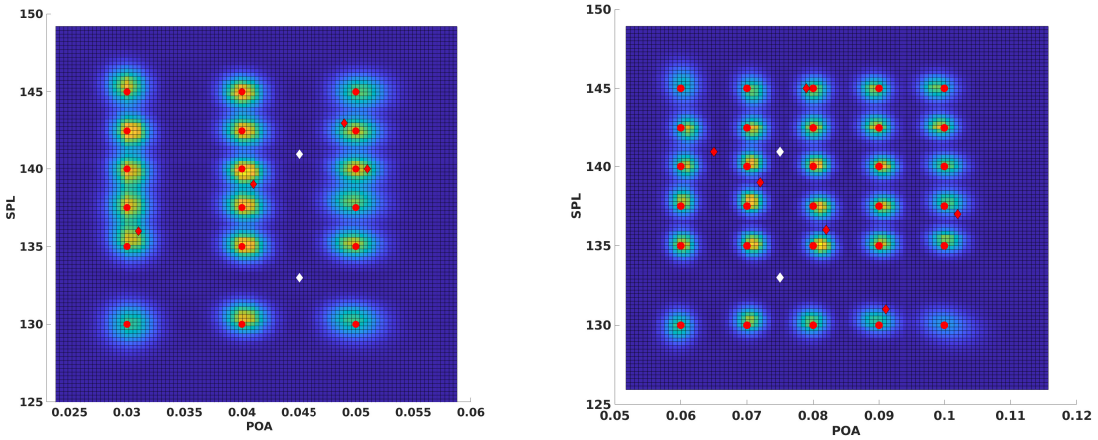


Figure 3.5: Convergence analysis of the conditional confidence region with a probability level $P_c = 98\%$ of the frequency-sampled impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)). The blue dashed curve and the blue zone represent the conditional expectation and conditional confidence regions estimated using the ANN-based statistical metamodel of the acoustic liner impedance. The red dashed curve represent the conditional mean values estimated using Eqn. (2.10). The red zone, with different outlines, represents the conditional confidence regions estimated using Eqns. (2.11) to (2.13) by varying the number of independent realizations N. Horizontal axis: Frequency in Hz.

required, figure 3.6 displays the bivariate probability density estimate of the two control parameters POA and SPL for the two partitions of dataset (see Chapter 2). The red dots represent the initial training dataset. The red diamonds are data points that are used for conditional statistics that are in the data rich zone. The white diamonds are the data points used for conditional statistics in the data poor zone. The colored contour around the red dots are the learned realizations from PLoM. The initial database was generated in a tight grid and thus has data rich and data poor zones. As seen from the contours of the random realizations generated by PLoM, many of the intermediate zones have very little to no learned realizations. In these data poor zones, both the statistical metamodel and the ANN-based metamodel



(a) Partition 1

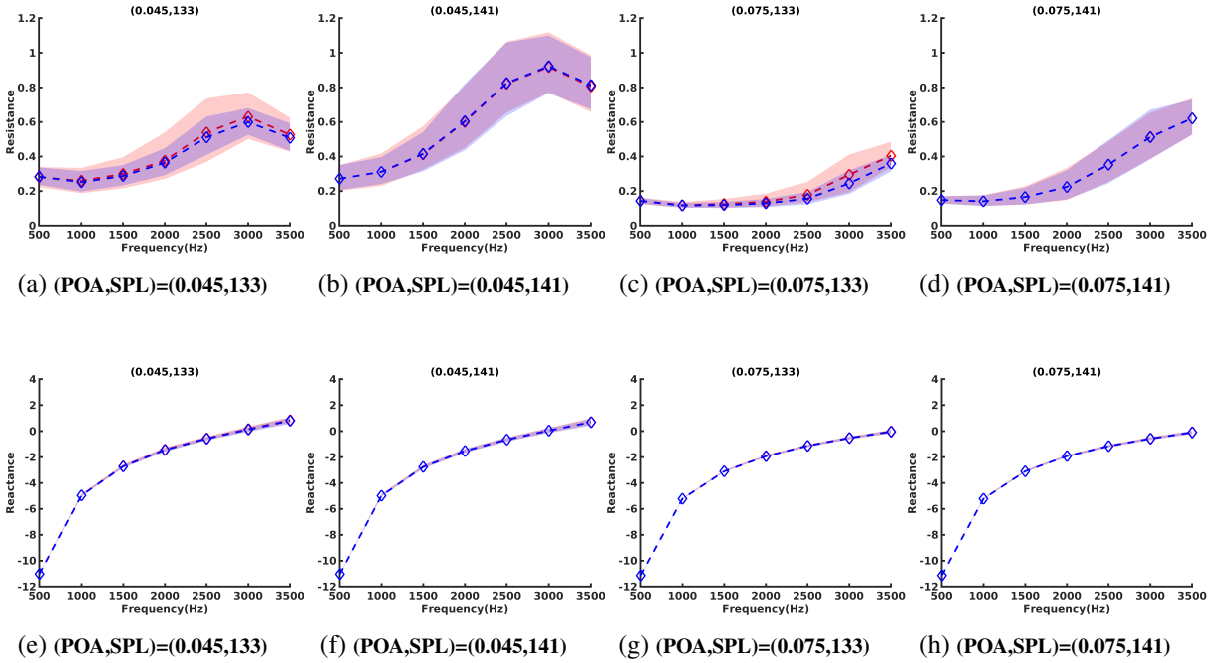
(b) Partition 2

Figure 3.6: The bivariate probability density estimate of POA and SPL for the two partitions of dataset. The red dots represent the initial training dataset. The red diamonds are data points that are used for conditional statistics that are in the data rich zone. The white diamonds are the data points used for conditional statistics in the data poor zone. The colored contour around the red dots are the learned realizations from PLoM.

must interpolate the results. It should be noted that these results are obtained for the standard value s of the Silverman bandwidth defined in Eqn. 3.4. In order to improve the convergence of the confidence region by GKDE-based estimate, and taking into account the interpolations into data poor zones of design parameter $\mathbf{w}$ into the dataset $\mathcal{D}^*_{\text{ar}}$, we propose to manually adjust the value of Silverman bandwidth s in decreasing its values at $s = 0.1186$ (this compensates the difference of the number of realizations between PLoM-GKDE-based statistical metamodel and ANN-based statistical metamodel). Using this new value of Silverman bandwidth and for given design parameters $\mathbf{W} = \mathbf{w}$ corresponding to data rich zones, the conditional mean value and the conditional confidence region with probability level $P_c = 98\%$, of the frequency-sampled impedance given $\mathbf{W} = \mathbf{w}$, corresponding to the ANN-based statistical metamodel, are presented in Figs. 3.8 and are compared with results presented in Chapter 2 (PLoM-GKDE-based statistical metamodel). Figures 3.8a to 3.8d correspond to the resistance and Figures 3.8e to 3.8h correspond to the reactance (the values of the control parameters are indicated in each sub-figure). The red dashed line (resp. the red domain) shows the conditional mean value (resp. the confidence region) from Chapter 2. The blue dashed line (resp. the blue domain) shows the conditional mean value (resp. confidence region) from the ANN-based statistical metamodel. In such data rich zones, the predictions have a relatively lower level of uncertainty *i.e.* smaller confidence regions predicted. Additionally, there is a good match between the predictions of conditional mean and conditional confidence region given $\mathbf{w}$ for the PLoM-GKDE-based statistical metamodel and ANN-based statistical metamodel.

Using the new value of Silverman bandwidth and for given design parameters $\mathbf{W} = \mathbf{w}$ corresponding to data poor zones, the conditional mean value and the conditional confidence region with probability level $P_c = 98\%$, of the frequency-sampled impedance given $\mathbf{W} = \mathbf{w}$, corresponding to the ANN-based statistical metamodel, are presented in Figs. 3.7 and are compared with results presented in Chapter 2 (PLoM-GKDE-based statistical metamodel). Figures 3.7a to 3.7d correspond to the resistance and Figures 3.7e to 3.7h correspond to the reactance (the values of the control parameters are indicated in each

sub-figure). The red dashed line (resp. the red domain) shows the conditional mean value (resp. the confidence region) from Chapter 2. The blue dashed line (resp. the blue domain) shows the conditional mean value (resp. confidence region) from the ANN-based statistical metamodel. It can be seen that,



(a) **(POA,SPL)=(0.045,133)** (b) **(POA,SPL)=(0.045,141)** (c) **(POA,SPL)=(0.075,133)** (d) **(POA,SPL)=(0.075,141)**

(e) **(POA,SPL)=(0.045,133)** (f) **(POA,SPL)=(0.045,141)** (g) **(POA,SPL)=(0.075,133)** (h) **(POA,SPL)=(0.075,141)**

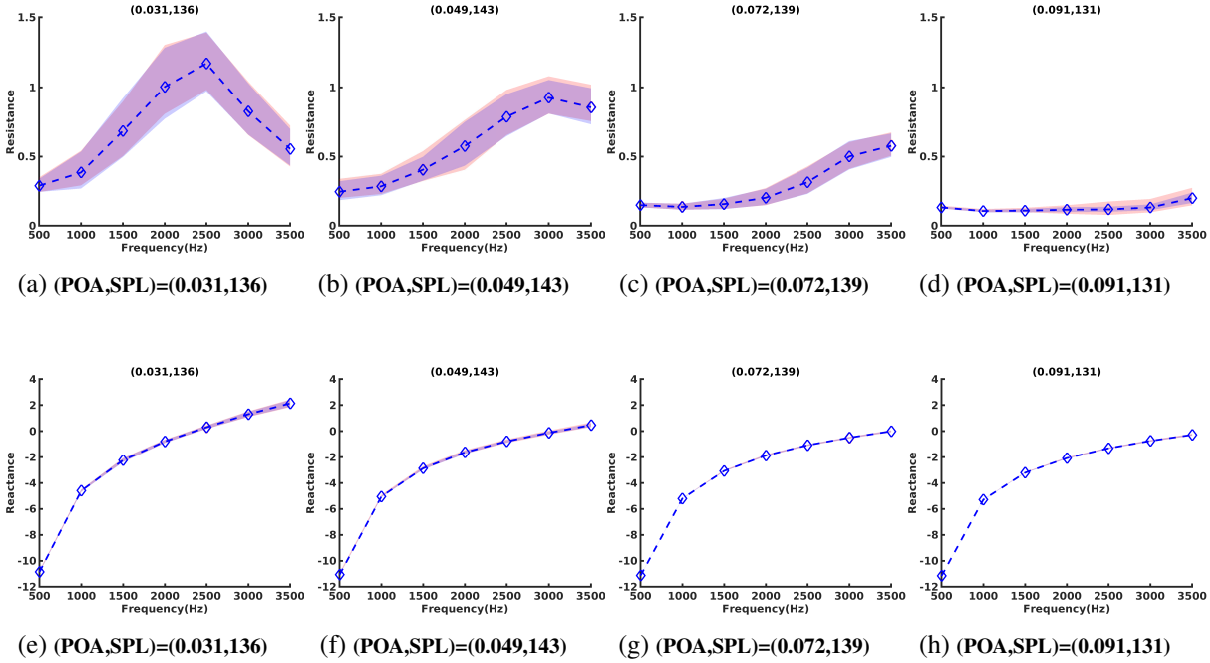
Figure 3.7: Conditional mean values and conditional confidence region with a probability level $P_c = 98\%$ of the frequency-sampled impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)). The red dashed curve and the red zone represent the conditional mean values and conditional confidence intervals estimated from Chapter 2 and the Silverman bandwidth tuned to $s = 0.1186$ for all the figures. The blue dashed curve and the blue zone represent the conditional expectation and conditional confidence intervals estimated using the ANN-based statistical metamodel of acoustic liner impedance. Horizontal axis is frequency in Hz.

concerning the conditional mean values of the frequency-sampled impedance, there is a good match between the ANN-based statistical metamodel presented in this chapter and the previous results presented in Chapter 2.

Finally, from figures 3.7 and 3.8, it can be seen that the ANN-based statistical metamodel, which has been fitted on the learned PLoM-GKDE-based estimates dataset $\mathcal{D}^*_{\mathbf{H}|\mathbf{W}}$ (with 60 000 realizations), shows conditional confidence region that are quantitatively the same as those calculated by GKDE-based estimate using the learned ACM dataset $\mathcal{D}^*_{\text{ar}}$ (260 000 realizations) with manually adjusted Silverman bandwidth s to the value 0.1186.

3.6 Conclusions and perspectives

In this chapter, an ANN-based statistical metamodel of the frequency-sampled acoustic liner impedance has been presented, for which only a small ACM data is available to fit its parameters. The control parameters are the POA and SPL with Mach number being kept at zero. The latent (unobserved) parameters introduce uncertainties that are taken into account by a probabilistic model introduced in



(a) **(POA,SPL)=(0.031,136)** (b) **(POA,SPL)=(0.049,143)** (c) **(POA,SPL)=(0.072,139)** (d) **(POA,SPL)=(0.091,131)**

(e) **(POA,SPL)=(0.031,136)** (f) **(POA,SPL)=(0.049,143)** (g) **(POA,SPL)=(0.072,139)** (h) **(POA,SPL)=(0.091,131)**

Figure 3.8: Conditional mean values and conditional confidence region with a probability level $P_c = 98\%$ of the frequency-sampled impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)). The red dashed curve and the red zone represent the conditional mean values and conditional confidence intervals estimated from Chapter 2 and the Silverman bandwidth fixed at $s = 0.1186$ for all the figures. The blue dashed curve and the blue zone represent the conditional expectation and conditional confidence intervals estimated using the ANN-based statistical metamodel of acoustic liner impedance. Horizontal axis: Frequency in Hz.

Chapter 2. The construction of the statistical metamodel uses a PCA to construct a statistical reduced representation of the frequency-sampled vector of the random log-resistance and the random reactance. For fitting the ANN-based statistical metamodel, a big training dataset is constructed using probabilistic learning on manifolds (PLoM). A *prior* conditional probability distribution of the reduced representation given the control parameters is then constructed and assumed to be Gaussian, yielding a multivariate log-normal distribution for the resistance and a multivariate Gaussian distribution for the reactance. The metamodels of the hyperparameters of such conditional probabilistic model is presented as fully-connected feedforward neural networks. As the reduced representation is modeled by a centered and normalized random vector, some constraints have to be taken into account in the minimization of the negative-log likelihood when fitting the parameters (biases and weights) of the neural network. The constrained problem is transformed in an unconstrained one, requiring the construction of a second training dataset to estimate the conditional mean vectors and conditional covariance matrices for which the learned realizations are generated using PLoM. The novelty of this chapter lies in the methodology used to construct an ANN-based statistical metamodel of acoustic liner impedance, which can be used as a low-computational cost metamodel to predict the confidence interval and the mean value of the impedance given any value of the control parameter. The gradients of the mean values and confidence regions can also easily be derived using classical backpropagation algorithms for very cheap computational cost. In Chapter 4, this approach is extended to non-zero Mach number.

Chapter 4

Statistical metamodel based on neural network for positive Mach numbers with missing data

4.1 Introduction

In order to facilitate the reading of this chapter we have reintroduced a part of the methodology and the formulas already given in Chapters 2 and 3.

4.1.1 Framework for positive Mach number with missing data

This chapter is devoted to the construction of the two statistical metamodels presented in Chapter 2 and 3, for the case of missing data with a small training dataset. Similar to the case of zero Mach number, one evaluation for positive Mach number, using an ACM, is computationally expensive and consequently, the ACM cannot be used many times for constructing the parameterized acoustic liner impedance. Consequently, a statistical metamodel is required. Furthermore, for some values of the design parameters the ACM might not converge for all the prescribed values of frequency ω. In this case those values are missing in the ACM dataset and strategy has to be proposed to mitigate the missing data, modeling the frequency as a random variable. This is the reason why we propose to develop a PLoM-GKDE-based statistical metamodel of the acoustic impedance parameterized by its frequency and the percentage of open area (POA) (as in Chapters 2 and 3) and additionally by the Mach number (M), hole diameter (d), and perforated plate thickness (t). Note that the effects of the sound pressure level (SPL) is considered as a second-order parameter with respect to the effects of the previous ones and consequently, will not be taken into account in the present analyses. The uncertainties induced by modeling errors will not be taken into account because its level cannot be identified with experiments for the following reasons: (i) impossibility to measure, in acoustic benches, detailed boundary layer profiles, friction velocity, and other thermodynamic parameters; (ii) lack of open literature for the conditions we

are interested in studying and experimental setup yielded unsatisfactory results.

4.1.2 Objectives

As previously, the objective of the chapter is to construct a PLoM-GKDE-based statistical metamodel and ANN-based statistical metamodel for the framework mentioned above, in particular for additional control parameters and the case of missing data.

4.1.3 Proposed methodology

The methodologies presented in Chapter 2 and 3 are reused. For the construction of the ANN-based statistical metamodel, we use the PLoM-GKDE-based statistical metamodel to mitigate any missing data within its admissible set thereby enhancing the fitting of the ANN-based statistical metamodel.

4.1.4 Organization of the chapter

The chapter is organized as follows. Section 4.2 briefly defines the control parameters and the ACM for calculating the acoustic impedance, which is to be used for constructing the ACM dataset (small size). Section 4.3 is dedicated to the construction of PLoM-GKDE-based statistical metamodel. Section 4.4 is devoted to the construction of ANN-based statistical metamodel. Section 4.5 deals with the problem of missing data presents the solution for it. It also shows the comparison of the two statistical metamodels.

4.2 Defining the liner system and ACM dataset

We consider again the perforated liner system whose scheme is shown in Fig. 2.1. The domain decomposition used is shown in Fig. 2.2. The parameters used to control the liner system are POA, Mach number (M), perforated plate thickness (t), hole diameter (d) and frequency (ω). Then for $n_{\mathbf{w}} = 5$, we define the $\mathbb{R}^{n_{\mathbf{w}}}$-valued control parameters as $\mathbf{w} = (w_1, \ldots, w_5)$, in which w_1 is POA, w_2 is Mach number M, w_3 is thickness t, w_4 is diameter d and w_5 is frequency ω. The reduced impedance is denoted as $\mathbf{w} \mapsto z^{\,\text{acm}}(\mathbf{w})$ where $z^{\,\text{acm}}(\mathbf{w}) = r^{\,\text{acm}}(\mathbf{w}) + \iota\, v^{\,\text{acm}}(\mathbf{w})$ that depends only on the control parameters $\mathbf{w}$, and in which $\iota = \sqrt{-1}$, $r^{\,\text{acm}}(\mathbf{w})$ is the resistance, and $v^{\,\text{acm}}(\mathbf{w})$ is the reactance. It should be noted that impedance is not written as $z^{\,\text{acm}}(\omega;\, \mathbf{w})$ but as $z^{\,\text{acm}}(\mathbf{w})$ because $\mathbf{w}$ is not the vector of design parameters (as in Chapters 2 and 3) but it is the vector of control parameters, for which the frequency ω is a component. The computational domain is centered around the resonator, within which the Navier-Stokes equations are solved. The computational model consists of 278 514 degrees of freedom. The ACM computes the resistance $r^{\,\text{acm}}(\mathbf{w})$ and the reactance $v^{\,\text{acm}}(\mathbf{w})$ by solving the Navier-Stokes equations using the numerical method presented in [8] for $\mathbf{w} = (w_1, w_2, w_3, w_4, w_5) \in [0.02, 0.05] \times [0, 0.5] \times [0.5, 1] \times [0.3, 0.8] \times [1000, 4000]$. Therefore, the values of $\mathbf{w}$ belongs to the finite set

$$\mathcal{C}_{\mathbf{w}} = \left\{\mathbf{w}^j, j = 1, \ldots, N_d\right\} \subset \mathcal{S}_{\mathbf{w}}, \quad \mathbf{w}^j \in \mathbb{R}^{n_{\mathbf{w}}}, \tag{4.1}$$

where $n_d = 343$ and $\mathbf{w}^j$ represents the points for which the ACM simulations were performed. The points in set $\mathcal{C}_{\mathbf{w}}$ can be considered as realizations of the random variable $\mathbf{W}$ and that $\mathcal{C}_{\mathbf{w}}$ belongs to the domain $\mathcal{S}_{\mathbf{w}} \in \mathbb{R}^{n_{\mathbf{w}}}$, which can be considered as a support of the prior probability distribution of $\mathbf{W}$, and consequently frequency is modeled as a random variable to take into account missing data for some values of frequency ω. For $j = 1, \ldots, n_d$, the corresponding resistance $r^{\text{acm},j} = r^{\text{acm}}(\mathbf{w}^j) \in \mathbb{R}$ and reactance $v^{\text{acm},j} = v^{\text{acm}}(\mathbf{w}^j) \in \mathbb{R}$ are given, and we define the vector $\mathbf{q}^{\text{acm},j} = (r^{\text{acm},j}, v^{\text{acm},j}) \in \mathbb{R}^{n_{\mathbf{q}}}$ where $n_{\mathbf{q}} = 2$. Finally, we define the ACM dataset $\mathcal{D}_{\text{acm}}$ as the set of points $\{(\mathbf{w}^j, \mathbf{q}^{\text{acm},j}), j = 1, \ldots, n_d\} \in \mathbb{R}^{n_{\mathbf{w}} \times n_{\mathbf{q}}}$.

Although, the level of model uncertainties cannot be experimentally identified, these uncertainties are taken into account by considering $\mathbf{Q} = (R, V)$ as a random vector whose conditional probability density function, given $\mathbf{W} = \mathbf{w}^j$, is denoted by $\mathbf{q} \mapsto p_{\mathbf{Q}|\mathbf{W}}(\mathbf{q} \,|\, \mathbf{w}^j)$, $p_{R|\mathbf{W}}(r \,|\, \mathbf{w}^j)$, and $p_{V|\mathbf{W}}(v \,|\, \mathbf{w}^j)$. Note that R and V are now scalar-valued random variables because the frequency is now considered as control parameter (there is no frequency sampling).

By employing PLoM algorithm, we learn the joint probability density function $p_{\mathbf{Q},\mathbf{W}}$ of the random vectors $\mathbf{Q}$ and $\mathbf{W}$, using $\mathcal{D}_{\text{acm}}$ as the training dataset. We then generate n_{ar} additional statistically independent realizations $\{(\mathbf{w}^{\ell}_{\text{ar}}, \mathbf{q}^{\ell}_{\text{ar}}), \ell = 1, \ldots, n_{\text{ar}}\}$ that constitutes the learned dataset $\mathcal{D}_{\text{ar}}$.

4.3 Construction of PLoM-GKDE-based statistical metamodel

As explained at the beginning of the chapter, we give again the formulas in order to help with the reading.

Considering the end of Section 4.2, random vector $\mathbf{Q}$ has two components, $Q_1 = R$ and $Q_2 = V$. The PLoM-GKDE-based statistical metamodel of the liner system is completely expressed by the conditional probability distribution $P_{\mathbf{Q}|\mathbf{W}}(d\mathbf{q}|\mathbf{w}) = p_{\mathbf{Q}|\mathbf{W}}(d\mathbf{q}|\mathbf{w})d\mathbf{q}$ of the random vector $\mathbf{Q}$ given $\mathbf{W} = \mathbf{w}$ for any $\mathbf{w} \in \mathcal{S}_{\mathbf{w}} \subset \mathbb{R}^{n_{\mathbf{w}}}$ where $p_{\mathbf{Q}|\mathbf{W}}$ is the conditional probability density function on $\mathbb{R}^{n_{\mathbf{q}}}$ given $\mathbf{W}$. As previously explained, this conditional probability distribution can be estimated using nonparametric statistics that require a large dataset of $(\mathbf{Q}, \mathbf{W})$ for convergence of its statistical estimate. Therefore, the learned dataset $\mathcal{D}_{\text{ar}}$ described in Section 4.2 is used. The connection between the joint probability density function of $(\mathbf{Q}, \mathbf{W})$ and the conditional probability density function of interest is the following

$$p_{\mathbf{Q}|\mathbf{W}}(\mathbf{q}|\mathbf{w}) = \frac{1}{p_{\mathbf{W}}(\mathbf{w})}\, p_{\mathbf{Q},\mathbf{W}}(\mathbf{q}, \mathbf{w})\,, \tag{4.2}$$

in which $p_{\mathbf{Q},\mathbf{W}}$ is the joint probability density function on $\mathbb{R}^{n_{\mathbf{q}}} \times \mathbb{R}^{n_{\mathbf{w}}}$ of random variables $\mathbf{Q}$ and $\mathbf{W}$, and where $p_{\mathbf{W}}(\mathbf{w}) = \int_{\mathbb{R}^{n_{\mathbf{q}}}} p_{\mathbf{Q},\mathbf{W}}(\mathbf{q}, \mathbf{w})\, d\mathbf{q}$ is the probability density function of $\mathbf{W}$ at point $\mathbf{w} \in \mathcal{S}_{\mathbf{w}}$. We are interested in deriving conditional statistics from conditional probability distribution $P_{\mathbf{Q}|\mathbf{W}}$, such as the conditional mean values and the conditional confidence regions. The results presented in Section 4.3.1 will be the conditional mathematical expectation,

$$E\{\mathbf{Q}|\mathbf{W} = \mathbf{w}\} = \int_{\mathbb{R}_{n_{\mathbf{q}}}} \mathbf{q}\, p_{\mathbf{Q}|\mathbf{W}}(\mathbf{q}|\mathbf{w})\, d\mathbf{q}\,, \tag{4.3}$$

for any $\mathbf{w}$ in $\mathcal{S}_{\mathbf{w}}$. We will also present the conditional confidence region of the random resistance R and reactance V for given $\mathbf{W} = \mathbf{w}$. Let Q_i be the component of $\mathbf{Q}$ representing either resistance R or

reactance V. Then the lower bound q_i^- and the upper bound q_i^+ of the conditional confidence interval of Q_i given $\mathbf{W} = \mathbf{w}$ for a probability level p_c are defined by

$$q_i^+ : \mathrm{Proba}\{\mathrm{Q_i} \leq \mathrm{q_i^+} \,|\, \mathbf{W} = \mathbf{w}\} = \mathrm{p_c}\,, \tag{4.4}$$

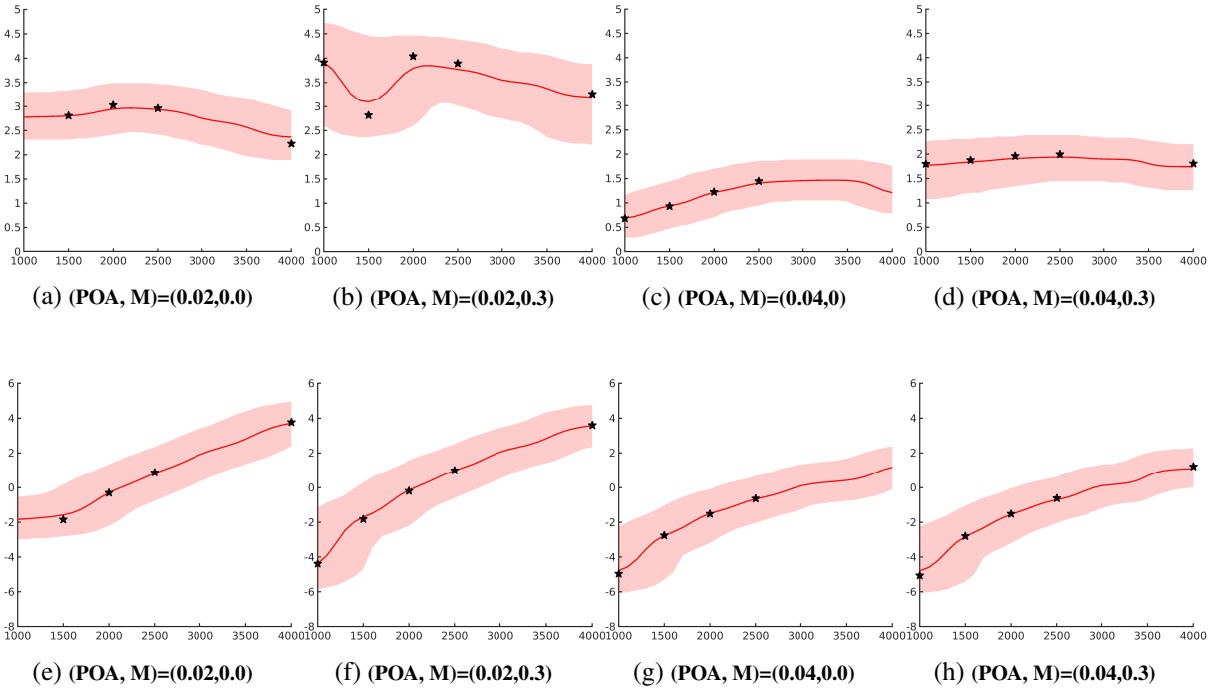
$$q_i^- : \mathrm{Proba}\{\mathrm{Q_i^-} \leq \mathrm{q_i^-} \,|\, \mathbf{W} = \mathbf{w}\} = 1 - \mathrm{p_c}\,, \tag{4.5}$$

and where the probability in Eqns. (4.4) and (4.5) are calculated with the conditional cumulative distribution function,

$$\mathrm{Proba}\{\mathrm{Q_i} \leq \mathrm{q_i^*} \,|\, \mathbf{W} = \mathbf{w}\} = \int_{-\infty}^{\mathrm{q_i^*}} \mathrm{p_{Q_i|W}}(\mathrm{q_i^*} \,|\, \mathbf{w})\, \mathrm{dq_i}\,, \tag{4.6}$$

in which the conditional probability density function $p_{Q_i|\mathbf{W}}$ is derived from conditional probability density function $p_{\mathbf{Q}|\mathbf{W}}$ by an integration on $\mathbb{R}$. The complete derivation of the conditional expectation and conditional confidence regions using the learned dataset, given any values $\mathbf{w} \in \mathcal{S}_\mathbf{W}$, and their explicit algebraic expressions can be found in Appendix B.

4.3.1 Results - PLoM-GKDE-based statistical Metamodel



(a) **(POA, M)=(0.02,0.0)** (b) **(POA, M)=(0.02,0.3)** (c) **(POA, M)=(0.04,0)** (d) **(POA, M)=(0.04,0.3)**

(e) **(POA, M)=(0.02,0.0)** (f) **(POA, M)=(0.02,0.3)** (g) **(POA, M)=(0.04,0.0)** (h) **(POA, M)=(0.04,0.3)**

Figure 4.1: Conditional mean values and conditional confidence region with a probability level $P_c = 98\%$ of the impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)) for given $\mathbf{w}$. The parameters $t = 1$ and $d = 0.4$ are fixed for all the graphs. The red curve and the red zone represent the conditional mean values and conditional confidence regions and the Silverman bandwidth fixed at $s = 0.1641$ for all the figures. The black stars represent the points that were present in the training dataset $\mathcal{D}_{\mathrm{acm}}$. Horizontal axis: Frequency in Hz.

The PLoM-GKDE-based statistical metamodel is constructed using $n_{\mathrm{ar}} = 1\ 000\ 188$ realizations. In order to assess the capability of this statistical metamodel to deal with missing data (for missing

frequencies in the ACM dataset), its predictions are compared with the values of acoustic impedance for the control parameters in the ACM dataset. Figures 4.1 show the graphs of the conditional mean values and conditional confidence region with a probability level $P_c = 98\%$ of the impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)) for given $\mathbf{w}$ in the ACM dataset. The control parameters POA and M vary (as shown in the figure) and parameters t and d are fixed in all the graphs to the values $t = 1$ and $d = 0.4$. For all the graphs the Silverman bandwidth is tuned to $s = 0.1641$. In the initial training dataset $\mathcal{D}_{\text{acm}}$, the points are shown with black stars. It can be seen that the PLoM-GKDE-based statistical metamodel is able to generalize well, even in zones for which training points are missing, with acceptable levels of uncertainty, relative to zones where training points are present. We are now interested in knowing the evolution of acoustic impedance with respect to other control parameters. In Figures 4.2 and 4.3,

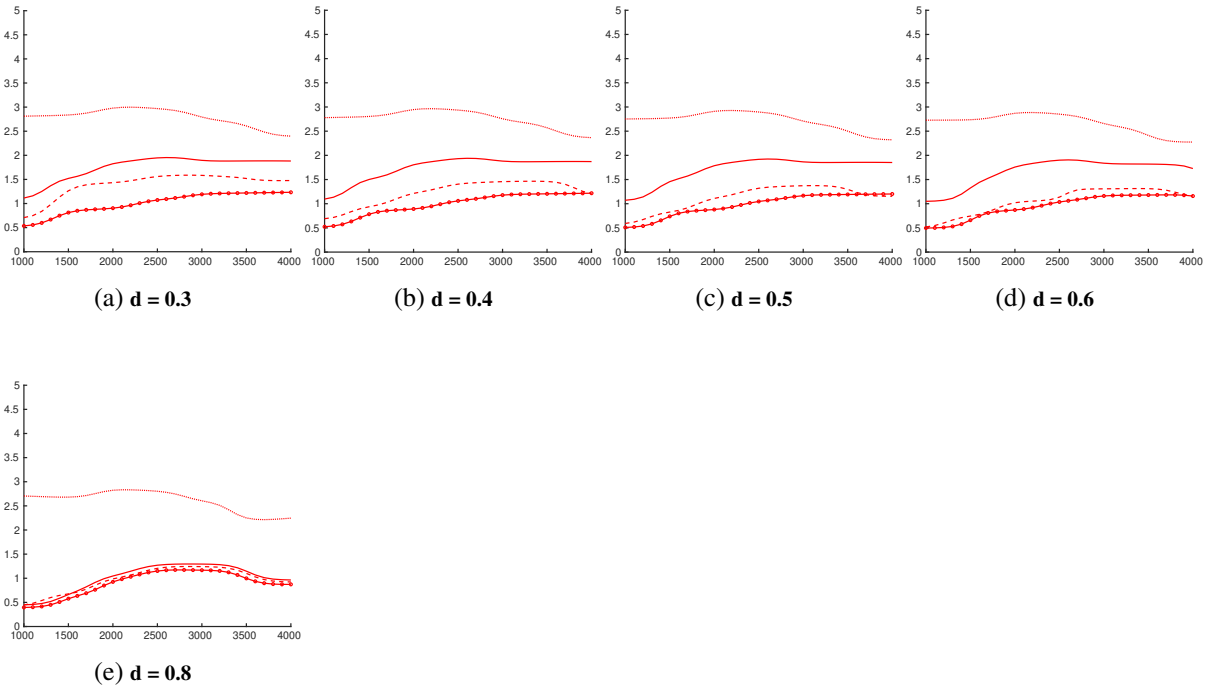


Figure 4.2: Conditional mean values of resistance given $\mathbf{w}$. The hole diameter d is varied from ((a) to (e)) as shown. The parameters $M = 0$ and $t = 1$ are fixed for all the graphs. The four different red curves represent different values of POA (between 0.02 and 0.04) as shown in the legend. Horizontal axis: Frequency in Hz.

the conditional mean values of resistance and reactance given $\mathbf{w}$ are plotted; Figs. (a) to (e) depict the variation of hole diameter d between 0.3 and 0.8, in which the horizontal axis is frequency in Hz. In all the figures, the parameters M and t are fixed to the values 0 and 1, respectively. The four different red curves represent different values of POA (between 0.02 and 0.04) as shown in the legend. In Figures 4.4 and 4.5, the conditional mean values of resistance and reactance given $\mathbf{w}$ are plotted; Figs. (a) to (e) depict the variation of hole diameter d between 0.3 and 0.8, in which the horizontal axis is frequency in Hz. In all the figures, the parameters M and t are fixed to the values 0.3 and 1, respectively. The four different red curves represent different values of POA (between 0.02 and 0.04) as shown in the legend.

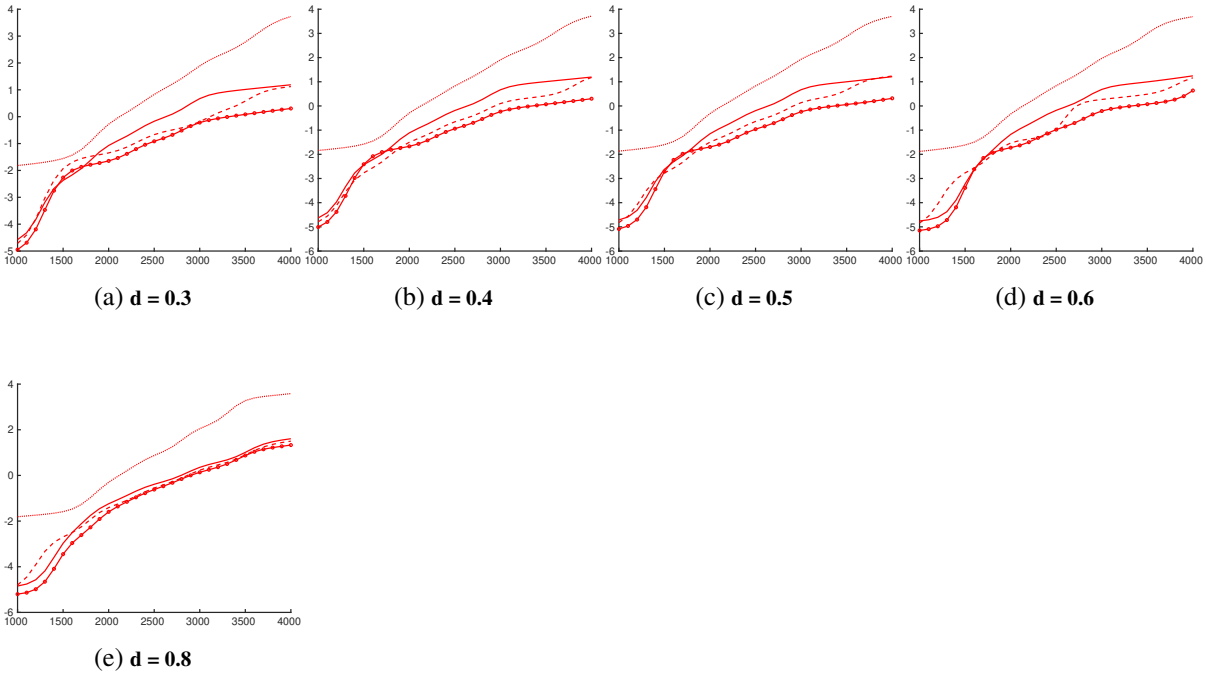


Figure 4.3: Conditional mean values of reactance given $\mathbf{w}$. The hole diameter d is varied from ((a) to (e)) as shown. The parameters $M = 0$ and $t = 1$ are fixed for all the graphs. The four different red curves represent different values of POA (between 0.02 and 0.04) as shown in the legend. Horizontal axis: Frequency in Hz.

4.4 Construction of the ANN-based statistical metamodel

For the construction of the ANN-based statistical metamodel, the learned dataset $\mathcal{D}_{\text{ar}}$ is modified as follows. Let $\{\mathbf{q}_{\text{ar}}^{\ell} = (\log r_{\text{ar}}^{\ell}, v_{\text{ar}}^{\ell}), \ell = 1, \ldots, n_{\text{ar}}\}$ be the set of the learned realizations of random vector $\mathbf{Q} = (\log R, V)$, whose conditional probability density function given any $\mathbf{W} = \mathbf{w}_{\text{ar}}^{\ell}$ is $\mathbf{q}_{\text{ar}} \mapsto p_{\mathbf{Q}|\mathbf{W}}(\mathbf{q}_{\text{ar}}|\mathbf{w}_{\text{ar}}^{\ell})$. The new learned dataset is denoted as $\mathcal{D}_{\text{ar}}^{*}$ and is composed of $\{(\mathbf{w}_{\text{ar}}^{\ell}, \mathbf{q}_{\text{ar}}^{\ell}), \ell = 1, \ldots, n_{\text{ar}}\}$, and will be used henceforth for the construction of the ANN-based statistical metamodel.

As explained in Chapter 3, in the context of the construction of an ANN-based statistical metamodel, we need to introduce an algebraic representation of the conditional probability distribution of $\mathbf{Q}$ given $\mathbf{W}$, depending on hyperparameters. We refer the reader to Chapter 3 for all the details of the construction. Only the architecture will be presented in the next subsection.

4.4.1 Architecture of the statistical ANN-based metamodel

As done in Chapter 3, rather than constructing two multi-outputs fully connected feedforward networks (one for $\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_1)$ and one for $\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_2)$), the architecture of the statistical ANN-based metamodel is chosen as m single output fully connected feedforward networks $\{\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_{1,1})\}_1, \ldots,$ $\{\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}; \boldsymbol{\theta}_{1,m})\}_m$ and $d = m(m+1)/2$ single output fully connected feedforward networks

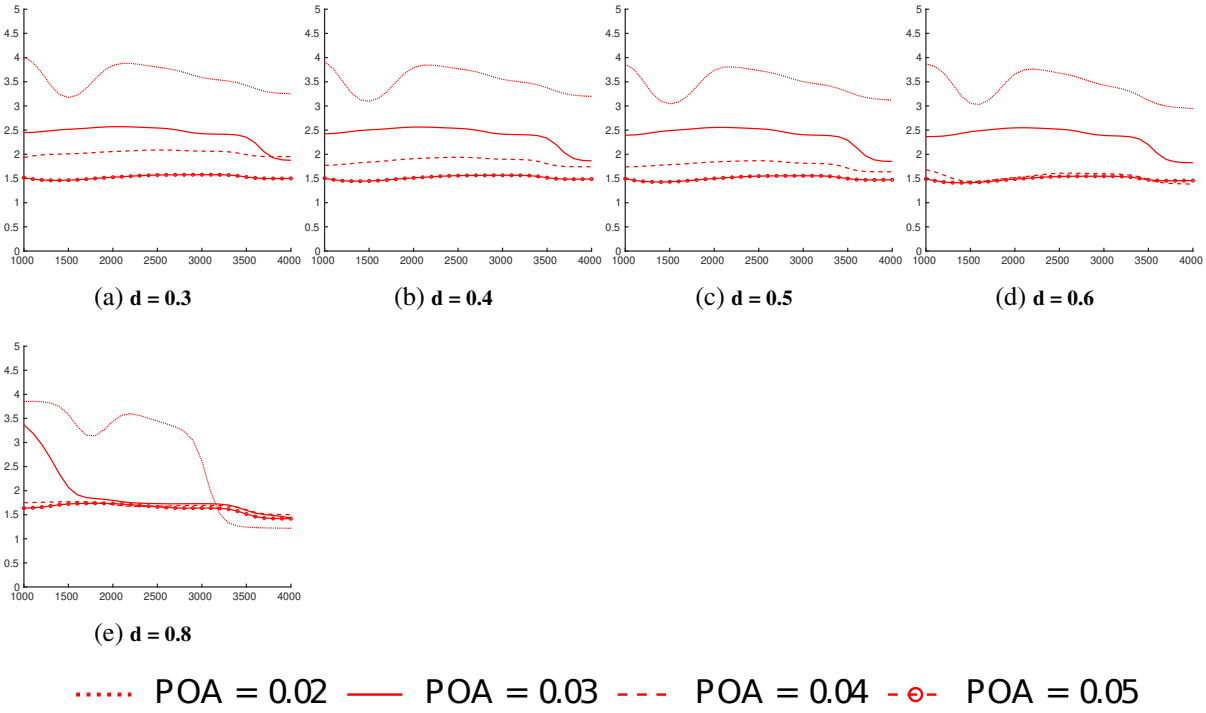


Figure 4.4: Conditional mean values of resistance given $\mathbf{w}$. The hole diameter d is varied from ((a) to (e)) as shown. The parameters $M = 0.3$ and $t = 1$ are fixed for all the graphs. The four different red curves represent different values of POA (between 0.02 and 0.04) as shown in the legend. Horizontal axis: Frequency in Hz.

$\{\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\, \boldsymbol{\theta}_{2,1})\}_1, \ldots, \{\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w};\, \boldsymbol{\theta}_{2,d})\}_d$, where $\boldsymbol{\theta}_1$ is rewritten as $\boldsymbol{\theta}_1 = (\boldsymbol{\theta}_{1,1}, \ldots, \boldsymbol{\theta}_{1,m})$ and $\boldsymbol{\theta}_2$ is rewritten as $\boldsymbol{\theta}_2 = (\boldsymbol{\theta}_{2,1}, \ldots, \boldsymbol{\theta}_{2,d})$. The training is also carried out in parallel on a cluster of 3 *Tesla V100 - 32 GB* GPUs. For each fully connected feedforward network, there are seven hidden layers, the number of units is 50, 425, 350, 250, 125, 100, and 75 respectively. This architecture implies a total number of 310 451 parameters (biases and weights). For the first and the seventh layer, Glorot [68] initialization is used and for the layers in between them, He [69] initialization is used. Rectified linear unit (ReLU) activation functions is used for each of the seven hidden layers.

4.4.2 Statistical convergence analysis for the learned GKDE-based estimates dataset

A statistical convergence analysis is carried out with respect to number ν (see Chapter 3). Figure 4.6 shows the graph of $n_{\text{ar}} \mapsto (n_{\text{ar}} - \nu + 1)^{-1} \sum_{j=\nu+1}^{n_{\text{ar}}} \| [C_{\mathbf{H}|\mathbf{W}}(\mathbf{w}_{\text{ar}}^j)] \|^2$ in which $\| \cdot \|$ is the Frobenius norm. The horizontal axis of Fig. 4.6 is the number $n_{\text{ar}} - \nu$ statistically independent realizations used in order to construct the GKDE-based estimates of conditional covariance of $\mathbf{H}$ given $\mathbf{W} = \mathbf{w}$ with $\nu = 160\,000$. It can be shown that convergence is reached for $n_{\text{ar}} - \nu = 500\,000$, that is to say $n_{\text{ar}} = 340\,000$.

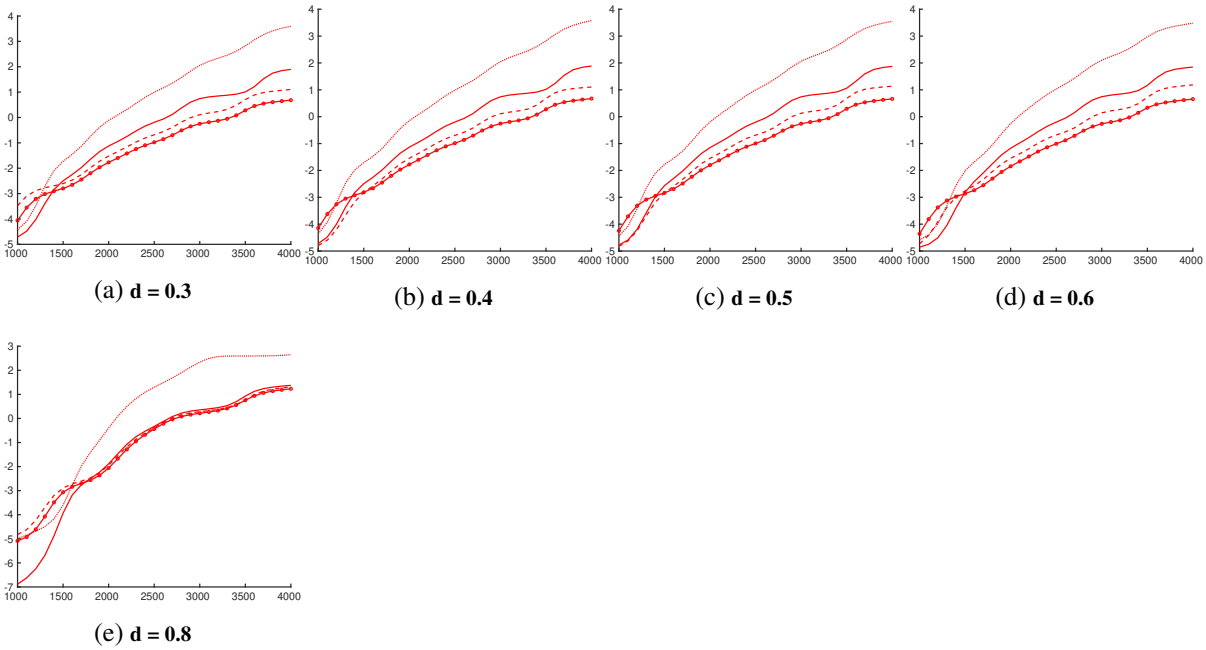


Figure 4.5: Conditional mean values of reactance given $\mathbf{w}$. The hole diameter d is varied from ((a) to (e)) as shown. The parameters $M = 0.3$ and $t = 1$ are fixed for all the graphs. The four different red curves represent different values of POA (between 0.02 and 0.04) as shown in the legend. Horizontal axis: Frequency in Hz.

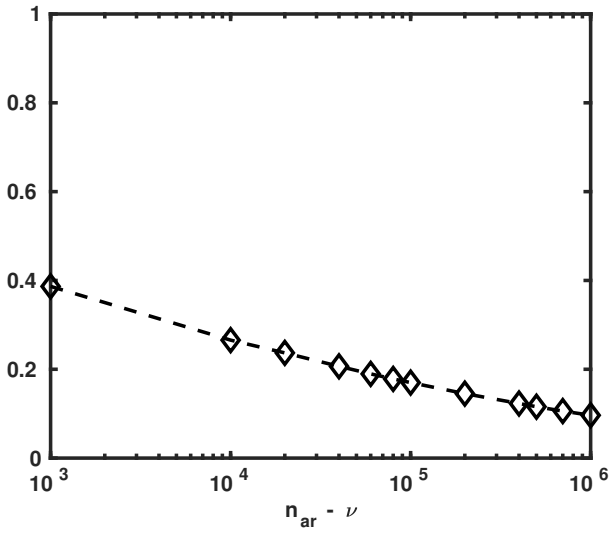
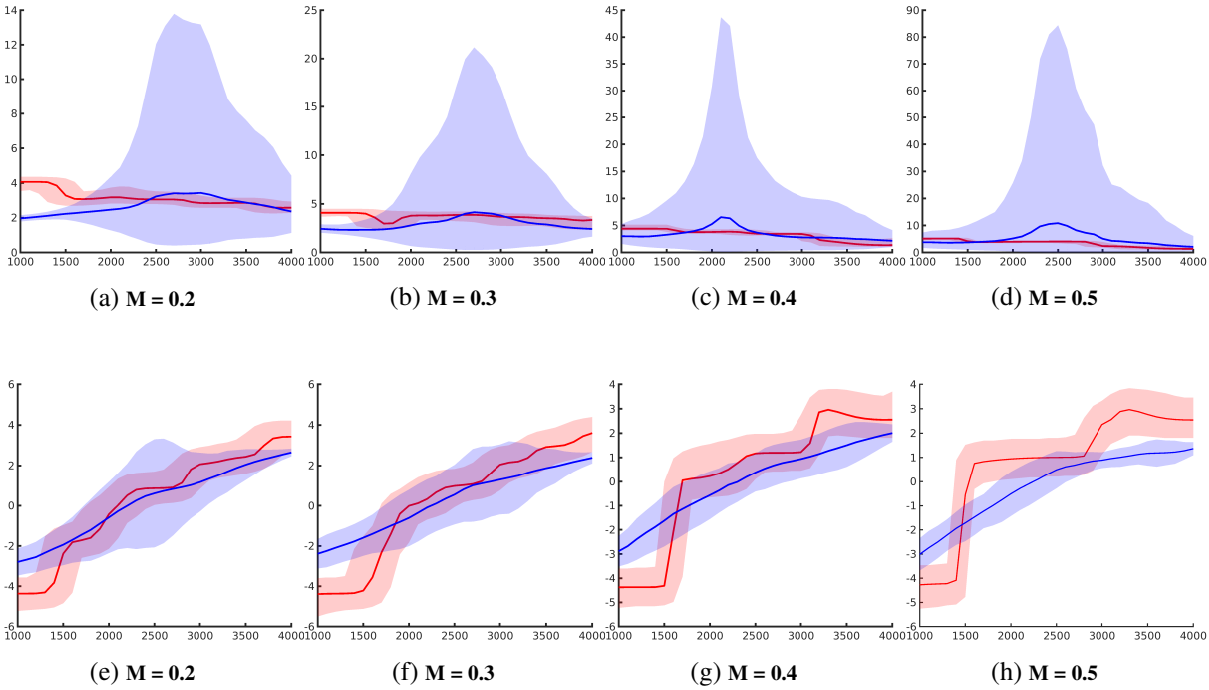


Figure 4.6: Statistical convergence analysis of the GKDE estimator for the conditional covariance matrix $[C_{\mathbf{H}|\mathbf{W}}(\mathbf{w})]$ of $\mathbf{H}$ given $\mathbf{W} = \mathbf{w}$. Graph of $n_{\text{ar}} \mapsto (n_{\text{ar}} - \nu + 1)^{-1} \sum_{j=\nu+1}^{n_{\text{ar}}} \| [C_{\mathbf{H}|\mathbf{W}}(\mathbf{w}_{\text{ar}}^j)] \|^2$. Horizontal axis: $n_{\text{ar}} - \nu$.

4.5 Results and Discussions

Figure 4.7 shows the mean values and confidence region with a probability level $P_c = 98\%$ of the impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)), for M varying between 0.2 and 0.5.

The parameters are: POA $= 0.02$, $t = 0.5$ and $d = 0.4$, that are fixed for all the graphs. The red curve and the red zone represent the conditional mean values and conditional confidence regions for given $\mathbf{w}$, using the PLoM-GDKE-based statistical metamodel for which the Silverman bandwidth is tuned to $s = 0.1641$ for all the graphs. The blue curve and the blue zone represent the conditional mean values and conditional confidence regions for given $\mathbf{w}$, using the ANN-based statistical metamodel. It can be seen that the ANN-based statistical metamodel does not predict the results given by PLoM-GDKE-based statistical metamodel. Upon inspection, it was found that the ACM dataset $\mathcal{D}_{\text{acm}}$, does not contain any value of the acoustic impedance at $t = 0.5$ for POA $= 0.02$, which in-turn means that the neural networks are unable to provide accurate predictions in these regions. This is typically a problem induced by an imbalanced dataset (missing data of control parameters). In order to solve this, it was chosen to regenerate



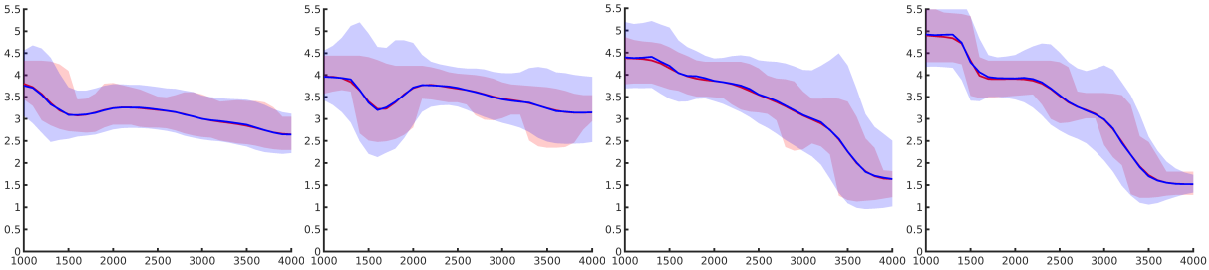
(a) **M = 0.2** (b) **M = 0.3** (c) **M = 0.4** (d) **M = 0.5**

(e) **M = 0.2** (f) **M = 0.3** (g) **M = 0.4** (h) **M = 0.5**

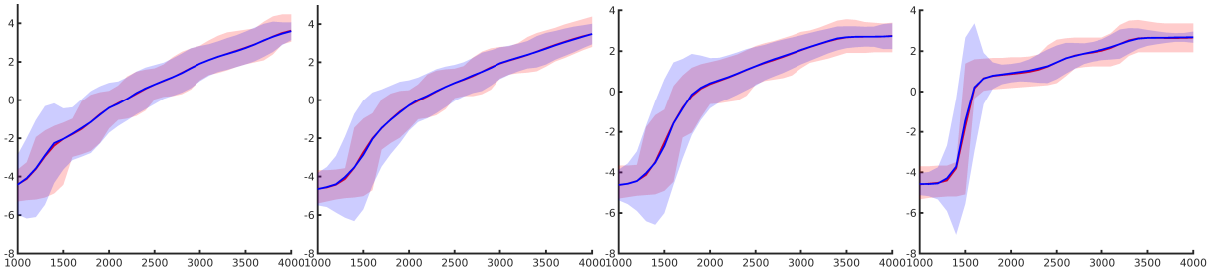
Figure 4.7: Mean values and confidence region with a probability level $P_c = 98\%$ of the impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)). The parameters POA $= 0.02$, $t = 0.5$ and $d = 0.4$ are fixed for all the graphs. The red curve and the red zone represent the conditional mean values and conditional confidence regions for given $\mathbf{w}$, using the PLoM-GKDE-based statistical metamodel for which the Silverman bandwidth has been tuned to $s = 0.1641$ for all the figures. The blue curve and the blue zone represent the conditional mean values and conditional confidence regions for given $\mathbf{w}$, using the ANN-based statistical metamodel. Horizontal axis: Frequency in Hz.

dataset $\mathcal{D}^*_{\mathbf{H}|\mathbf{W}}$.

Random vector $\mathbf{W}$ is then modeled as a multivariate random variable, that is uniform on $\mathcal{S}_{\mathbf{w}}$. Since domain $\mathcal{S}_{\mathbf{w}}$ is a product of intervals in $\mathbb{R}$, the components of random vector $\mathbf{W}$ are statistically independent. Let $\mathbf{w}^1, \ldots \mathbf{w}^{n_{\text{ar}}}$ be n_{ar} statistically independent realizations of $\mathbf{W}$. The new training dataset $\mathcal{D}^*_{\mathbf{H}|\mathbf{W}}$ is then made up of new samples $(\boldsymbol{\mu}^1, \boldsymbol{\zeta}^1), \ldots, (\boldsymbol{\mu}^\nu, \boldsymbol{\zeta}^\nu)$ for $\nu < n_{\text{ar}}$ and $\nu = 160\,000$, constructed by substituting $\mathbf{w}^j_{\text{ar}}$ and $\mathbf{w}^\ell_{\text{ar}}$ by $\mathbf{w}^j$ and $\mathbf{w}^\ell$ in Eqns. (3.15) and (3.16). For this new training dataset $\mathcal{D}^*_{\mathbf{H}|\mathbf{W}}$, Fig. 4.8 shows the conditional mean values and confidence region for given $\mathbf{w}$ with a probability level $P_c = 98\%$ of the acoustic impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)), and for M varying between



(a) **M = 0.2** (b) **M = 0.3** (c) **M = 0.4** (d) **M = 0.5**

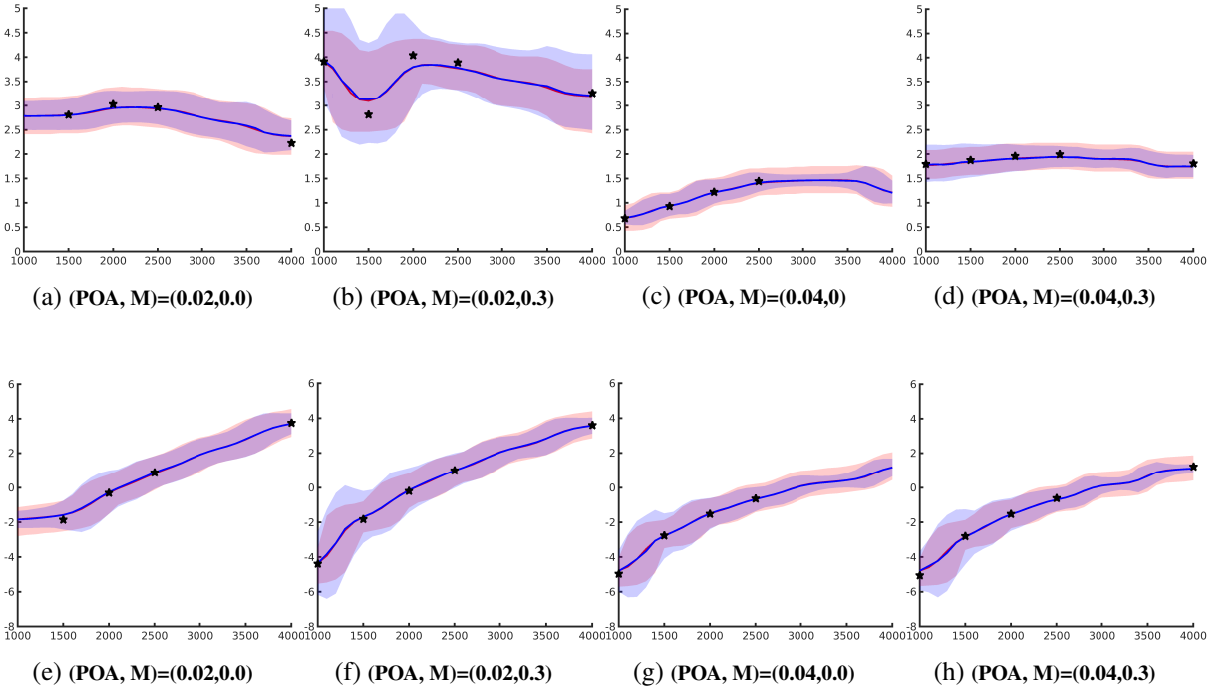


(e) **M = 0.2** (f) **M = 0.3** (g) **M = 0.4** (h) **M = 0.5**

Figure 4.8: Mean values and confidence region with a probability level $P_c = 98\%$ of the impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)). The parameters POA $= 0.02$, $t = 0.5$ and $d = 0.4$ are fixed for all the graphs. The red dashed curve and the red zone represent the conditional mean values and conditional confidence regions for given $\mathbf{w}$, using the PLoM-GKDE-based statistical metamodel. The Silverman bandwidth is tuned to $s = 0.1641$ for all the figures. The blue curve and the blue zone represent the conditional mean values and conditional confidence regions for given $\mathbf{w}$, using the ANN-based statistical metamodel. Horizontal axis: Frequency in Hz.

0.2 and 0.5. The parameters POA $= 0.02$, $t = 0.5$ and $d = 0.4$ have been fixed for all the graphs. The red curve and the red zone represent the conditional mean values and conditional confidence regions for given $\mathbf{w}$, using the PLoM-GKDE-based statistical metamodel for the Silverman bandwidth tuned to $s = 0.1641$ for all the graphs. The blue curve and the blue zone represent the conditional mean values and conditional confidence regions for given $\mathbf{w}$, using the ANN-based statistical metamodel that was trained on the regenerated dataset $\mathcal{D}^*_{\mathbf{H}|\mathbf{W}}$. It can be seen that the choice of a multidimensional uniform distribution for the control parameters solves the problem.

In order to assess the capability of the ANN-based statistical metamodel to deal with missing data (for missing values of the control parameters in the ACM dataset), its predictions are compared with the ACM values of the acoustic impedance for the values of the control parameters that exist in the ACM dataset. Figure. 4.9 shows the graphs of the conditional mean values and conditional confidence regions with a probability level $P_c = 98\%$ of the acoustic impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)), for given $\mathbf{w}$. For all the graphs, the control parameters POA and M vary as indicated in Fig. 4.9, $t = 1$ and $d = 0.4$ are fixed, and the Silverman bandwidth is tuned to $s = 0.1641$. In the initial training dataset $\mathcal{D}_{\text{acm}}$, the points are shown with black stars. The red (blue) curve and the red (blue) zone represent the conditional mean values and conditional confidence regions for given $\mathbf{w}$, using the PLoM-GKDE-based statistical metamodel (the ANN-based statistical metamodel). Figures 4.10 and 4.11 shows the conditional mean values of resistance and reactance given $\mathbf{w}$: For all the graphs in Figs.(a)



(a) **(POA, M)=(0.02,0.0)** (b) **(POA, M)=(0.02,0.3)** (c) **(POA, M)=(0.04,0)** (d) **(POA, M)=(0.04,0.3)**

(e) **(POA, M)=(0.02,0.0)** (f) **(POA, M)=(0.02,0.3)** (g) **(POA, M)=(0.04,0.0)** (h) **(POA, M)=(0.04,0.3)**

Figure 4.9: Conditional mean values and conditional confidence region with a probability level $P_c = 98\%$ of the impedance, for the resistance ((a) to (d)) and reactance ((e) to (h)) for given $\mathbf{w}$. The parameters $t = 1$ and $d = 0.4$ are fixed for all the graphs. The red curve and the red zone represent the conditional mean values and conditional confidence regions and the Silverman bandwidth tuned to $s = 0.1641$ for all the figures. The blue curve and the blue zone represent the conditional mean values and conditional confidence regions for given $\mathbf{w}$, using the ANN-based metamodel. The points in the initial training dataset $\mathcal{D}_{\text{acm}}$ are shown with black stars. Horizontal axis: Frequency in Hz.

to (e), hole diameter d is between 0.3 and 0.8, $M = 0$ and $t = 1$, and the horizontal axis is frequency in Hz. The four different blue curves represent different values of POA (between 0.02 and 0.04) as shown in the legend.

Similarly, Figures 4.12 and 4.13 shows the conditional mean values of resistance and reactance given $\mathbf{w}$, for the same values of the control parameters except for the Mach number that is $M = 0.3$. It can be seen that, concerning the conditional mean values given $\mathbf{w}$, there is a good match between the PLoM-GKDE-based statistical metamodel and the ANN-based statistical metamodel.

4.6 Conclusions and discussions

In this chapter, an ANN-based statistical metamodel has been developed for the case of missing data points in the small ACM dataset and for which the control parameters are POA, Mach number, plate thickness, hole diameter and frequency. The PLoM-GKDE-based statistical metamodel is used to mitigate missing data in the ACM dataset that allows for improved training of the ANN-based statistical metamodel. Concerning the results obtained, the resistance and reactance behave as expected with respect to the geometry and flow parameters:

(i) increasing POA yields a reduction of the resistance.

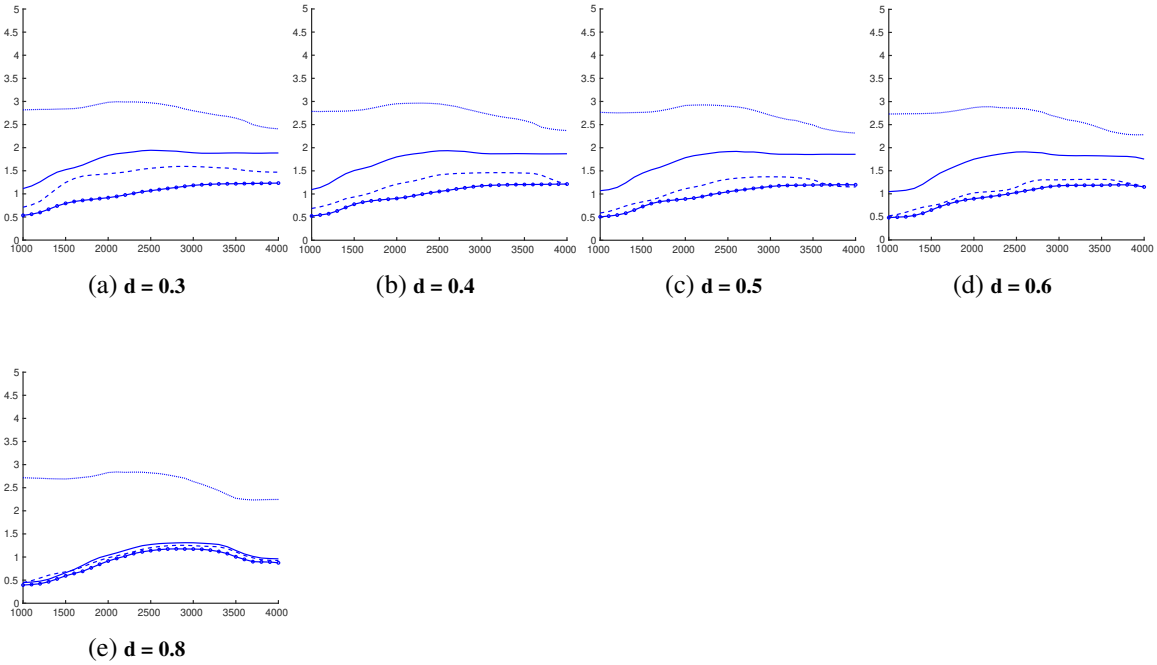
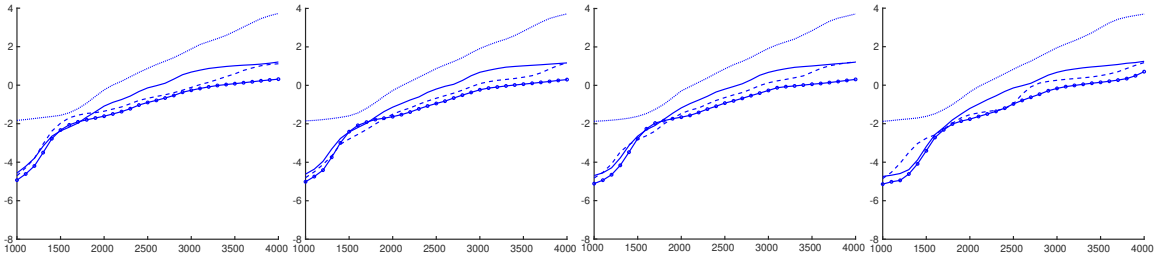


Figure 4.10: Conditional mean values of resistance given **w**. The hole diameter d varies from ((a) to (e)) as shown. The parameters $M = 0$ and $t = 1$ are fixed for all the graphs. The four different blue curves represent different values of POA (between 0.02 and 0.04) as shown in the legend. Horizontal axis: Frequency in Hz.

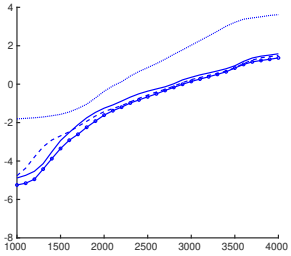
(ii) increasing the hole diameter yields a reduction of the resistance and an increase of the reactance.

(ii) increasing Mach number yields an increase of the resistance and a slight reduction of the reactance.

The novelty of this chapter lies in the methodology used to deal with a small imbalanced dataset with missing data and also in extending the statistical metamodel and statistical ANN-based metamodel of previous chapters to positive Mach.



(a) **d = 0.3** (b) **d = 0.4** (c) **d = 0.5** (d) **d = 0.6**



(e) **d = 0.8**

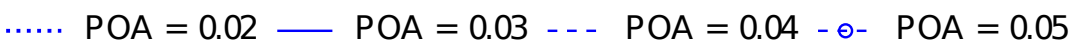
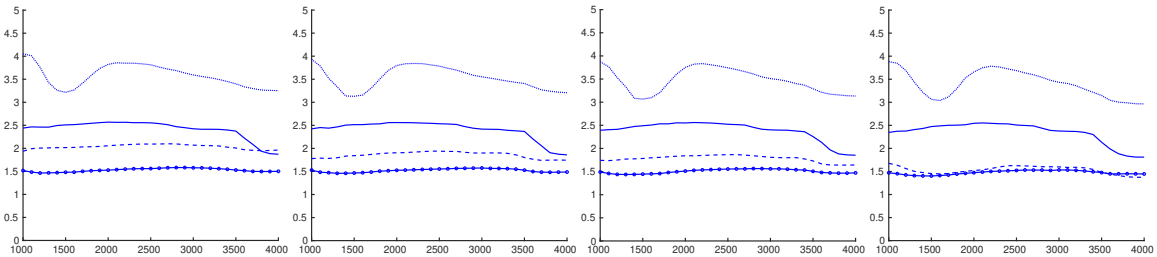
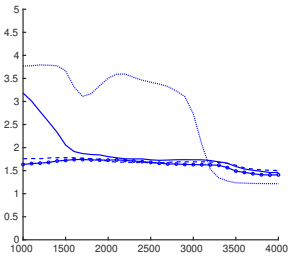


Figure 4.11: Conditional mean values of reactance given **w**. The hole diameter d is varied from ((a) to (e)) as shown. The parameters $M = 0$ and $t = 1$ are fixed for all the graphs. The four different blue curves represent different values of POA (between 0.02 and 0.04) as shown in the legend. Horizontal axis: Frequency in Hz.



(a) **d = 0.3** (b) **d = 0.4** (c) **d = 0.5** (d) **d = 0.6**



(e) **d = 0.8**

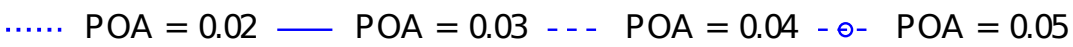


Figure 4.12: Conditional mean values of resistance given **w**. The hole diameter d is varied from ((a) to (e)) as shown. The parameters $M = 0.3$ and $t = 1$ are fixed for all the graphs. The four different blue curves represent different values of POA (between 0.02 and 0.04) as shown in the legend. Horizontal axis: Frequency in Hz.

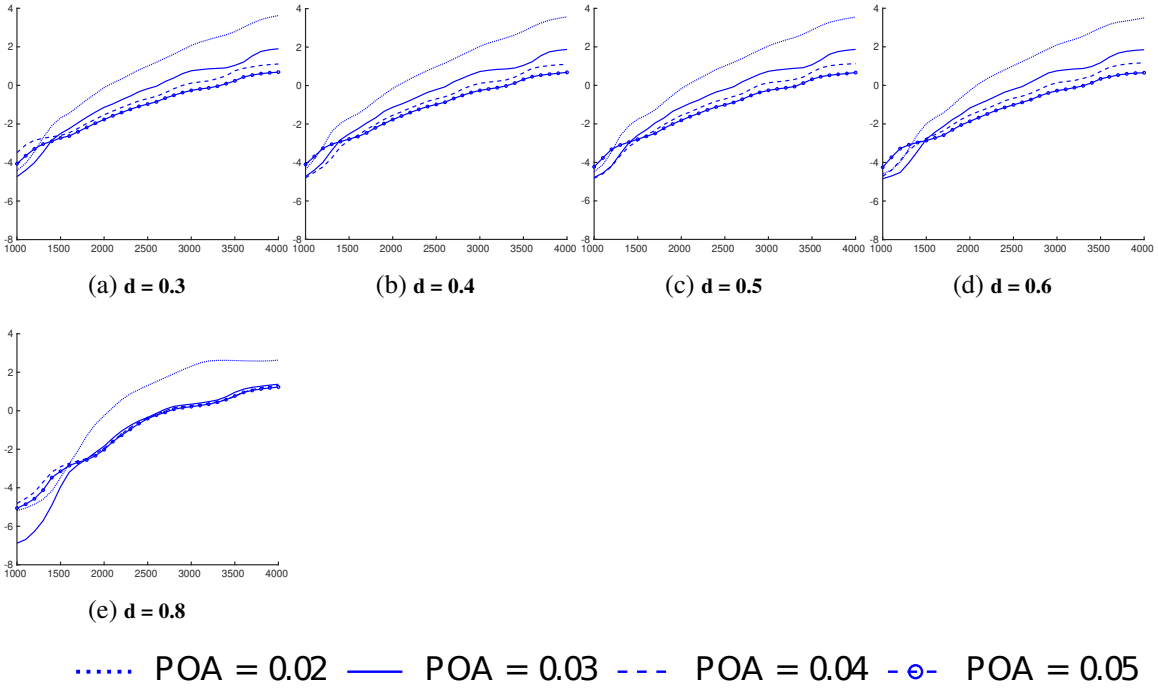


Figure 4.13: Conditional mean values of reactance given **w**. The hole diameter d is varied from ((a) to (e)) as shown. The parameters $M = 0.3$ and $t = 1$ are fixed for all the graphs. The four different blue curves represent different values of POA (between 0.02 and 0.04) as shown in the legend. Horizontal axis: Frequency in Hz.

Chapter 5

Supplementary insights into training Neural Networks

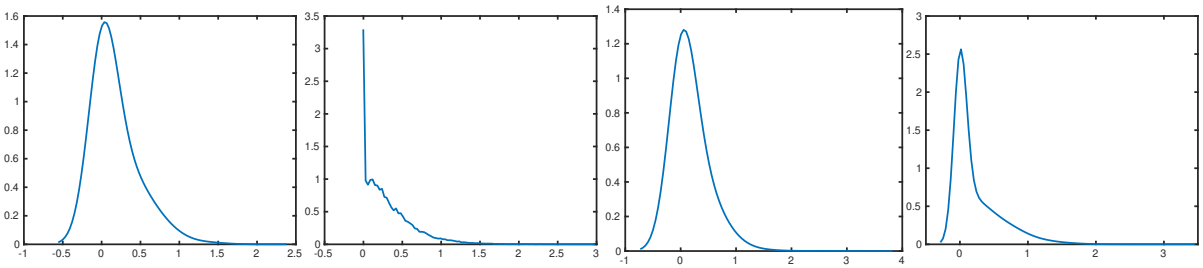
In this chapter, we give supplementary insights concerning the implementation of the feedforward networks used in Chapters 3 and 4. It concerns the network architecture, the initialization, and the training algorithms.

5.1 Network architecture and initialization

The selection of the network architecture was guided by three primary criteria: (1) the size of the training database, (2) the network initialization, and (3) the convergence with respect to loss minimization while maximizing accuracy.

Concerning the architecture and in order to circumvent computationally expensive hyperparameter optimization [49] of the neural networks, a trial-error approach is employed to achieve a satisfactory architecture, using cross-validation, over a set of randomly chosen architecture.

Concerning the initialization of the neural network, which is important for a good convergence, the method used consists in examining the probability density function of the output of each hidden layer. A single forward pass, using a small subset of the training dataset, is done after initialization. The random seed state and the number of neurons are modified, until the output of the hidden layer satisfies two conditions: (1) it is no longer saturated at extreme values (*i.e.*, at 0 and 1 due to the use of ReLU activation function), and (2) the hidden layer is capable of generalizing by drawing values from a non-Gaussian probability distribution. This step fully incorporates the first two criteria and partially addresses the third criterion. For a network with four hidden layers, Figs. 5.1a to 5.1d show the probability density estimate of the output of each hidden layer as a function of the layer number. It shows that the network is not getting saturated during training (if not, we will not obtain a density, but we would have concentrated measures close to a Dirac measure).



(a) **hidden layer 1 - 20 neurons** (b) **hidden layer 2 - 250 neurons** (c) **hidden layer 3 - 75 neurons** (d) **hidden layer 4 - 25 neurons**

Figure 5.1: For a network with four hidden layers, probability density estimate of the output of each hidden layer as a function of the layer number ((a) to (d)).

5.2 Optimization algorithms for neural network training

For performing the training of neural networks introduced in Chapters 3 and 4, three optimization algorithms for network parameters have been tested. The first two algorithms are first-order optimization methods: ADAptive Moment estimation (ADAM) and Stochastic Gradient Descent with Momentum (SGDM). Additionally, a second-order algorithm, the Levenberg-Marquardt algorithm with an approximated Hessian estimation, has been considered. Initially, ADAM and SGDM were not the priority due to the large number of networks to be trained (refer to Section 3.4 of Chapter 3), requiring close monitoring of the training the networks in order to adjust the parameters. On the other hand, the Levenberg-Marquardt algorithm does not require any supervision but, as a drawback GPU is not used with such an algorithm. Therefore, a stochastic formulation of the Levenberg-Marquardt algorithm (SLM) is proposed to address GPU limitations by using minibatches. In theory, the SLM allowed for the curvature information of the loss function to be considered, thereby reducing training time and eliminating the need for monitoring network convergence. This is achieved through a blending factor within the SLM, enabling the network to alternate between classic gradient descent and a Gauss-Newton algorithm. A line search algorithm automatically updates the network and the SLM parameters, using a local gradient information. Although in the numerical experiments, the SLM algorithms performed well, when compared with the two first-order algorithms using dynamic learning rates or learn rate schedulers, they do not offer significant advantages. The subsequent sections will summarize the development and implementation of the SLM, along with a brief recap of the other two algorithms, followed by a comparison of their performance.

5.2.1 Stochastic Levenberg-Marquardt algorithm

The Levenberg-Marquardt (LM) algorithm [71, 70], which alternates between Gauss-Newton algorithm and the gradient-descent method, is an efficient approach for neural network training because second-order optimization method use the curvature information for adapting the gradients. Unfortunately, this method requires the estimation of a Hessian matrix, which can be computationally expensive as its memory complexity evolves with $\mathcal{O}(n^2)$, and is practically impossible for large databases. An algorithm

simplification is presented to adapt LM to large database by using a stochastic Levenberg-Marquardt algorithm, which differs from proposed SLM versions in [73, 72]. Let the loss function to be minimized be written as

$$J(\boldsymbol{\theta}) = \sum_{j=1}^{\nu} g(f(\mathbf{w}^j;\boldsymbol{\theta}), y(\mathbf{w}^j))\,, \tag{5.1}$$

where $f(\mathbf{w}^j;\boldsymbol{\theta})$ is a real-valued hyperparameter corresponding to the output of the network, which is either hyperparameter $\{\boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}^j;\boldsymbol{\theta})\}_k$ or $\{\boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}^j;\boldsymbol{\theta})\}_k$ defined in Section 3.5, where $y(\mathbf{w}^j)$ corresponds to the target for given input vector $\mathbf{w}^j$, which is either $\{\boldsymbol{\mu}^j\}_k$ or $\{\boldsymbol{\zeta}^j\}_k$, and where $\boldsymbol{\theta}$ is the network parameters to be optimized for minimizing the loss function. For example, $g(u,y) = (y-u)^2$ define the usual least-square regression problem. The gradient of the cost function $J(\boldsymbol{\theta})$ is written as

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_{j=1}^{\nu} g'(f(\mathbf{w}^j;\boldsymbol{\theta}), y(\mathbf{w}^j))\nabla_{\boldsymbol{\theta}} f(\mathbf{w}^j;\boldsymbol{\theta})\,, \tag{5.2}$$

where g' is the partial derivative of $(u,\, y) \mapsto g(u,\, y)$ with respect to u. The first-order approximation of $\boldsymbol{\theta} \mapsto g'(f(\mathbf{w}^j;\boldsymbol{\theta}), y(\mathbf{w}^j))$ at point $\boldsymbol{\theta}_0$ is written as,

$$g'(f(\mathbf{w}^j;\boldsymbol{\theta}), y(\mathbf{w}^j)) = g'(f(\mathbf{w}^j;\boldsymbol{\theta}_0), y(\mathbf{w}^j)) + g''(f(\mathbf{w}^j;\boldsymbol{\theta}_0), y(\mathbf{w}^j))\nabla_{\boldsymbol{\theta}} f(\mathbf{w}^j;\boldsymbol{\theta}_0)^T(\boldsymbol{\theta}-\boldsymbol{\theta}_0) \tag{5.3}$$

Substituting Eqn. (5.3) into (5.2) yields,

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = &\sum_{j=1}^{\nu} g'(f(\mathbf{w}^j;\boldsymbol{\theta}_0), y(\mathbf{w}^j))\nabla_{\boldsymbol{\theta}} f(\mathbf{w}^j;\boldsymbol{\theta}) \\ &+ \sum_{j=1}^{\nu} g''(f(\mathbf{w}^j;\boldsymbol{\theta}_0), y(\mathbf{w}^j))\nabla_{\boldsymbol{\theta}} f(\mathbf{w}^j;\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} f(\mathbf{w}^j;\boldsymbol{\theta}_0)^T(\boldsymbol{\theta}-\boldsymbol{\theta}_0)\end{aligned} \tag{5.4}$$

Using the first-order approximation of $\boldsymbol{\theta} \mapsto f(\mathbf{w}^j;\boldsymbol{\theta})$, the equation $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = 0$ is rewritten as:

$$\boldsymbol{\theta} = \boldsymbol{\theta}_0 - [H(\boldsymbol{\theta}_0)]^{-1}\mathbf{d}(\boldsymbol{\theta}_0)\,, \tag{5.5}$$

where $\mathbf{d}(\boldsymbol{\theta}_0)$ and $[H(\boldsymbol{\theta}_0)]$ are defined as:

$$d(\boldsymbol{\theta}_0) = \sum_{j=1}^{\nu} g'(f(\mathbf{w}^j;\boldsymbol{\theta}_0), y(\mathbf{w}^j))\nabla_{\boldsymbol{\theta}} f(\mathbf{w}^j;\boldsymbol{\theta}_0) = \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0)\,, \tag{5.6}$$

$$[H(\boldsymbol{\theta}_0)] = \sum_{j=1}^{\nu} g''(f(\mathbf{w}^j;\boldsymbol{\theta}_0), y(\mathbf{w}^j))\nabla_{\boldsymbol{\theta}} f(\mathbf{w}^j;\boldsymbol{\theta}_0)\nabla_{\boldsymbol{\theta}} f(\mathbf{w}^j;\boldsymbol{\theta}_0)^T\,, \tag{5.7}$$

in which matrix $[H(\boldsymbol{\theta}_0)]$ is assumed to be invertible. Matrix $[H(\boldsymbol{\theta}_0)]$ is an approximation of the Hessian of $J(\boldsymbol{\theta}_0)$, which can be written as,

$$[H(\boldsymbol{\theta}_0)] = [U(\boldsymbol{\theta}_0)][G(\boldsymbol{\theta}_0)][U(\boldsymbol{\theta}_0)]^T\,, \tag{5.8}$$

where, for $j,\ j' = 1, \ldots, \nu$

$$[G(\boldsymbol{\theta}_0)]_{jj'} = g''(f(\mathbf{w}^j; \boldsymbol{\theta}_0), y(\mathbf{w}^j))\delta_{jj'}\,, \tag{5.9}$$

and

$$[U(\boldsymbol{\theta}_0)] = [\nabla_{\boldsymbol{\theta}} f(\mathbf{w}^1; \boldsymbol{\theta}_0) \ldots \nabla_{\boldsymbol{\theta}} f(\mathbf{w}^\nu; \boldsymbol{\theta}_0)]\,. \tag{5.10}$$

The iterative scheme $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - [H(\boldsymbol{\theta}_i)]^{-1}\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_i)$ corresponds to a Newton scheme. The iterative Levenberg-Marquardt scheme is obtained by introducing a blending factor λ_i as follows

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \left([H(\boldsymbol{\theta}_i)] + \lambda_i[I]\right)^{-1} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_i) \quad , \quad \lambda_i \gg 1\,, \tag{5.11}$$

Using the Sherman-Morrison-Woodbury formula for matrix inversion, the approximation of Hessian under the inversion can be simplified as

$$([H(\boldsymbol{\theta}_i)] + \lambda_i[I])^{-1} = \frac{1}{\lambda_i}[I] - \frac{1}{\lambda_i^2}[U(\boldsymbol{\theta}_i)][G(\boldsymbol{\theta}_i)]\left([I_\nu] + \frac{1}{\lambda_i}[U(\boldsymbol{\theta}_i)]^T[U(\boldsymbol{\theta}_i)][G(\boldsymbol{\theta}_i)]\right)^{-1}[U(\boldsymbol{\theta}_i)]^T\,, \tag{5.12}$$

where $[I_\nu]$ is an identity matrix of size ν. Substituting Eqn. (5.12) in (5.11) we obtain

$$\begin{aligned}\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i &- \frac{1}{\lambda_i}\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_i) \\ &+ \frac{1}{\lambda_i^2}[U(\boldsymbol{\theta}_i)][G(\boldsymbol{\theta}_i)]\left([I_\nu] + \frac{1}{\lambda_i}[U(\boldsymbol{\theta}_i)]^T[U(\boldsymbol{\theta}_i)][G(\boldsymbol{\theta}_i)]\right)^{-1}[U(\boldsymbol{\theta}_i)]^T\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_i)\,,\end{aligned} \tag{5.13}$$

which can then be rewritten as:

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \frac{1}{\lambda_i}\mathbf{d}(\boldsymbol{\theta}_i) + \frac{1}{\lambda_i^2}[U(\boldsymbol{\theta}_i)][G(\boldsymbol{\theta}_i)][V(\boldsymbol{\theta}_i)]^{-1}[U(\boldsymbol{\theta}_i)]^T\mathbf{d}(\boldsymbol{\theta}_i) \tag{5.14}$$

where $\mathbf{d}(\boldsymbol{\theta}_i) = \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_i)$ and

$$[V(\boldsymbol{\theta}_i)] = [I_\nu] + \frac{1}{\lambda_i}[U(\boldsymbol{\theta}_i)]^T[U(\boldsymbol{\theta}_i)][G(\boldsymbol{\theta}_i)]\,, \tag{5.15}$$

where $[I_\nu]$ is the $(\nu \times \nu)$ identity matrix. Therefore the final update can be written as

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \frac{1}{\lambda_i}\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_i) + \frac{1}{\lambda_i^2}[\mathbf{d}_4(\boldsymbol{\theta}_i)]\,, \tag{5.16}$$

where $\mathbf{d}_1(\boldsymbol{\theta}_i), \ldots, \mathbf{d}_4(\boldsymbol{\theta}_i)$ are successively calculated as

$$\mathbf{d}_1(\boldsymbol{\theta}_i) = [U(\boldsymbol{\theta}_i)]^T\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_i) \tag{5.17}$$

$$\mathbf{d}_2(\boldsymbol{\theta}_i) = [V(\boldsymbol{\theta}_i)]^{-1}\mathbf{d}_1(\boldsymbol{\theta}_i) \tag{5.18}$$

$$\mathbf{d}_3(\boldsymbol{\theta}_i) = [G(\boldsymbol{\theta}_i)]\mathbf{d}_2(\boldsymbol{\theta}_i) \tag{5.19}$$

$$\mathbf{d}_4(\boldsymbol{\theta}_i) = [U(\boldsymbol{\theta}_i)]\mathbf{d}_3(\boldsymbol{\theta}_i) \tag{5.20}$$

It should be noted that this algorithm requires the construction and the inversion of $(\nu \times \nu)$ full matrix $[V(\boldsymbol{\theta}_i)]$. Such operations require sufficient amount of memory and consequently, due to memory limitations on GPU, such a Levenberg-Marquardt algorithm is usually not implemented on GPU. In order to circumvent such limitations, a Stochastic Levenberg-Marquardt (SLM) algorithm is implemented by introducing m_{batch} minibatches of size N_{batch} ($N_{\text{batch}} < \nu$) such that $\nu = m_{\text{batch}} \times N_{\text{batch}}$, so that there are m_{batch} iterations for every epoch. For each minibatch indexed by $b = 1, \ldots, m_{\text{batch}}$, the previous developments of the Levenberg-Marquardt is implemented, with the following cost function $J_b(\boldsymbol{\theta}) = \sum_{j=j_{b,0}}^{j_{b,1}} g(f(\mathbf{w}^j; \boldsymbol{\theta}), y(\mathbf{w}^j))$, in which $j_{b,0} = (b-1)\, N_{\text{batch}} + 1$ and $j_{b,1} = b \times N_{\text{batch}}$. We then use Eqns. (5.11), (5.12), (5.15) and (5.16) to (5.20) with data points $\mathbf{w}_{\text{ar}}^{j_{b,0}}, \ldots, \mathbf{w}_{\text{ar}}^{j_{b,1}}$. Consequently, for each minibatch, the size of matrix $[V(\boldsymbol{\theta}_i)]$ is $N_{\text{batch}} \times N_{\text{batch}}$. Each minibatch is initialized with the parameter $\boldsymbol{\theta}^*$ obtained from the previous minibatch optimization. Such parameters $\boldsymbol{\theta}^*$ are not necessarily the minimum of $J_{b-1}(\boldsymbol{\theta}^*)$. For each minibatch, the Levenberg-Marquardt iterations are stopped as soon as the parameter $\boldsymbol{\theta}$ is found such that $J_b(\boldsymbol{\theta}) < J_b(\boldsymbol{\theta}^*)$. This parameter $\boldsymbol{\theta}$ is the $\boldsymbol{\theta}^*$ that is passed onto the next minibatch. It should be noted that if we have the condition $J_b(\boldsymbol{\theta}) < J_{b-1}(\boldsymbol{\theta}^*)$, then we fall into an infinite loop.

5.2.2 ADAptive Moment estimate algorithm

Training a neural network using ADAM algorithm [67] solves for the network parameter $\boldsymbol{\theta}$ update but with an added momentum vector. It keeps an element-wise moving average of both the parameter gradient vector $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ and its element-wise squared value. We introduce

$$\mathbf{m}_{i+1} = \beta_1 \mathbf{m}_i + (1 - \beta_1) \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_i) \tag{5.21}$$

$$\{\mathbf{v}_{i+1}\}_k = \beta_2 \{\mathbf{v}_i\}_k + (1 - \beta_2) \{\nabla J(\boldsymbol{\theta}_i)\}_k^2 , \tag{5.22}$$

where $\mathbf{m}_i$ is the momentum vector at i-th epoch, $\beta_1 = 0.9$ is a gradient decay factor, $\mathbf{v}_i$ is the velocity vector at i-th epoch and $\beta_2 = 0.99$ is a square of the gradient decay factor. ADAM updates the network parameters $\boldsymbol{\theta}_i$ to $\boldsymbol{\theta}_{i+1}$ written as

$$\{\boldsymbol{\theta}_{i+1}\}_k = \{\boldsymbol{\theta}_i\}_k - \frac{\gamma_i \{\mathbf{m}_i\}_k}{\left(\sqrt{\{\mathbf{v}_i\}_k} + \epsilon\right)} , \tag{5.23}$$

where $\epsilon = 1 \times 10^{-8}$, and a learning rate γ_i scheduler that adjusts the learning rate over the course of training as

$$\gamma_i = \max\left\{\gamma_o \, \alpha^{(i-1)/\Delta}, \gamma_{\text{min}}\right\} , \tag{5.24}$$

where γ_o is the initial learning rate (default value is $\gamma_o = 0.001$), $\alpha = 0.95$ is the decay factor, $\Delta = 5$ is the decay period, and $\gamma_{\text{min}} = 1 \times 10^{-7}$ is the minimum leaning rate.

5.2.3 Stochastic gradient descent with momentum algorithm

The standard gradient descent algorithm iteratively updates the network parameters (weights and biases) $\boldsymbol{\theta}$ to minimize the loss function. It achieves this by taking incremental steps, determined by the negative gradient of the loss function, in the direction that minimizes the loss. The standard gradient-descent algorithm processes the entire dataset in each iteration. In contrast, the stochastic gradient-descent algorithm computes gradients $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ and updates parameters using a minibatch of the training data in each iteration. The stochastic gradient-descent algorithm may oscillate along the path of the steepest descent towards the optimum. One approach to mitigate this oscillation [74] is to incorporate a momentum term into the parameter update process,

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \gamma_i \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_i) + \beta(\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}) \,, \tag{5.25}$$

where i is the epoch, $\beta = 0.95$ is the momentum γ_i is the learning rate defined by the learn rate scheduler as,

$$\gamma_i = \max\Big\{ \frac{\gamma_o}{1 + \alpha(i-1)/\Delta}, \gamma_{\text{min}} \Big\} \,, \tag{5.26}$$

where γ_o is the initial learning rate (default value is $\gamma_o = 0.001$), $\alpha = 0.1$ is the decay factor, $\Delta = 5$ is the decay period, and $\gamma_{\text{min}} = 1 \times 10^{-7}$ is the minimum leaning rate.

5.2.4 Comparison of the optimization algorithms

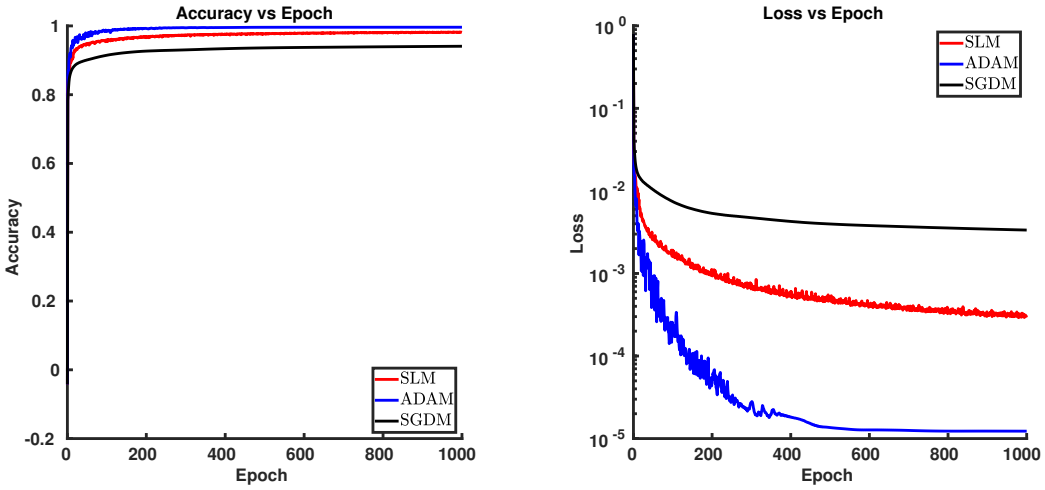
The convergence of the neural network training is evaluated using loss function minimization given by the Eqn. (3.18) in Chapter 3 as,

$$\mathcal{J}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \frac{1}{2} \left(\sum_{j=1}^{\nu} \|\boldsymbol{\mu}^j - \boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}_{\text{ar}}^j; \boldsymbol{\theta}_1)\|^2 + \sum_{j=1}^{\nu} \|\boldsymbol{\zeta}^j - \boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}_{\text{ar}}^j; \boldsymbol{\theta}_2)\|^2 \right) \quad . \tag{5.27}$$

Similarly, the accuracy can also be evaluated using,

$$\mathcal{A}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \frac{1}{2} \left(\frac{1}{\nu} \sum_{j=1}^{\nu} \Big(1 - \|\boldsymbol{\mu}^j - \boldsymbol{\mu}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}_{\text{ar}}^j; \boldsymbol{\theta}_1)\|\Big) + \frac{1}{\nu} \sum_{j=1}^{\nu} \Big(1 - \|\boldsymbol{\zeta}^j - \boldsymbol{\zeta}_{\mathbf{H}|\mathbf{W}}(\mathbf{w}_{\text{ar}}^j; \boldsymbol{\theta}_2)\|\Big) \right) \quad , \tag{5.28}$$

Figure 5.2 shows the comparison of the evolution of the (a) accuracy and (b) loss, over epochs for the three optimization algorithms. It can be seen that ADAM optimizer is able to attain the best accuracy and the lowest loss of the three models, followed SLM optimizer and SGDM optimizer respectively. In Table 5.1, the average time taken per epoch for the three algorithms, is presented, in which the first column represents the training of the networks for conditional mean vector and the second column represents the training of the networks for conditional covariance matrix. It can be seen that ADAM optimizer is the most efficient optimizer and was thus used for the training of the networks.



(a) Accuracy vs. epoch

(b) Loss vs. epoch

Figure 5.2: Comparison of (a) accuracy over epoch and (b) loss over epoch. The blue curve represents ADAM optimization, the black curve represents SGDM optimization and the red curve represents SLM optimization.

	Mean	Covariance
ADAM	25.5	23.19
SGDM	25.05	24.88
SLM	47.31	56.76

Table 5.1: The table shows the average time taken in seconds for one epoch, for each of the optimizer algorithms. The two columns shown are for the training of mean value and covariance matrix components.

Conclusion: contributions and novelty

This thesis is devoted to the construction of a statistical metamodel of a parameterized aeroacoustic liner impedance, that is robust with respect to uncertainties. As a matter of fact, design of acoustic liners must account for in service variabilities related to performance of the aircraft. For this thesis we have focused on a liner that is adapted to low-frequency tonal noise.

The first problem that has been addressed is limited availability of data points (small datasets). High-fidelity computational models used for obtaining frequency-sampled liner acoustic impedance are computationally expensive, limiting their feasibility for numerous evaluations across design parameters. Nevertheless, we have used such a computationally expensive high-fidelity model, referred to as aeroacoustic computational model (ACM) to generate a small training dataset. We have then made use of a learning process tool based on Probabilistic Learning on Manifolds (PLoM) to learn the *a priori* joint probability distribution function of the data. From the ACM dataset, this approach allows for generating a large database for the liner acoustic impedance, which are constituted of realizations for the control parameters as well as the quantity of interest (QoI) that is the frequency-sampled liner impedance. For the first part of the thesis, the control parameters have been chosen as the POA and SPL with Mach number being kept at zero, corresponding to the ground configuration. In order to account for uncertainties induced by the latent parameters (uncontrolled parameters), a probability model has been developed. This has been done by modeling the QoIs as random variables for which an *a priori* probability distribution has been constructed. Although the aeroacoustic simulation has been conducted with a large high-fidelity computational model, some approximations have been introduced, generating model errors. To account for these model errors, a probability model has been developed and directly integrated into the training dataset, which in turn introduce the model errors into the random variables. To calibrate the model errors, we have used dimensionless experiments available from the open literature. In addition, we have also introduced a sensitivity parameter to the level of model uncertainties. Despite the fact that only a very small amount of data was available, we have succeeded in proposing a robust statistical metamodel, based on the use of a learned dataset and nonparametric conditional statistics, which is novel and whose predictions are consistent. This statistical metamodel furnishes a confidence region, which gives an information about the level of uncertainties about the aeroacoustic liner impedance as a function of the frequency and the control parameters.

A statistical ANN-based metamodel of the frequency-sampled liner acoustic impedance has then been presented. The construction of this metamodel uses a PCA to construct a statistical reduced representation of the frequency-sampled vector of the random log-resistance and the random reactance. For fitting

the statistical ANN-based metamodel, a big training dataset has been generated using PLoM. A prior conditional probability distribution of the reduced representation, given the control parameters, has then been constructed and assumed to be Gaussian, yielding a multivariate log-normal distribution for the resistance and a multivariate Gaussian distribution for the reactance. For the ANN-based metamodels, each hyperparameter of the conditional probabilistic model has been represented by a fully-connected feedforward neural network. As the reduced representation has been modeled by a centered and normalized random vector, some constraints have to be taken into account in the minimization of the negative log-likelihood when training the parameters (biases and weights) of the neural network. The constrained problem has been transformed in an unconstrained one, requiring the construction of a second training dataset to estimate the conditional mean vectors and conditional covariance matrices for which the learned realizations have been generated using PLoM. We have also presented complementary information about the training of statistical ANN-based metamodel. For instance, a stochastic formulation of Levenberg-Marquardt algorithm for neural network training, which is an optimization algorithm making use of second-order gradient information, has been carried out, and its results have been compared with other first-order gradient algorithms. This metamodel is intended to serve as an alternative representation of the statistical metamodel, offering increased versatility and user-friendliness (from an engineering perspective).

As a last step, we have presented the construction of the statistical metamodel and its ANN-based representation for positive Mach number. For this step, the control parameters have been chosen as POA, Mach number, perforated plate thickness, hole diameter, and frequency. The challenges have consisted in constructing the statistical metamodels for small imbalanced dataset, which is further complicated by the presence of missing data.

The main novelties of this thesis are as follows.

- We have presented a novel methodology, which is independent of the choice of the ACM (more or less simplified) and of the choice of experiments used to estimate the level of uncertainties.
- A statistical PLoM-GKDE-based metamodel and a statistical ANN-based metamodel of liner acoustic impedance have been developed, which can be used as low-computational cost metamodels to predict the conditional statistical mean value and the conditional confidence region of the random liner impedance given any value of the control parameters.
- Using the statistical ANN-based metamodel, the gradients of the mean values and the confidence regions can also easily be derived using classical backpropagation algorithms for very cheap computational cost, integrating into design optimization of liners.
- Concerning the presented application, the experimental data and the aeroacoustic computation could be replaced by others.
- Finally, PLoM and nonparametric conditional statistics has been used to mitigate any imbalanced dataset, which was further complicated by the presence of missing data.

Appendix A

Summary of the probabilistic learning on manifolds (PLoM) algorithm and its parameterization

In this appendix, we summarize the PLoM algorithm in order to facilitate the reading of this thesis. The following text is a reproduction of the one presented in [75].

The probabilistic learning is a very active domain of research for constructing surrogate models (see, for instance [79, 77, 76, 78, 37, 36, 23]). Probabilistic learning on manifolds (PLoM) is a tool in computational statistics, introduced in [21] and which can be viewed as a tool for scientific machine learning. The PLoM approach has specifically been developed for the small dataset cases [21, 38, 39, 80, 22]. The method avoids the scattering of learned realizations associated with the probability distribution in order to preserve its concentration in the neighborhood of the random manifold defined by the parameterized computational model. This method allows for solving unsupervised and supervised problems under uncertainty for which the training datasets are small. This situation is encountered in many problems of physics and engineering science with expensive function evaluations. The exploration of the admissible solution space in these situations is thus hampered by available computational resources. Several extensions have been proposed to take into account implicit constraints induced by physics, computational models, and measurements [44, 45, 46], to reduce the stochastic dimension using a statistical partition approach [23], and to update the prior probability distribution by a target dataset whose points are, for instance, experimental realizations of the system observations [47]. Consequently, PLoM constrained by a stochastic computational model and statistical moments or samples and realizations allows performing probabilistic learning inference and constructing predictive statistical surrogate models for large parameterized stochastic computational models. This last capability of PLoM can also be viewed as an alternative method to the Bayesian inference for the high dimension [84, 85, 82, 90, 87, 86, 81, 83, 89, 88] and is a complementary approach to existing methods in machine learning for sampling distributions on manifolds under constraints (although a Bayesian inference methodology has also been developed using the probabilistic learning on manifolds for the high dimensions [80]). PLoM has successfully been adapted to tackle these challenges for several related problems including

nonconvex optimization under uncertainty [95, 25, 26, 96, 94, 93, 91, 92], fracture paths in random composites [97], ultrasonic transmission technique in cortical bone microstructures [80], updating digital twins under uncertainties [98], updating of under observed dynamical system [99], calculation of the Sobol indices [100], dynamic monitoring [101], and surrogate modeling of structural seismic response [102].

The PLoM approach, starts from a training set $\mathcal{D}_d$ made up of a relatively small number N_d of points. It is assumed that $\mathcal{D}_d$ is generated with an underlying stochastic manifold related to a $\mathbb{R}^n$-valued random variable $\mathbf{X} = (\mathbf{Q}, \mathbf{W})$, defined on a probability space $(\Theta, \mathcal{T}, \mathcal{P})$, in which $\mathbf{Q}$ is the quantity of interest that is a $\mathbb{R}^{n_q}$-random variable, where $\mathbf{W}$ is the control parameter that is a $\mathbb{R}^{n_w}$-random variable, and where $n = n_q + n_w$. Another $\mathbb{R}^{n_u}$- valued random variable $\mathbf{U}$ defined on $(\Theta, \mathcal{T}, \mathcal{P})$ can also be considered, which is an uncontrolled parameter and/or a noise. Random variable $\mathbf{Q}$ is assumed to be written as $\mathbf{Q} = \mathbf{f}(\mathbf{U}, \mathbf{W})$ in which the measurable mapping $\mathbf{f}$ is not explicitly known. The joint probability distribution $P_{\mathbf{W},\mathbf{U}}(d\mathbf{w}, d\mathbf{u})$ of $\mathbf{W}$ and $\mathbf{U}$ is assumed to be given. The non-Gaussian probability measure $P_{\mathbf{X}}(\mathbf{x}) = P_{\mathbf{Q},\mathbf{W}}(d\mathbf{q}, d\mathbf{w})$ of $\mathbf{X} = (\mathbf{Q}, \mathbf{W})$ is concentrated in a region of $\mathbb{R}^n$ for which the only available information is the cloud of the points of training set $\mathcal{D}_d$. The PLoM method makes it possible to generate the learned set $\mathcal{D}_{\text{ar}}$ for $\mathbf{X}$ whose $n_{\text{MC}} \gg N_d$ points (learned realizations) are generated by the non-Gaussian probability measure that is estimated using the training set. The concentration of the probability measure is preserved thanks to the use of a diffusion-maps basis that allows to enrich the available information from the training set. Using the learned set $\mathcal{D}_{\text{ar}}$, PLoM allows for carrying out any conditional statistics such as $\mathbf{w} \mapsto E\{\boldsymbol{\xi}(\mathbf{Q})|\mathbf{W} = \mathbf{w}\}$ from $\mathcal{C}_w$ in $\mathbb{R}^{n_\xi}$, in which $\boldsymbol{\xi}$ is a given measurable mapping from $\mathbb{R}^{n_q}$ into $\mathbb{R}^{n_\xi}$, that is to say to construct statistical surrogate models (metamodels) in a probabilistic framework. The training dataset $\mathcal{D}_d$ is made up of the N_d independent realizations $\mathbf{x}_d^j = (\mathbf{q}_d^j, \mathbf{w}_d^j)$ in $\mathbb{R}^n = \mathbb{R}^{n_q} \times \mathbb{R}^{n_w}$ for $j \in \{1, \ldots, N_d\}$ of random variable $\mathbf{X} = (\mathbf{Q}, \mathbf{W})$. The PLoM method allows for generating the learned dataset $\mathcal{D}_{\text{ar}}$ made up of $N_{\text{ar}} \gg N_d$ learned realizations $\{\mathbf{x}_{\text{ar}}^\ell, \ell = 1, \ldots, N_{\text{ar}}\}$ of random vector $\mathbf{X}$. As soon as the learned dataset has been constructed, the learned realizations for $\mathbf{Q}$ and $\mathbf{W}$ can be extracted as $(\mathbf{q}_{\text{ar}}^\ell, \mathbf{w}_{\text{ar}}^\ell) = \mathbf{x}_{\text{ar}}^\ell$ for $\ell = 1, \ldots, N_{\text{ar}}$.

(A.1) Reduced representation. The N_d independent realizations $\{\mathbf{x}_d^j, j = 1, \ldots, N_d\}$ are represented by the matrix $[x_d] = [\mathbf{x}_d^1 \ldots \mathbf{x}_d^{N_d}]$ in $\mathbb{M}_{n,N_d}$. Let $[\mathbf{X}] = [\mathbf{X}^1, \ldots, \mathbf{X}^{N_d}]$ be the random matrix with values in $\mathbb{M}_{n,N_d}$, whose columns are N_d independent copies of random vector $\mathbf{X}$. Using the PCA of $\mathbf{X}$, random matrix $[\mathbf{X}]$ is written as,

$$[\mathbf{X}] = [\underline{x}] + [\varphi]\,[\mu]^{1/2}\,[\mathbf{H}]\,, \tag{A.1}$$

in which $[\mathbf{H}] = [\mathbf{H}^1, \ldots, \mathbf{H}^{N_d}]$ is a $\mathbb{M}_{\nu,N_d}$-valued random matrix, where $\nu \leq n$, and where $[\mu]$ is the $(\nu \times \nu)$ diagonal matrix of the ν positive eigenvalues of the empirical estimate of the covariance matrix of $\mathbf{X}$. The $(n \times \nu)$ matrix $[\varphi]$ is made up of the associated eigenvectors such $[\varphi]^T\,[\varphi] = [I_\nu]$. The matrix $[\underline{x}]$ in $\mathbb{M}_{n,N_d}$ has identical columns, each one being equal to the empirical estimate $\underline{\mathbf{x}} \in \mathbb{R}^n$ of the mean value of random vector $\mathbf{X}$. The columns of $[\mathbf{H}]$ are N_d independent copies of a random vector $\mathbf{H}$ with values in $\mathbb{R}^\nu$. The realization $[\eta_d] = [\boldsymbol{\eta}_d^1 \ldots \boldsymbol{\eta}_d^{N_d}] \in \mathbb{M}_{\nu,N_d}$ of $[\mathbf{H}]$ is computed by $[\eta_d] = [\mu]^{-1/2}[\varphi]^T\,([x_d] - [\underline{x}])$. The value ν is classically calculated in order that the L^2- error function $\nu \mapsto \text{err}_{\mathbf{X}}(\nu)$ defined by

$$\text{err}_{\mathbf{X}}(\nu) = 1 - \frac{\sum_{\alpha=1}^{\nu} \mu_\alpha}{E\{\|\mathbf{X}\|^2\}}\,, \tag{A.2}$$

be smaller than ε_{PCA}. If $\nu < n$, then there is a statistical reduction.

(A.2) Construction of a reduced-order diffusion-maps basis. For preserving the concentration of the learned realizations in the region in which the points of the training dataset are concentrated at, the PLoM relies on the diffusion-maps method [103, 104]. This is an algebraic basis of vector space $\mathbb{R}^{N_d}$, which is constructed using the diffusion maps. Let $[K]$ and $[b]$ be the matrices such that, for all i and j in $\{1,\ldots,N_d\}$, $[K]_{ij} = \exp\{-(4\,\varepsilon_{\text{DM}})^{-1}\|\boldsymbol{\eta}_d^i - \boldsymbol{\eta}_d^j\|^2\}$ and $[b]_{ij} = \delta_{ij}\,b_i$ with $b_i = \sum_{j=1}^{N_d}[K]_{ij}$, in which $\varepsilon_{\text{DM}} > 0$ is a smoothing parameter. The eigenvalues $\lambda_1,\ldots,\lambda_{N_d}$ and the associated eigenvectors $\boldsymbol{\psi}^1,\ldots,\boldsymbol{\psi}^{N_d}$ of the right-eigenvalue problem $[\mathbb{P}]\,\boldsymbol{\psi}^\alpha = \lambda_\alpha\,\boldsymbol{\psi}^\alpha$ are such that $1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_{N_d}$ and are computed by solving the generalized eigenvalue problem $[K]\,\boldsymbol{\psi}^\alpha = \lambda_\alpha\,[b]\,\boldsymbol{\psi}^\alpha$ with the normalization $<[b]\,\boldsymbol{\psi}^\alpha, \boldsymbol{\psi}^\beta> = \delta_{\alpha\beta}$. The eigenvector $\boldsymbol{\psi}^1$ associated with $\lambda_1 = 1$ is a constant vector. For a given integer $\kappa \geq 0$, the diffusion-maps basis $\{\mathbf{g}^1,\ldots,\mathbf{g}^\alpha,\ldots,\mathbf{g}^{N_d}\}$ is a vector basis of $\mathbb{R}^{N_d}$ defined by $\mathbf{g}^\alpha = \lambda_\alpha^\kappa\,\boldsymbol{\psi}^\alpha$. For a given integer m, the reduced-order diffusion-maps basis of order m is defined as the family $\{\mathbf{g}^1,\ldots,\mathbf{g}^m\}$ that is represented by the matrix $[g_m] = [\mathbf{g}^1\ldots\mathbf{g}^m] \in \mathbb{M}_{N_d,m}$ with $\mathbf{g}^\alpha = (g_1^\alpha,\ldots,g_{N_d}^\alpha)$ and $[g_m]_{\ell\alpha} = g_\ell^\alpha$. This basis depends on two parameters, ε_{DM} and m, which have to be identified. It is proven in [22], that the PLoM method does not depend on κ that can therefore be chosen to 0.

We have to find the optimal value $m_{\text{opt}} \leq N_d$ of m and the smallest value $\varepsilon_{\text{opt}} > 0$ of ε_{DM} such that (see [23])

$$1 = \lambda_1 > \lambda_2(\varepsilon_{\text{opt}}) \simeq \ldots \simeq \lambda_{m_{\text{opt}}}(\varepsilon_{\text{opt}}) \gg \lambda_{m_{\text{opt}}+1}(\varepsilon_{\text{opt}}) \geq \ldots \geq \lambda_{N_d}(\varepsilon_{\text{opt}}) > 0\,, \tag{A.3}$$

with an amplitude jump equal to an order of magnitude (a factor 10 as demonstrated in [22]) between $\lambda_{m_{\text{opt}}}(\varepsilon_{\text{opt}})$ and $\lambda_{m_{\text{opt}}+1}(\varepsilon_{\text{opt}})$. A further in-depth analysis makes it possible to state the following algorithm to estimate ε_{opt} and m_{opt}. Let $\varepsilon_{\text{DM}} \mapsto \text{Jump}(\varepsilon_{\text{DM}})$ be the function on $]0,+\infty[$ defined by

$$\text{Jump}(\varepsilon_{\text{DM}}) = \lambda_{m_{\text{opt}}+1}(\varepsilon_{\text{DM}})/\lambda_2(\varepsilon_{\text{DM}})\,. \tag{A.4}$$

The algorithm is the following:

- set the value of m to $m_{\text{opt}} = \nu + 1$;

- identify the smallest possible value ε_{opt} of ε_{DM} in order that $\text{Jump}(\varepsilon_{\text{opt}}) \leq 0.1$ and such that Equation (A.3) be verified.

(A.3) Reduced-order representation of random matrices $[\boldsymbol{H}]$ and $[\boldsymbol{X}]$. The diffusion-maps vectors $\mathbf{g}^1,\ldots,\mathbf{g}^m \in \mathbb{R}^{N_d}$ span a subspace of $\mathbb{R}^{N_d}$ that characterizes, for the optimal values m_{opt} and ε_{opt} of m and ε_{DM}, the local geometry structure of dataset $\{\boldsymbol{\eta}_d^j, j = 1,\ldots,N_d\}$. So the PLoM method introduces the $\mathbb{M}_{\nu,N_d}$-valued random matrix $[\mathbf{H}_m] = [\mathbf{Z}_m]\,[g_m]^T$ with $m \leq N_d$, corresponding to a data-reduction representation of random matrix $[\mathbf{H}]$, in which $[\mathbf{Z}_m]$ is a $\mathbb{M}_{\nu,m}$-valued random matrix. The MCMC generator of random matrix $[\mathbf{Z}_m]$ belongs to the class of Hamiltonian Monte Carlo methods, is explicitly described in [21], and is mathematically detailed in Theorem 6.3 of [22]. For generating the learned dataset, the best probability measure of $[\mathbf{H}_m]$ is obtained for $m = m_{\text{opt}}$ and using the previously defined $[g_{m_{\text{opt}}}]$. For these optimal quantities m_{opt} and $[g_{m_{\text{opt}}}]$, the generator allows for computing n_{MC} realizations $\{[\mathbf{z}_{\text{ar}}^\ell], \ell = 1,\ldots,n_{\text{MC}}\}$ of $[\mathbf{Z}_{m_{\text{opt}}}]$ and therefore, for deducing the n_{MC} realizations $\{[\boldsymbol{\eta}_{\text{ar}}^\ell], \ell = 1,\ldots,n_{\text{MC}}\}$

of $[\mathbf{H}_{m_{\text{opt}}}]$. The reshaping of matrix $[\boldsymbol{\eta}_{\text{ar}}^{\ell}] \in \mathbb{M}_{\nu,N_d}$ allows for obtaining $N_{\text{ar}} = n_{\text{MC}} \times N_d$ learned realizations $\{\boldsymbol{\eta}_{\text{ar}}^{\ell'}, \ell' = 1, \ldots, N_{\text{ar}}\}$ of $\mathbf{H}$. These learned realizations allow for estimating converged statistics on $\mathbf{H}$ and then on $\mathbf{X}$, such as pdf, moments, or conditional expectation of the type $E\{\boldsymbol{\xi}(\mathbf{Q}) \,|\, \mathbf{W} = \mathbf{w}\}$ for $\mathbf{w}$ given in $\mathbb{R}^{n_w}$ and for any given vector-valued function $\boldsymbol{\xi}$ defined on $\mathbb{R}^{n_q}$.

(A.4) Criterion for quantifying the concentration of the probability measure of random matrix $[\boldsymbol{H}_{m_{\text{opt}}}]$. For $m \leq N_d$, the concentration of the probability measure of random matrix $[\mathbf{H}_m]$ is defined (see [22]) by

$$d_{N_d}^2(m) = E\{\|[\mathbf{H}_m] - [\eta_d]\|^2\}/\|[\eta_d]\|^2 \,. \tag{A.5}$$

Let $\mathcal{M}_{\text{opt}} = \{m_{\text{opt}}, m_{\text{opt}} + 1, \ldots, N_d\}$ in which m_{opt} is the optimal value of m previously defined. Theorem 7.8 of [22] shows that $\min_{m \in \mathcal{M}_{\text{opt}}} d_{N_d}^2(m) \leq 1 + m_{\text{opt}}/(N_d - 1) < d_{N_d}^2(N_d)$, which means that the PLoM method, for $m = m_{\text{opt}}$ and $[g_{m_{\text{opt}}}]$ is a better method than the usual one corresponding to $d_{N_d}^2(N_d) = 1 + N_d/(N_d - 1) \simeq 2$. Using the n_{MC} realizations $\{[\boldsymbol{\eta}_{\text{ar}}^{\ell}], \ell = 1, \ldots, n_{\text{MC}}\}$ of $[\mathbf{H}_{m_{\text{opt}}}]$, we have the estimate $d_{N_d}^2(m_{\text{opt}}) \simeq (1/n_{\text{MC}}) \sum_{\ell=1}^{n_{\text{MC}}} \{\|[\boldsymbol{\eta}_{\text{ar}}^{\ell}] - [\eta_d]\|^2\}/\|[\eta_d]\|^2$.

(A.5) Generation of learned realizations $\{\boldsymbol{\eta}_{\text{ar}}^{\ell'}, \ell' = 1, \ldots, N_{\text{ar}}\}$ *of random vector* $\boldsymbol{H}$. The MCMC generator is detailed in [21]. Let $\{([\boldsymbol{\mathcal{Z}}(t)], [\boldsymbol{\mathcal{Y}}(t)]), t \in \mathbb{R}^+\}$ be the unique asymptotic (for $t \to +\infty$) stationary diffusion stochastic process with values in $\mathbb{M}_{\nu,m_{\text{opt}}} \times \mathbb{M}_{\nu,m_{\text{opt}}}$, of the following reduced-order ISDE (stochastic nonlinear second-order dissipative Hamiltonian dynamic system), for $t > 0$,

$$\begin{aligned} d[\boldsymbol{\mathcal{Z}}(t)] &= [\boldsymbol{\mathcal{Y}}(t)]\, dt \,, \\ d[\boldsymbol{\mathcal{Y}}(t)] &= [\mathcal{L}([\boldsymbol{\mathcal{Z}}(t)])]\, dt - \frac{1}{2} f_0\, [\boldsymbol{\mathcal{Y}}(t)]\, dt \\ &\quad + \sqrt{f_0}\, [d\boldsymbol{\mathcal{W}}^{\text{wien}}(t)] \,, \end{aligned}$$

with $[\boldsymbol{\mathcal{Z}}(0)] = [\eta_d]\,[a]$ and $[\boldsymbol{\mathcal{Y}}(0)] = [\boldsymbol{\mathcal{N}}]\,[a]$, in which

$$[a] = [g_{m_{\text{opt}}}]\,([g_{m_{\text{opt}}}]^T\,[g_{m_{\text{opt}}}])^{-1} \in \mathbb{M}_{N_d,m_{\text{opt}}} \,.$$

(1) $[\mathcal{L}([\boldsymbol{\mathcal{Z}}(t)])] = [L([\boldsymbol{\mathcal{Z}}(t)]\,[g_{m_{\text{opt}}}]^T)]\,[a]$ is a random matrix with values in $\mathbb{M}_{\nu,m_{\text{opt}}}$. For all $[u] = [\mathbf{u}^1 \ldots \mathbf{u}^{N_d}]$ in $\mathbb{M}_{\nu,N_d}$ with $\mathbf{u}^j = (u_1^j, \ldots, u_\nu^j)$ in $\mathbb{R}^\nu$, the matrix $[L([u])]$ in $\mathbb{M}_{\nu,N_d}$ is defined, for all $k = 1, \ldots, \nu$ and for all $j = 1, \ldots, N_d$, by

$$[L([u])]_{kj} = \frac{1}{p(\mathbf{u}^j)}\, \{\boldsymbol{\nabla}_{\mathbf{u}^j}\, p(\mathbf{u}^j)\}_k \,, \tag{A.6}$$

$$p(\mathbf{u}^j) = \frac{1}{N_d} \sum_{j'=1}^{N_d} \exp\{-\frac{1}{2\widehat{s}_\nu^2} \|\frac{\widehat{s}_\nu}{s_\nu} \boldsymbol{\eta}^{j'} - \mathbf{u}^j\|^2\} \,,$$

$$\boldsymbol{\nabla}_{\mathbf{u}^j}\, p(\mathbf{u}^j) = \frac{1}{\widehat{s}_\nu^2\, N_d} \sum_{j'=1}^{N_d} (\frac{\widehat{s}_\nu}{s_\nu} \boldsymbol{\eta}^{j'} - \mathbf{u}^j)\, \exp\{-\frac{1}{2\widehat{s}_\nu^2} \|\frac{\widehat{s}_\nu}{s_\nu} \boldsymbol{\eta}^{j'} - \mathbf{u}^j\|^2\} \,,$$

in which $\widehat{s}_\nu$ is the modified Silverman bandwidth s_ν, which has been introduced in [105],

$$\widehat{s}_\nu = \frac{s_\nu}{\sqrt{s_\nu^2 + \frac{N_d-1}{N_d}}} \quad , \quad s_\nu = \left\{ \frac{4}{N_d(2+\nu)} \right\}^{1/(\nu+4)} .$$

(2) $[\boldsymbol{\mathcal{W}}^{\text{wien}}(t)] = [\mathbb{W}^{\text{wien}}(t)]\,[a]$ where $\{[\mathbb{W}^{\text{wien}}(t)],\, t \in \mathbb{R}^+\}$ is the $\mathbb{M}_{\nu,N_d}$-valued normalized Wiener process.

(3) $[\boldsymbol{\mathcal{N}}]$ is the $\mathbb{M}_{\nu,N_d}$-valued normalized Gaussian random matrix that is independent of process $[\mathbb{W}^{\text{wien}}]$.

(4) The free parameter f_0, such that $0 < f_0 < 4/\widehat{s}_\nu$, allows the dissipation term of the nonlinear second-order dynamic system (dissipative Hamiltonian system) to be controlled in order to kill the transient part induced by the initial conditions. A common value is $f_0 = 4$ (note that $\widehat{s}_\nu < 1$).

(5) We then have $[\mathbf{Z}_{m_{\text{opt}}}] = \lim_{t\to+\infty} [\boldsymbol{\mathcal{Z}}(t)]$ in probability distribution. The Störmer-Verlet scheme is used for solving the reduced-order ISDE, which allows for generating the learned realizations, $[z^1_{\text{ar}}], \ldots, [z^{n_{\text{MC}}}_{\text{ar}}]$, and then, generating the learned realizations $[\eta^1_{\text{ar}}], \ldots, [\eta^{n_{\text{MC}}}_{\text{ar}}]$ such that $[\eta^\ell_{\text{ar}}] = [z^\ell_{\text{ar}}]\,[g_{m_{\text{opt}}}]^T$.

(6) The learned realizations $\{\mathbf{x}^{\ell'}_{\text{ar}}, \ell' = 1, \ldots, N_{\text{ar}}\}$ of random vector $\mathbf{X}$ are then calculated (see Eq. (A.1)) by $\mathbf{x}^{\ell'}_{\text{ar}} = \underline{\mathbf{x}} + [\varphi]\,[\mu]^{1/2}\,\boldsymbol{\eta}^{\ell'}_{\text{ar}}$.

(A.6) Constraints on the second-order moments of the components of $\boldsymbol{H}$. In general, the mean value of $\mathbf{H}$ estimated using the N_{ar} learned realizations $\{\boldsymbol{\eta}^{\ell'}_{\text{ar}}, \ell' = 1, \ldots, N_{\text{ar}}\}$, is sufficiently close to zero. Likewise, the estimate of the covariance matrix of $\mathbf{H}$, which must be the identity matrix, is sufficiently close to a diagonal matrix. However, sometimes the diagonal entries of the estimated covariance matrix can be lower than 1. Normalization can be recovered by imposing constraints

$$\{E\{(H_k)^2\} = 1, k = 1, \ldots, \nu\}\,,$$

in the algorithm presented in paragraph (v). For that, we use the method and the iterative algorithm presented in [23] (that is based on Sections 5.5 and 5.6 of [44]). The constraints are imposed by using the Kullback-Leibler minimum cross-entropy principle. The resulting optimization problem is formulated using a Lagrange multiplier $\mathbf{v} = (v_1, \ldots, v_\nu)$ associated with the constraints. The optimal solution of the Lagrange multiplier is computed using an efficient iterative algorithm. At each iteration, the MCMC generator detailed in paragraph *A.5* is used. The constraints are rewritten as

$$E\{\mathbf{h}(\mathbf{H})\} = \mathbf{b}\,,$$

in which the function $\mathbf{h} = (h_1, \ldots, h_\nu)$ and the vector $\mathbf{b} = (b_1, \ldots, b_\nu)$ are such that $h_k(\mathbf{H}) = (H_k)^2$ and $b_k = 1$ for k in $\{1, \ldots, \nu\}$. To take into account the constraints in the algorithm of paragraph *A.5*, Eq. (A.6) is replaced by the following one,

$$[L_{\boldsymbol{\lambda}}([u])]_{kj} = \frac{1}{p(\mathbf{u}^j)}\,\{\boldsymbol{\nabla}_{\mathbf{u}^j}\, p(\mathbf{u}^j)\}_k - 2\,\lambda_k u^j_k\,.$$

The iteration algorithm for computing $\boldsymbol{\lambda}^{i+1}$ as a function of $\boldsymbol{\lambda}^i$ is the following,

$$\begin{aligned} \boldsymbol{\lambda}^{i+1} &= \boldsymbol{\lambda}^i - \alpha_i [\Gamma''(\boldsymbol{\lambda}^i)]^{-1} \, \boldsymbol{\Gamma}'(\boldsymbol{\lambda}^i) \quad , \quad i \geq 0 \, , \\ \boldsymbol{\lambda}^0 &= \mathbf{0}_\nu \, , \end{aligned} \tag{A.7}$$

in which $\boldsymbol{\Gamma}'(\boldsymbol{\lambda}^i) = \mathbf{b} - E\{\mathbf{h}(\mathbf{H}_{\boldsymbol{\lambda}^i})\}$ and $[\Gamma''(\boldsymbol{\lambda}^i)] = [\text{cov}\{\mathbf{h}(\mathbf{H}_{\boldsymbol{\lambda}^i})\}]$ (the covariance matrix), and where α_i is a relaxation function (less than 1) that is introduced for controlling the convergence as a function of iteration number i. For given $i_2 \geq 2$, for given β_1 and β_2 such that $0 < \beta_1 < \beta_2 \leq 1$, α_i can be defined by:

- for $i \leq i_2$, $\alpha_i = \beta_1 + (\beta_2 - \beta_1)(i-1)/(i_2 - 1)$;

- for $i > i_2$, $\alpha_i = \beta_2$.

The convergence of the iteration algorithm is controlled by the error function $i \mapsto \text{err}(i)$ defined by

$$\text{err}(i) = \|\mathbf{b} - E\{\mathbf{h}(\mathbf{H}_{\boldsymbol{\lambda}^i})\}\| / \|\mathbf{b}\| \, . \tag{A.8}$$

At each iteration i, $E\{\mathbf{h}(\mathbf{H}_{\boldsymbol{\lambda}^i})\}$ and $[\text{cov}\{\mathbf{h}(\mathbf{H}_{\boldsymbol{\lambda}^i})\}]$ are estimated by using the N_{ar} learned realizations of $\mathbf{H}_{m_{\text{opt}}}(\boldsymbol{\lambda}^i)$ obtained by reshaping the learned realizations.

Appendix B

Algebraic expressions of the conditional statistics

The following algebraic expressions of the conditional statistics are those presented in [95, 96]. The conditional expectation is defined by

$$E\{\mathbf{Q}|\mathbf{W}=\mathbf{w}_o\}=\int_{\mathbb{R}_{n_q}}\mathbf{q}\,p_{\mathbf{Q}|\mathbf{W}}(\mathbf{q}|\mathbf{w}_o)\,d\mathbf{q}\,, \tag{B.1}$$

while the conditional cumulative distribution function is defined by,

$$\mathrm{Proba}\{\mathrm{Q_i}\leq \mathrm{q_i^*}\,|\,\mathbf{W}=\mathbf{w_o}\}=\int_{-\infty}^{\mathrm{q_i^*}}\mathrm{p_{Q_i|W}}(\mathrm{q_i^*}\,|\,\mathbf{w_o})\,\mathrm{dq_i}\,, \tag{B.2}$$

which allows to estimate the confidence region.

Let $\widetilde{\mathbf{Q}}=(\widetilde{Q}_1,\ldots,\widetilde{Q}_{n_q})$ and $\widetilde{\mathbf{W}}=(\widetilde{Q}_1,\ldots,\widetilde{W}_{n_w})$ be the normalized random vector whose components are defined by

$$\widetilde{Q}_i=(Q_i-\underline{q}_i)/\sigma_{Q_i}\quad,\quad \widetilde{W}_k=(W_k-\underline{w}_k)/\sigma_{W_k}\,, \tag{B.3}$$

in which $\underline{q}_i$, $\underline{w}_k$, and σ_{Q_i}, σ_{W_k} are the mean values and the standard deviations of the random variables Q_i and W_k, which are estimated with the empirical statistical estimators using the learned realizations $\{(\mathbf{q}_{\mathrm{ar}}^\ell,\mathbf{w}_{\mathrm{ar}}^\ell),\ell=1,\ldots,N_{\mathrm{ar}}\}$. The Gaussian KDE estimation of the joint probability distribution of $\widetilde{\mathbf{Q}}$ and $\widetilde{\mathbf{W}}$ is written as,

$$p_{\widetilde{\mathbf{Q}},\widetilde{\mathbf{W}}}(\widetilde{\mathbf{q}},\widetilde{\mathbf{w}})=\frac{1}{N_{\mathrm{ar}}}\sum_{\ell=1}^{N_{\mathrm{ar}}}\frac{1}{(\sqrt{2\pi}s)^{n_q}}\exp(-\frac{1}{2s^2}\|\widetilde{\mathbf{q}}-\widetilde{\mathbf{q}}_{\mathrm{ar}}^\ell\|^2)\frac{1}{(\sqrt{2\pi}s)^{n_w}}\exp(-\frac{1}{2s^2}\|\widetilde{\mathbf{w}}-\widetilde{\mathbf{w}}_{\mathrm{ar}}^\ell\|^2)\,, \tag{B.4}$$

in which s is the Silverman bandwidth given by

$$s=\left\{\frac{4}{N_{\mathrm{ar}}(2+n)}\right\}^{1/(n+4)}\quad,\quad n=n_q+n_w\,. \tag{B.5}$$

The derived algebraic expression of conditional mathematical expectation of component Q_i given $\mathbf{W} = \mathbf{w}_o$ is written as

$$E\{Q_i|\mathbf{W} = \mathbf{w}_o\} = \underline{q}_i + \sigma_{Q_i}\, E\{\widetilde{Q}_i|\widetilde{\mathbf{W}} = \widetilde{\mathbf{w}}_o\} \quad , \quad \widetilde{w}_{o,k} = (w_{o,k} - \underline{w}_k)/\sigma_{W_k}\,, \tag{B.6}$$

$$E\{\widetilde{Q}_i|\widetilde{\mathbf{W}} = \widetilde{\mathbf{w}}_o\} = \frac{\sum_{\ell=1}^{N_{\text{ar}}} \widetilde{q}^{\ell}_{\text{ar},i} \times \exp(-\frac{1}{2s^2}\|\widetilde{\mathbf{w}}_o - \widetilde{\mathbf{w}}^{\ell}_{\text{ar}}\|^2)}{\sum_{\ell=1}^{N_{\text{ar}}} \exp(-\frac{1}{2s^2}\|\widetilde{\mathbf{w}}_o - \widetilde{\mathbf{w}}^{\ell}_{\text{ar}}\|^2)}\,. \tag{B.7}$$

The conditional cdf $F_{Q_i|\mathbf{W}}(q_i^*|\mathbf{w}_o) = \text{Proba}\{\text{Q}_\text{i} \leq \text{q}_\text{i}^* \,|\, \mathbf{W} = \mathbf{w}_\text{o}\}$ is written as,

$$F_{Q_i|\mathbf{W}}(q_i^*|\mathbf{w}_o) = \frac{\sum_{\ell=1}^{N_{\text{ar}}} \widetilde{F}_{Q_i^{\ell}}(\widetilde{q}_i^*) \times \exp(-\frac{1}{2s^2}\|\widetilde{\mathbf{w}}_o - \widetilde{\mathbf{w}}^{\ell}\|^2)}{\sum_{\ell=1}^{N_{\text{ar}}} \exp(-\frac{1}{2s^2}\|\widetilde{\mathbf{w}}_o - \widetilde{\mathbf{w}}^{\ell}\|^2)} \quad , \quad \widetilde{q}_i^* = (q_i^* - \underline{q}_i)/\sigma_{Q_i}\,, \tag{B.8}$$

$$F_{Q_i|\mathbf{W}}(q_i^*|\mathbf{w}_o) = \frac{1}{2} + \frac{1}{2}\text{erf}(\frac{1}{\sqrt{2}\,s}(\widetilde{q}_i^* - \widetilde{q}^{\ell}_{\text{ar},i})) \quad , \quad \text{erf}(y) = \frac{2}{\sqrt{\pi}}\int_0^y e^{-t^2} dt\,. \tag{B.9}$$

References

[1] Martha Brown, Michael Jones, and Willie Watson. "Uncertainty analysis of the grazing flow impedance tube". In: *18th AIAA/CEAS Aeroacoustics Conference (33rd AIAA Aeroacoustics Conference)*. 2012, p. 2296. DOI: 10.2514/6.2012-2296.

[2] Michael Jones, Tony Parrott, and Willie Watson. "Uncertainty and sensitivity analyses of a two-parameter impedance prediction model". In: *14th AIAA/CEAS Aeroacoustics Conference (29th AIAA Aeroacoustics Conference)*. 2008, p. 2928. DOI: 10.2514/6.2008-2928.

[3] Douglas M Nark, Michael G Jones, and Estelle Piot. "Assessment of axial wave number and mean flow uncertainty on acoustic liner impedance eduction". In: *2018 AIAA/CEAS Aeroacoustics Conference*. 2018, p. 3444. DOI: 10.2514/6.2018-3444.

[4] André Mateus Netto Spillere, Lucas Araujo Bonomo, Julio Apolinario Cordioli, and Edward James Brambley. "Experimentally testing impedance boundary conditions for acoustic liners with flow: Beyond upstream and downstream". In: *Journal of Sound and Vibration* 489 (2020), p. 115676. DOI: 10.1016/j.jsv.2020.115676.

[5] Roger Ohayon and Christian Soize. *Structural acoustics and vibration: Mechanical models, variational formulations and discretization*. Academic Press, 1998.

[6] Loris Casadei, Hugues Deniau, Estelle Piot, and Thomas Node-Langlois. "Time-domain impedance boundary condition implementation in a CFD solver and validation against experimental data of acoustical liners". In: *eForum Acusticum* (2020), pp. 359–366. DOI: 10.48465/fa.2020.0088.

[7] Vincent Dangla, Christian Soize, Guilherme Cunha, Aurélien Mosson, Morad Kassem, and Benoit Van Den Nieuwenhof. "Stochastic computational model of 3D acoustic noise predictions for nacelle liners". In: *AIAA Aviation 2020 Forum*. 2020, p. 2545. DOI: 10.2514/6.2020-2545.

[8] Maud Lavieille, Touffic Abboud, Ahmed Bennani, and Nolwenn Balin. "Numerical simulations of perforate liners: Part I - Model description and impedance validation". In: *19th AIAA/CEAS Aeroacoustics Conference*. 2013. DOI: 10.2514/6.2013-2269.

[9] Lucas Pascal, Estelle Piot, and Grégoire Casalis. "A new implementation of the extended Helmholtz resonator acoustic liner impedance model in time domain CAA". In: *Journal of Computational Acoustics* 24.01 (2016), pp. 1663–1674. DOI: 10.1142/S0218396X15500150.

[10] Bernard Van Antwerpen, Yves Detandt, Diego Copiello, Eveline Rosseel, and Eloi Gaudry. "Performance improvements and new solution strategies of Actran/TM for nacelle simulations". In: *20th AIAA/CEAS Aeroacoustics Conference*. 2014, p. 2315. DOI: 10.2514/6.2014-2315.

[11] Julian Winkler, Jeffrey M Mendoza, C Aaron Reimann, Kenji Homma, and Jose S Alonso. "High fidelity modeling tools for engine liner design and screening of advanced concepts". In: *International Journal of Aeroacoustics* 20.5-7 (2021), pp. 530–560. DOI: 10.1177/1475472X211023884.

[12] Andrew T Chambers, James M Manimala, and Michael G Jones. "Design and optimization of 3D folded-core acoustic liners for enhanced low-frequency performance". In: *AIAA Journal* 58.1 (2020), pp. 206–218. DOI: 10.2514/1.J058017.

[13] Vincent Dangla, Christian Soize, Guilherme Cunha, Aurélien Mosson, Morad Kassem, and Benoit Van den Nieuwenhof. "Robust three-dimensional acoustic performance probabilistic model for nacelle liners". In: *AIAA Journal* 59.10 (2021), pp. 4195–4211. DOI: 10.2514/1.J060299.

[14] Douglas M Nark and Michael G Jones. "Design of an advanced inlet liner for the quiet technology demonstrator 3". In: *25th AIAA/CEAS Aeroacoustics Conference*. 2019, p. 2764. DOI: 10.2514/6.2019-2764.

[15] Emre Özkaya, Nicolas R Gauger, Junis Abdel Hay, and Frank Thiele. "Efficient Design Optimization of Acoustic Liners for Engine Noise Reduction". In: *AIAA Journal* 58.3 (2020), pp. 1140–1156. DOI: 10.2514/1.J057776.

[16] André MN Spillere, Danilo S Braga, Leonardo A Seki, Lucas A Bonomo, Julio A Cordioli, Bernardo M Rocamora Jr, Paulo C Greco Jr, Danillo C dos Reis, and Eduardo LC Coelho. "Design of a single degree of freedom acoustic liner for a fan noise test rig". In: *International Journal of Aeroacoustics* 20.5-7 (2021), pp. 708–736. DOI: 10.1177/1475472X211023831.

[17] Daniel L Sutliff, Douglas M Nark, Michael G Jones, and Noah H Schiller. "Design and acoustic efficacy of a broadband liner for the inlet of the DGEN aero-propulsion research turbofan". In: *25th AIAA/CEAS Aeroacoustics Conference*. 2019, p. 2582. DOI: 10.2514/6.2019-2582.

[18] Benoit van Den Nieuwenhof, Yves Detandt, Gregory Lielens, Eveline Rosseel, Christian Soize, Vincent Dangla, Morad Kassem, and Aurélien Mosson. "Optimal design of the acoustic treatments damping the noise radiated by a turbo-fan engine". In: *23rd AIAA/CEAS Aeroacoustics Conference*. 2017, p. 4035. DOI: 10.2514/6.2017-4035.

[19] Y Buot de l'Épine, J-D Chazot, and J-M Ville. "Bayesian identification of acoustic impedance in treated ducts". In: *The Journal of the Acoustical Society of America* 138.1 (2015), EL114–EL119. DOI: 10.1121/1.4923013.

[20] Rémi Roncen, Fabien Mery, Estelle Piot, and Frank Simon. "Statistical inference method for liner impedance eduction with a shear grazing flow". In: *AIAA Journal* 57.3 (2019), pp. 1055–1065. DOI: 10.2514/1.J057559.

[21] Christian Soize and Roger Ghanem. "Data-driven probability concentration and sampling on manifold". In: *Journal of Computational Physics* 321 (2016), pp. 242–258. DOI: 10.1016/j.jcp.2016.05.044.

[22] Christian Soize and Roger Ghanem. “Probabilistic learning on manifolds”. In: *Foundations of Data Science* 2.3 (2020), pp. 279–307. DOI: 10.3934/fods.2020013.

[23] Christian Soize and Roger Ghanem. “Probabilistic learning on manifolds (PLoM) with partition”. In: *International Journal for Numerical Methods in Engineering* 123.1 (2022), pp. 268–290. DOI: 10.1002/nme.6856.

[24] Christian Soize. “Software_PLoM_with_partition_2021_06_24”. In: (June 2021). URL: https://hal-upec-upem.archives-ouvertes.fr/hal-03275052.

[25] Roger Ghanem, Christian Soize, and Charanraj Thimmisetty. “Optimal well-placement using probabilistic learning”. In: *Data-Enabled Discovery and Applications* 2.1 (2018), pp. 4, 1–16. DOI: 10.1007/s41688-017-0014-x.

[26] Christian Soize. “Design optimization under uncertainties of a mesoscale implant in biological tissues using a probabilistic learning algorithm”. In: *Computational Mechanics* 62.3 (2018), pp. 477–497. DOI: 10.1007/s00466-017-1509-x.

[27] Kevin B Korb and Ann E Nicholson. *Bayesian artificial intelligence*. Boca Raton: CRC press, 2010.

[28] Zoubin Ghahramani. “Probabilistic machine learning and artificial intelligence”. In: *Nature* 521.7553 (2015), pp. 452–459. DOI: 10.1038/nature14541.

[29] Stuart Russel and Peter Norvig. *Artifical Intelligence, A Modern Approach*. Harlow: Third Edition, Pearson, 2016.

[30] Thomas Daniel, Fabien Casenave, Nissrine Akkari, and David Ryckelynck. “Model order reduction assisted by deep neural networks (ROM-net)”. In: *Advanced Modeling and Simulation in Engineering Sciences* 7 (2020), pp. 1–27. DOI: 10.1186/s40323-020-00153-6.

[31] V.N. Vapnik. *The Nature of Statistical Learning Theory*. New York: Springer, 2000. DOI: 10.1007/978-1-4757-3264-1.

[32] Jonathan Taylor and Robert J Tibshirani. “Statistical learning and selective inference”. In: *Proceedings of the National Academy of Sciences* 112.25 (2015), pp. 7629–7634. DOI: 10.1073/pnas.1507583112.

[33] Renee Swischuk, Laura Mainini, Benjamin Peherstorfer, and Karen Willcox. “Projection-based model reduction: Formulations for physics-based machine learning”. In: *Computers & Fluids* 179 (2019), pp. 704–717. DOI: 10.1016/j.compfluid.2018.07.021.

[34] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*. Vol. 112. Springer, 2013.

[35] George El Haber, Jonathan Viquerat, Aurelien Larcher, David Ryckelynck, Jose Alves, Aakash Patil, and Elie Hachem. “Deep learning model to assist multiphysics conjugate problems”. In: *Physics of Fluids* 34.1 (2022). DOI: 10.1063/5.0077723.

[36] Yannis Kevrekidis. “Manifold Learning for Parameter Reduction”. In: *Bulletin of the American Physical Society* 65 (2020). DOI: 10.1016/j.jcp.2019.04.015.

[37] Guillaume Perrin, Christian Soize, and N Ouhbi. “Data-driven kernel representations for sampling with an unknown block dependence structure under correlation constraints”. In: *Computational Statistics & Data Analysis* 119 (2018), pp. 139–154. DOI: 10.1016/j.csda.2017.10.005.

[38] Christian Soize and Roger Ghanem. “Polynomial chaos representation of databases on manifolds”. In: *Journal of Computational Physics* 335 (2017), pp. 201–221. DOI: 10.1016/j.jcp.2017.01.031.

[39] Christian Soize, Roger Ghanem, Cosmin Safta, Xun Huan, Zachary P. Vane, Joseph C. Oefelein, Guilhem Lacaze, Habib N. Najm, Q. Tang, and X. Chen. “Entropy-based closure for probabilistic learning on manifolds”. In: *Journal of Computational Physics* 388 (2019), pp. 528–533. DOI: 10.1016/j.jcp.2018.12.029.

[40] Guillaume Perrin, Christian Soize, Sophie Marque-Pucheu, and Josselin Garnier. “Nested polynomial trends for the improvement of Gaussian process-based predictors”. In: *Journal of Computational Physics* 346 (2017), pp. 389–402. DOI: 10.1016/j.jcp.2017.05.051.

[41] Katiana Kontolati, Dimitrios Loukrezis, Ketson RM dos Santos, Dimitrios G Giovanis, and Michael D Shields. “Manifold learning-based polynomial chaos expansions for high-dimensional surrogate models”. In: *International Journal for Uncertainty Quantification* 12.4 (2022). DOI: 10.1615/Int.J.UncertaintyQuantification.2022039936.

[42] Thomas Daniel, Fabien Casenave, Nissrine Akkari, Ali Ketata, and David Ryckelynck. “Physics-informed cluster analysis and a priori efficiency criterion for the construction of local reduced-order bases”. In: *Journal of Computational Physics* 458 (2022), p. 111120. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2022.111120.

[43] Shaowu Pan and Karthik Duraisamy. “Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability”. In: *SIAM Journal on Applied Dynamical Systems* 19.1 (2020), pp. 480–509. DOI: 10.1137/19M1267246.

[44] Christian Soize and Roger Ghanem. “Physics-constrained non-Gaussian probabilistic learning on manifolds”. In: *International Journal for Numerical Methods in Engineering* 121.1 (2020), pp. 110–145. DOI: 10.1002/nme.6202.

[45] Christian Soize and Roger Ghanem. “Probabilistic learning on manifolds constrained by nonlinear partial differential equations for small datasets”. In: *Computer Methods in Applied Mechanics and Engineering* 380 (2021), p. 113777. DOI: 10.1016/j.cma.2021.113777.

[46] Christian Soize. “Probabilistic learning inference of boundary value problem with uncertainties based on Kullback-Leibler divergence under implicit constraints”. In: *Computer Methods in Applied Mechanics and Engineering* 395 (2022), p. 115078. DOI: 10.1016/j.cma.2022.115078.

[47] Christian Soize. “Probabilistic learning constrained by realizations using a weak formulation of Fourier transform of probability measures”. In: *Computational Statistics* (2022), 1–30, published online 23 December 2022. DOI: 10.1007/s00180-022-01300-w.

[48] Howard B. Demuth, Mark H. Beale, Orlando De Jess, and Martin T. Hagan. *Neural Network Design*. 2nd. Stillwater, OK, USA: Martin Hagan, Oklahoma State University, 2014. DOI: 10.5555/2721661.

[49] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[50] Martin T. Hagan, Howard B. Demuth, and Mark Beale. *Neural Network Design*. 20 Park Plaza Boston, MA, USA: PWS Publishing Co., 1997. DOI: 10.5555/249049.

[51] Mohamad H Hassoun. *Fundamentals of artificial neural networks*. MIT press, 1995.

[52] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. 2nd. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. DOI: 10.5555/521706.

[53] Alexandr Komkin, Aleksei Bykov, and Mikhail Mironov. "Experimental study of nonlinear acoustic impedance of circular orifices". In: *The Journal of the Acoustical Society of America* 148.3 (Sept. 2020), pp. 1391–1403. DOI: 10.1121/10.0001940.

[54] Zacharie Laly, Noureddine Atalla, and Sid-Ali Meslioui. "Acoustical modeling of micro-perforated panel at high sound pressure levels using equivalent fluid approach". In: *Journal of Sound and Vibration* 427 (2018), pp. 134–158. DOI: https://doi.org/10.1016/j.jsv.2017.09.011.

[55] Ronald L. Panton and Allen L. Goldman. "Correlation of nonlinear orifice impedance". In: *The Journal of the Acoustical Society of America* 60.6 (1976), pp. 1390–1396. DOI: 10.1121/1.381232.

[56] Amritesh Sinha, Christian Soize, Christophe Desceliers, and Guilherme Cunha. "Aeroacoustic liner impedance metamodel from simulation and experimental data using probabilistic learning". In: *AIAA Journal* 61.11 (2023), pp. 4926–4934. DOI: 10.2514/1.J062991.

[57] Christian Soize. *Uncertainty quantification*. Springer, 2017. DOI: 10.1007/978-3-319-54339-0.

[58] Suresh Palani, Paul Murray, Alan McAlpine, and Christoph Richter. "Optimisation of slanted septum core and multiple folded cavity acoustic liners for aero-engines". In: *AIAA Aviation 2021 Forum*. 2021, pp. 2021–2172. DOI: 10.2514/6.2021-2172.

[59] Suresh Palani, Paul Murray, Alan McAlpine, Daisuke Sasaki, and Christoph Richter. "Slanted septum and multiple folded cavity liners for broadband sound absorption". In: *International Journal of Aeroacoustics* 20.5-7 (2021), pp. 633–661. DOI: 10.1177/1475472X211023835.

[60] Suresh Palani, Paul Murray, Alan McAlpine, Kylie Manouk Knepper, and Christoph Richter. "Assessment of novel acoustic liners for aero-engine applications with sheared mean flow". In: *28th AIAA/CEAS Aeroacoustics 2022 Conference*. 2022, pp. 2022–2900. DOI: 10.2514/6.2022-2900.

[61] *PlotDigitizer: Version 3.1.5*. 2023. URL: https://plotdigitizer.com.

[62] Amritesh Sinha, Christophe Desceliers, Christian Soize, and Guilherme Cunha. "Statistical metamodel of liner acoustic impedance based on neural network and probabilistic learning for small datasets". In: (Submitted in April 2024).

[63] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

[64] David Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, 1997.

[65] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer, 2006. DOI: 10.1007/978-0-387-40065-5.

[66] Paul H Calamai and Jorge J Moré. "Projected gradient methods for linearly constrained problems". In: *Mathematical programming* 39.1 (1987), pp. 93–116. DOI: 10.1007/BF02592073.

[67] Diederik P. Kingma and Jimmy Ba. *Adam: A method for stochastic optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[68] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256.

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.

[70] Y Chen and D.S Oliver. "Levenberg–Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification". In: *Computational Geosciences* 17 (2013), pp. 689–703. DOI: 10.1007/s10596-013-9351-5.

[71] Mark Transtrum and James Sethna. "Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization". In: *arXiv preprint arXiv:arXiv:1201.5885* (Jan. 2012).

[72] Yuxi Hong, Houcine Bergou, Nicolas Doucet, Hao Zhang, Jesse Cranney, Hatem Ltaief, Damien Gratadour, Francois Rigaut, and David E Keyes. "Stochastic Levenberg-Marquardt for solving optimization problems on hardware accelerators". In: *35th IEEE International Parallel & Distributed Processing Symposium*. 2020.

[73] Yi Ren and Donald Goldfarb. "Efficient subsampled Gauss-Newton and natural gradient methods for training neural networks". In: *arXiv preprint arXiv:1906.02353* (2019).

[74] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[75] Christian Soize. "An overview on uncertainty quantification and probabilistic learning on manifolds in multiscale mechanics of materials". In: *Mathematics and Mechanics of Complex Systems* 11 (Oct. 2023), pp. 87–174. DOI: 10.2140/memocs.2023.11.87.

[76] Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. "Sampling via measure transport: An introduction". In: *Handbook of uncertainty quantification* (2016), pp. 1–41. DOI: 10.1007/978-3-319-11259-6{_}23-1.

[77] A Cengiz Öztireli, Marc Alexa, and Markus Gross. "Spectral sampling of manifolds". In: *ACM Transactions on Graphics (TOG)* 29.6 (2010), pp. 1–8. DOI: 10.1145/1882261.1866190.

[78] Matthew D Parno and Youssef M Marzouk. “Transport map accelerated markov chain Monte Carlo”. In: *SIAM/ASA Journal on Uncertainty Quantification* 6.2 (2018), pp. 645–682. DOI: 10.1137/17M1134640.

[79] Ameet Talwalkar, Sanjiv Kumar, and Henry Rowley. “Large-scale manifold learning”. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587670.

[80] Christian Soize, Roger Ghanem, and Christophe Desceliers. “Sampling of Bayesian posteriors with a non-Gaussian probabilistic learning on manifolds from a small dataset”. In: *Statistics and Computing* 30.5 (2020), pp. 1433–1457. DOI: 10.1007/s11222-020-09954-6.

[81] Masoumeh Dashti and Andrew M Stuart. “The Bayesian approach to inverse problems”. In: *Handbook of Uncertainty Quantification*. Ed. by Roger Ghanem, David Higdon, and Owhadi Houman. Cham, Switzerland: Springer, 2017. Chap. 10, pp. 311–428. DOI: 10.1007/978-3-319-12385-1{_}7.

[82] James E Gentle. *Computational Statistics*. New York: Springer, 2019. DOI: 10.1007/978-0-387-98144-4.

[83] Roger Ghanem, David Higdon, and Houman Owhadi. *Handbook of Uncertainty Quantification*. Vol. 1 to 3. Cham, Switzerland: Springer, 2017. DOI: 10.1007/978-3-319-12385-1.

[84] Marc C Kennedy and Anthony O’Hagan. “Bayesian calibration of computer models”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.3 (2001), pp. 425–464. DOI: 10.1111/1467-9868.00294.

[85] Youssef M Marzouk, Habib N Najm, and Larry A Rahn. “Stochastic spectral methods for efficient Bayesian solution of inverse problems”. In: *Journal of Computational Physics* 224.2 (2007), pp. 560–586. DOI: 10.1016/j.jcp.2006.10.010.

[86] Hermann G Matthies, Elmar Zander, Bojana V Rosić, Alexander Litvinenko, and Oliver Pajonk. “Inverse problems in a Bayesian setting”. In: *Computational Methods for Solids and Fluids*. Vol. 41. Springer, 2016, pp. 245–286. DOI: 10.1007/978-3-319-27996-1{_}10.

[87] Houman Owhadi, Clint Scovel, and Tim Sullivan. “On the brittleness of Bayesian inference”. In: *SIAM Review* 57.4 (2015), pp. 566–582. DOI: 10.1137/130938633.

[88] Guillaume Perrin and Christian Soize. “Adaptive method for indirect identification of the statistical properties of random fields in a Bayesian framework”. In: *Computational Statistics* 35.1 (2020), pp. 111–133. DOI: 10.1007/s00180-019-00936-5.

[89] Alessio Spantini, Tiangang Cui, Karen Willcox, Luis Tenorio, and Youssef Marzouk. “Goal-oriented optimal approximations of Bayesian linear inverse problems”. In: *SIAM Journal on Scientific Computing* 39.5 (2017), S167–S196. DOI: 10.1137/16M1082123.

[90] Andrew M Stuart. “Inverse problems: a Bayesian perspective”. In: *Acta Numerica* 19 (2010), pp. 451–559. DOI: 10.1017/S0962492910000061.

[91] Jeferson O Almeida and Fernando A Rochinha. “A Probabilistic Learning Approach Applied to the Optimization of Wake Steering in Wind Farms”. In: *Journal of Computing and Information Science in Engineering* 23.1 (2022), p. 011003. DOI: 10.1115/1.4054501.

[92] Jeferson O Almeida and Fernando A Rochinha. “Uncertainty quantification of water-flooding in oil reservoirs computational simulations using a probabilistic learning approach”. In: *International Journal for Uncertainty Quantification* online (2023).

[93] Evangéline Capiez-Lernout and Christian Soize. “Nonlinear stochastic dynamics of detuned bladed disks with uncertain mistuning and detuning optimization using a probabilistic machine learning tool”. In: *International Journal of Non-Linear Mechanics* 143 (2022), p. 104023. DOI: 10.1016/j.ijnonlinmec.2022.104023.

[94] Charbel Farhat, Radek Tezaur, Todd Chapman, Philip Avery, and Christian Soize. “Feasible probabilistic learning method for model-form uncertainty quantification in vibration analysis”. In: *AIAA Journal* 57.11 (2019), pp. 4978–4991. DOI: 10.2514/1.J057797.

[95] Roger Ghanem and Christian Soize. “Probabilistic nonconvex constrained optimization with fixed number of function evaluations”. In: *International Journal for Numerical Methods in Engineering* 113.4 (2018), pp. 719–741. DOI: 10.1002/nme.5632.

[96] Roger Ghanem, Christian Soize, Cosmin Safta, Xun Huan, Guilhem Lacaze, Joseph C. Oefelein, and Habib N. Najm. “Design optimization of a scramjet under uncertainty using probabilistic learning on manifolds”. In: *Journal of Computational Physics* 399 (2019), p. 108930. DOI: 10.1016/j.jcp.2019.108930.

[97] Johann Guilleminot and John E Dolbow. “Data-driven enhancement of fracture paths in random composites”. In: *Mechanics Research Communications* 103 (2020), p. 103443. DOI: 10.1016/j.mechrescom.2019.103443.

[98] Roger Ghanem, Christian Soize, Loujaine Mehrez, and Venkat Aitharaju. “Probabilistic learning and updating of a digital twin for composite material systems”. In: *International Journal for Numerical Methods in Engineering* 123.13 (2022), pp. 3004–3020. DOI: 10.1002/nme.6430.

[99] Olivier Ezvan, Christian Soize, Christophe Desceliers, and Roger Ghanem. “Updating an uncertain and expensive computational model in structural dynamics based on one single target FRF using a probabilistic learning tool”. In: *Computational Mechanics* 71 (2023), pp. 1161–1177. DOI: 10.1007/s00466-023-02301-2.

[100] Maarten Arnst, Christian Soize, and Kevin Bulthies. “Computation of Sobol indices in global sensitivity analysis from small data sets by probabilistic learning on manifolds”. In: *International Journal for Uncertainty Quantification* 11.2 (2021), pp. 1–23. DOI: 10.1615/Int.J.UncertaintyQuantification.2020032674.

[101] Christian Soize and André Orcesi. “Machine learning for detecting structural changes from dynamic monitoring using the probabilistic learning on manifolds”. In: *Structure and Infrastructure Engineering Journal* 17.10 (2021), pp. 1418–1430. DOI: 10.1080/15732479.2020.1811991.

[102] Kuanshi Zhong, Javier G Navarro, Sanjay Govindjee, and Gregory G Deierlein. “Surrogate modeling of structural seismic response using Probabilistic Learning on Manifolds”. In: *Earthquake Engineering and Structural Dynamics* Online (2023), pp. 1–22.

[103] R.R. Coifman and S. Lafon. “Diffusion maps”. In: *Applied and Computational Harmonic Analysis* 21.1 (2006), pp. 5–30. DOI: 10.1016/j.acha.2006.04.006.

[104] Stephane Lafon and Ann B Lee. “Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization”. In: *IEEE transactions on pattern analysis and machine intelligence* 28.9 (2006), pp. 1393–1403. DOI: 10.1109/TPAMI.2006.184.

[105] Christian Soize. “Polynomial chaos expansion of a multimodal random vector”. In: *SIAM-ASA Journal on Uncertainty Quantification* 3.1 (2015), pp. 34–60. DOI: 10.1137/140968495.

[106] Hamza Boukraichi, Nissrine Akkari, Fabien Casenave, and David Ryckelynck. “A priori compression of convolutional neural networks for wave simulators”. In: *Engineering Applications of Artificial Intelligence* 126 (2023), p. 106973. DOI: 10.1016/j.engappai.2023.106973.

[107] A.W. Guess. “Calculation of perforated plate liner parameters from specified acoustic resistance and reactance”. In: *Journal of Sound and Vibration* 40.1 (1975), pp. 119–137. DOI: 10.1016/S0022-460X(75)80234-3.

[108] Gual J Zhong K and Govindjee S. *PLoM python package*. Version v1.0. 2021. URL: https://github.com/sanjayg0/PLoM.

[109] Christian Soize, Roger Ghanem, Cosmin Safta, Xun Huan, Zachary P. Vane, Joseph C. Oefelein, Guilhem Lacaze, and Habib N. Najm. “Enhancing model predictability for a scramjet using probabilistic learning on manifolds”. In: *AIAA Journal* 57.1 (2019), pp. 365–378. DOI: 10.2514/1.J057069.