Prediction of aircraft vibration environment using machine learning

Stéphane Février

université PARIS-SACLAY

Prediction of aircraft vibration environment using machine learning

Prédiction de l'ambiance vibratoire d'un avion par apprentissage automatique

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°579 : Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMaG)
Spécialité de doctorat: Mécanique des solides
Graduate School : Sciences de l'ingénierie et des systèmes. Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **LISN** (Université Paris-Saclay, CNRS), sous la direction de **Bérengère PODVIN**, directrice de recherche CNRS, le co-encadrement de **Lionel MATHELIN**, chargé de recherche CNRS, la co-supervision de **Frédéric GIORDANO**, ingénieur Dassault Aviation, et **Stéphane NACHAR**, ingénieur Dassault Aviation.

Thèse soutenue à Paris-Saclay, le 05 Décembre 2023, par

Stéphane FÉVRIER

Composition du jury

Membres du jury avec voix délibérative

David RYCKELYNCK Professeur des universités, MINES ParisTech, Centre des Matériaux	Président & Rapporteur
Christophe DESCELIERS Professeur des universités, Université Gustave Eiffel, Laboratoire MSME	Rapporteur & Examinateur
Mathilde CHEVREUIL Maître de conférences, Nantes Université, Laboratoire GeM	Examinatrice
Laurent OUDRE Professeur des universités, ENS Paris-Saclay, Centre Borelli	Examinateur

THESE DE DOCTORAT

NNT : 2023UPAST181

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)

Titre : Prédiction de l'ambiance vibratoire d'un avion par apprentissage automatique
Mots clés : Aéronautique, vibrations des structures, essais en vol, méthode orientée données, apprentissage automatique

Résumé : L'ambiance vibratoire correspond aux sollicitations haute fréquence subies par les équipements embarqués durant un vol. Ces sollicitations peuvent être de plusieurs types (aléatoires plus ou moins large bande, sinusoïdales) et avoir plusieurs origines (interaction avec champs aérodynamiques, machines tournantes, etc.). Leur étude permet de spécifier les niveaux vibratoires que les équipements doivent être capables de supporter pour garantir un bon fonctionnement sur la durée de vie de l'avion. Les moyens de simulation usuels du comportement dynamique de l'avion ne permettent pas d'être prédictif dans la plage de fréquence à considérer, qui s'étend jusqu'à 2000 Hz. Des essais en vols sont réalisés, générant un important volume de données accélérométriques à haute fréquence d'échantillonnage (5000 Hz), accompagnées des mesures de différents paramètres représentant l'état et l'attitude avion à tout instant.

L'objectif de cette thèse est la mise en place de modèles permettant de prédire l'ambiance vibratoire subie par un équipement en vol sur un avion au moyen de méthodes d'apprentissage, et ce à partir d'un ensemble de paramètres avion. Pour cela, la thèse est structurée en trois parties : l'exploitation et la représentation des données, la prédiction en tout point de vol, et la prédiction en tout point d'une structure avion.

La première étape est de mettre en oeuvre des algorithmes permettant l'extraction et le traitement des données d'essais à très haute fréquence d'échantillonnage. Un accent est mis sur la représentation des données, avec la proposition d'une méthode de séparation du large bande et des raies cinématiques pouvant se combiner à des méthodes de réduction orientées métier.

La deuxième étape est de développer des modèles de régression pour prédire les niveaux vibratoires mesurés à partir des paramètres avion. Différents modèles sont proposés pour différentes problématiques de prédiction (raies cinématiques au sol, large bande en phases stationnaires, extension au transitoire), et une attention particulière est portée à l'évaluation des capacités de généralisation de ces méthodes, offrant de bonnes performances. Une étude de l'interprétabilité de certains modèles est proposée en se basant sur la méthode SHAP, provenant de la théorie des jeux coopératifs.

La troisième étape consiste en la généralisation des prédictions à tout point d'une structure avion. Un modèle numérique réduit est tout d'abord utilisé avec l'objectif d'analyser les chemins de propagation des vibrations, et un algorithme itératif permettant d'identifier le chemin principal est proposé. Enfin, un algorithme de traitement du signal sur graphe permettant d'inférer une structure à partir de mesures spatiales ponctuelles est évalué. Les structures apprises sont cohérentes avec les connaissances métier et offrent de bonnes performances en reconstruction de signaux.

Les outils proposés par cette thèse ont des applications aussi bien auprès des essais en vol que pour l'aide à la conception. À terme, ils pourraient servir de base pour développer des méthodes d'hybridation de modèles numériques et de données expérimentales, et ainsi permettre la prédiction dans un avion complet encore en phase de conception.

Title: Prediction of aircraft vibration environment using machine learning
Keywords: Aeronautics, structural vibrations, flight testing, data-driven, machine learning

Abstract: During a flight, on-board equipment are subjected to a high frequency vibration environment and must meet their nominal performance. The excitations can be of several types (broadband random, sinusoidal) and have several sources (interaction with aerodynamic fields, rotating machinery, etc.). Their study allows to specify the vibration levels that equipment must be able to withstand and demonstrate that they will continue to operate satisfactorily throughout their life. The current means to simulate an aircraft's dynamic behavior do not allow useful predictions in the frequency range of interest, which extends up to 2000 Hz. In consequence, flight tests are conducted, generating a large volume of high sampling frequency data from accelerometers (5000 Hz), along with measurements of various parameters representing the state and attitude of the aircraft at any given time.

The objective of this thesis is to establish models for predicting the vibrational environment experienced by equipment in flight on an aircraft using machine learning methods, based on a set of aircraft parameters. To achieve this, the thesis is structured into three parts: data extraction and representation, prediction at any point in the flight, and prediction at any point within an aircraft structure.

The first step is to implement algorithms for the extraction and processing of data generated at a high sampling rate. Particular attention is paid to data representation, and a method for separating the broadband component from kinematic lines is proposed, which can be combined with field-specific reduction methods.

The second step involves the implementation of regression models to predict measured vibrational levels from aircraft parameters. Various models are proposed for different prediction challenges: kinematic lines during ground tests, broadband levels during stabilized flight phases, and extension to unsteady flight phases, with a particular focus on evaluating the generalization capabilities of these methods. Good performances are obtained and analyzed using the SHAP method, which is based on interpretability concepts from game theory.

The third step involves generalizing predictions to any point within an aircraft structure. Initially, a reduced numerical model is used to analyze vibration propagation paths, and an iterative algorithm for identifying the main path is proposed. Then, a Graph Signal Processing algorithm for inferring a structure from point-wise spatial measurements is evaluated. The learned structures align with field-specific knowledge and demonstrate good performance in signal reconstruction.

The tools introduced in this thesis have applications in both flight testing and aircraft design. In the long term, they could serve as a foundation for developing hybrid methods that leverage numerical models and experimental data, thereby enabling meaningful predictions for an entire aircraft from the design phase.

Remerciements

Ces trois années de thèse ont été une expérience remarquablement enrichissante, marquée par un apprentissage permanent et modelée par de nombreuses interactions. Aussi, je souhaiterais remercier les personnes qui ont contribué à rendre cette aventure possible.

Avant tout, je tiens à exprimer ma profonde gratitude envers mes encadrants pour leur soutien sans faille, ainsi que pour la liberté académique et la confiance qu'ils m'ont accordées. Je remercie tout d'abord Bérengère Podvin qui a dirigé cette thèse avec une bienveillance et un soutien exceptionnels, s'investissant continuellement pour garantir son bon déroulement. Mes remerciements vont également à Lionel Mathelin pour ses idées novatrices et son intérêt marqué pour le sujet, ainsi que pour ses conseils de rédaction. Merci à Stéphane Nachar pour son soutien quotidien et nos nombreux échanges, pour ses enseignements techniques, et pour tous les à-côtés. Enfin, un grand merci à Frédéric Giordano qui m'a énormément appris aussi bien sur la partie métier que sur la façon de transmettre les idées, ainsi que pour sa vision forte du sujet qui a joué un rôle précieux. Sans cette fameuse conversation a priori anodine sur les thèses, rien de tout cela n'aurait pu voir le jour.

Je remercie également mon jury de thèse, David Ryckelynck et Christophe Desceliers pour avoir accepté d'examiner ce manuscrit en tant que rapporteurs, ainsi que Mathilde Chevreuil et Laurent Oudre pour avoir participé à ma soutenance en tant qu'examinateurs. J'ai sincèrement apprécié nos échanges.

Merci à Dassault Aviation pour avoir financé ce projet, et notamment à Bruno Stoufflet pour avoir reconnu son potentiel. Je tiens également à exprimer ma gratitude envers Jean-Luc Guillen, sans qui l'idée initiale de la thèse n'aurait pas pu se concrétiser, et dont le soutien a été d'une grande aide. Je remercie Eric Garrigues pour ses suggestions éclairées de pistes à explorer et son regard expert sur le sujet, et je me réjouis de l'intérêt qu'il a pu y porter. Merci également à Isabelle Barber pour ses apports réguliers et fins, ayant permis d'ancrer le projet au cœur des problématiques industrielles. Enfin, merci à Ludovic Colo notamment pour les quelques mois ayant précédé la thèse où j'ai pu beaucoup apprendre sur les enjeux métier liés au sujet.

Je remercie toutes les personnes avec qui j'ai pu échanger et qui ont grandement contribué au bon déroulement de cette thèse, que ce soit techniquement ou humainement. Merci tout particulièrement à Racha Baker pour avoir fourni les modèles numériques réduits, et avec qui travailler était un réel plaisir. Merci également à Agatha, Amine, Arthur, Artur, Constance, Daniele, Elsa, Florent, Gen, Gustave, Karl, Kevin, Louis, Maël, Malik, Mathieu, Melvin, Michele, Nathan, Nemo, Remy, Sabrina, Sayu, Sean, Thibault, Vincent, ou encore Yanis. De manière plus générale, merci à l'ensemble des personnes avec qui j'ai pu interagir et qui ne sont pas expressément nommées, que ce soit côté LISN au sein du département Mécanique des fluides - Énergétique, ou côté Dassault Aviation au sein de CVAC et du laboratoire Phénomènes Dynamiques et Structures.

La vie et le fonctionnement du laboratoire ne seraient pas possibles sans les différentes équipes de soutien et support à la recherche, que je voudrais remercier. En particulier, merci à Valérie Séné pour sa disponibilité et son efficacité dans la gestion des missions, et à Laurent Pointal qui a rendu la soutenance de thèse possible d'un point de vue technique.

Merci à mes amis pour tous ces moments passés ensemble, je pense à vous. Enfin, rien de tout cela n'aurait été possible sans mes parents qui ont toujours su m'apporter leur soutien le plus sincère. Merci pour tout.

À mon père

Contents

Nomenclature

Relationships

$=$	Equality
$\triangleq$	Definition
$\equiv$	Equivalence
$\leftarrow$	Assignment

Vectors and matrices

a	Scalar
$\mathbf{a}$	Vector
$\mathbf{A}$	Matrix
$A_{ij}, A_{i:}, A_{:j}$	(i,j)-entry, i-th row, j-th column of matrix $\mathbf{A}$
$\mathbf{a}^\mathsf{T}, \mathbf{A}^\mathsf{T}$	Transpose of vector $\mathbf{a}$, transpose of matrix $\mathbf{A}$
$\mathbf{A}^*$	Conjugate transpose, or hermitian transpose, of matrix $\mathbf{A}$
$\operatorname{tr}(\cdot)$	Trace operator
$\operatorname{diag}(\mathbf{a})$	Diagonal matrix containing the vector $\mathbf{a}$
$\left\|\mathbf{a}\right\|_p \triangleq \left(\sum_{j=1}^{n} \left|a_j\right|^p\right)^{1/p}$	ℓ^p-norm of $\mathbf{a} \in \mathbb{R}^n$, ℓ^1 is the absolute norm, ℓ^2 the Euclidean norm
$\left\|\mathbf{A}\right\|_\mathrm{F} \triangleq \sqrt{\sum_{i,j} A_{ij}^2} = \sqrt{\operatorname{tr}(\mathbf{A}\mathbf{A}^*)}$	Frobenius norm of matrix $\mathbf{A}$
$\langle \cdot, \cdot \rangle$	Inner product
$\odot$	Hadamard product, or element-wise product
$\mathbf{1}, \mathbf{0}$	Constant unitary or null vector
$\mathbf{I}$	Identity matrix

Signal processing

β	Spectral resolution
$\mathbf{u}$	Temporal domain signal
$\mathbf{s}$	Frequency domain signal
$\widetilde{\mathbf{s}}$	Reduced frequency representation of $\mathbf{s}$
$s(f)$	Value of $\mathbf{s}$ at frequency f

$\mathbf{s}[f_1, f_2]$	Values of $\mathbf{s}$ in the frequency interval $[f_1, f_2]$
$\mathbf{b}, \mathbf{k}$	Broadband and kinematic lines spectra of $\mathbf{s}$
$\boldsymbol{\kappa}_i^{(h)}$	RMS values of kinematic line at point i for harmonic h
$\boldsymbol{\gamma}_{ij}^{(h)}$	Kinematic Lines RMS Ratio (KLR) from j to i at harmonic h

Machine Learning

$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{1,\dots,m}$	Dataset of m labeled datapoints
$\mathbf{x}, \mathbf{y}, \hat{\mathbf{y}}$	Input, output, and predicted output
$\phi, \phi_{\boldsymbol{\theta}}, \boldsymbol{\theta}$	Model, parameterized model, parameters
$C(\boldsymbol{\theta})$	Cost function
$\mathcal{X} \subset \mathbb{R}^n$	Input space of dimension n
$\mathcal{Y} \subset \mathbb{R}^d$	Output space of dimension d
Φ, Θ	Model space, parameters space

Graph

$\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$	Graph with nodes set $\mathcal{V}$, edges set $\mathcal{E}$, and weights set $\mathcal{W}$
$\mathbf{L}, \mathbf{D}, \mathbf{W}$	Laplacian, degree, and weighted adjacency matrix of $\mathcal{G}$

Numerical models

$\mathbf{Z}, \mathbf{H}$	Equivalent impedance matrix, Transfer Function matrix
$\mathbf{X}, \mathbf{F}$	Displacement matrix, Force matrix
$\mathbf{T}^{\mathsf{G}}, \mathbf{T}^{\mathsf{D}}$	Global transfer matrix, Direct transfer matrix
$\mathcal{H}$	Hilbert space

Acronyms

ANN Artificial Neural Network. 18, 47, 49, 71, 74–76, 79, 104, 106–114, 116, 118, 120, 123, 181

ATPA Advanced Transfer Path Analysis. 19, 127, 139–141, 143, 145, 152, 167, 175

BEM Boundary Element Method. 17, 129, 130, 132, 134

DSE Dynamic SuperElement. 19, 127, 134, 135, 137–139, 141, 143, 144, 146, 152, 166, 167, 169, 182

ERS Extreme Response Spectrum. 34

FDS Fatigue Damage Spectrum. 34, 179

FEM Finite Element Method. 17, 129–134, 143

FFT Fast Fourier Transform. 30, 31, 55, 90, 102, 115

FRF Frequency Response Function. 27, 135, 144–146

FTC Flight Test Center. 18, 35, 36, 43, 65, 70, 167, 169

FTI Flight Test Instrumentation. 17, 19, 26, 63, 89

GRU Gated Recurrent Unit. 78

GSP Graph Signal Processing. 3, 19, 127, 155, 158, 159, 167, 169, 170

GVT Ground Vibration Test. 27, 56

i.i.d. independent and identically distributed. 83, 92, 105, 108

KLR Kinematic Lines RMS Ratio. 14, 94–99, 127–129, 180, 181

LIME Local Interpretable Model-agnostic Explanations. 85, 87, 88

LSTM Long Short-Term Memory. 18, 77, 78, 116, 172, 181

MAE Mean Absolute Error. 79, 80

MALE Mean Absolute Log Error. 80, 93, 105, 107, 118, 121, 163, 165, 181, 185

MAPE Mean Absolute Percentage Error. 80

maxALE Max Absolute Log Error. 118

MLAR Mean Log Accuracy Ratio. 80

MSE Mean Squared Error. 47, 72, 73, 79

NMF Non-negative Matrix Factorization. 46, 47, 58–60, 92, 179

PCA Principal Component Analysis. 44–47, 49, 57–64, 179, 180

PSD Power Spectral Density. 17, 28–34, 37, 39, 41, 55, 56, 90, 92, 93, 95, 102, 112, 115, 116, 119, 120, 179, 185

ReLU Rectified Linear Unit. 60, 75, 104, 111

RMS Root Mean Square. 14, 31, 32, 37, 39, 41, 42, 53, 56–64, 89–99, 102, 103, 105, 109, 110, 112, 115, 118–121, 123, 161, 174, 179–181

RNN Recurrent Neural Network. 76–78, 116, 118–121, 123, 180, 181

SEA Statistical Energy Analysis. 17, 129, 132, 133, 139, 170, 172, 182

SHAP SHapley Additive exPlanations. 2, 3, 70, 76, 85, 87, 88, 112–114, 120, 123, 170, 175, 177, 181

SRS Shock Response Spectrum. 33, 34, 179

SVD Singular Value Decomposition. 45

t-SNE t-distributed Stochastic Neighbor Embedding. 64, 180

TPA Transfer Path Analysis. 138, 139

XGBoost eXtreme Gradient Boosting. 72, 73, 104, 106, 107, 109–114, 116, 118, 120

Introduction

Every aircraft passenger has already experienced in-flight or ground vibration. The prediction and mitigation of aircraft vibration is one of the critical aspects of passengers' comfort. In addition, aircraft vibration can have a significant impact on the safety and durability of the aircraft and its equipment [Vib12]. Therefore, accurate prediction of the vibration environment of an aircraft is essential for ensuring safe and efficient operations.

Vibration environment is a term used to describe the 10-2000 Hz dynamic loads experienced by the structure and on-board equipment during the use of the aircraft. They are generated by various external excitations, such as aerodynamic field interactions, or internal forces from rotating machinery, such as engines and pumps. Specifying the vibration environment of an aircraft must be done at an early stage in the design for civil and military aircraft and consists of providing the functional and fatigue levels that the equipment must withstand. While specifications must be given in the early design phase of the aircraft, the actual vibration environment can only be determined during flight testing, which occurs late in the development program.

The vibration environment is measured using uniaxial or triaxial accelerometers with a high sampling rate, here set at 5000 Hz. Located at the root of equipment assembly, they are connected to an FTI (Flight Test Instrumentation), which makes it possible to measure all the necessary information on the state of the aircraft at any given time, referred to from now as general parameters. The signals measured by the accelerometers are usually analyzed in terms of their frequency spectrum using PSDs (Power Spectral Densities).

There is currently no reliable and comprehensive approach to evaluate the dynamic behavior of a complete aircraft over the full range of frequencies considered, which extends up to 2000 Hz. As a result, specifications are based on existing standards and the experience of previous aircraft. This can lead to two negative outcomes: equipment overdimensioning, which has a negative impact on the overall aircraft mass, or equipment being lately qualified to demonstrate the required levels.

More precise specifications would require overcoming current computational limitations and developing reliable and comprehensive **numerical models** of the entire aircraft. Various models of the dynamic environment of an aircraft have been developed for different frequency ranges, but none are suitable for our specific needs.

In the low-frequency domain, current techniques focus on a modal behavior to extract responses localized in frequency. Deterministic techniques such as the Finite Element Method [ZTZ10] or the Boundary Element Method [BB94] are used. In practice, these methods are limited to low frequencies because they require a very fine discretization in space and time, and a high degree of interpolation to remove numerical pollution and dispersion effects [DBB99]. In the high-frequency domain, the interest is in describing a global behavior in terms of energy because the modal overlap is high. Statistical Energy Analysis [LM62, LDD95, Smi62, LB04, CLK05] is well suited for this frequency domain because it models the vibration energy transfers across partitions (subsystems) of a structure. The

energy flow between these partitions is described using a set of linear equations for the input, storage, transmission, and dissipation of energy with a set of coefficients that needs to be determined. The problem is that the response is averaged over frequency and space, and is hence not suitable for local characterization. In the mid-frequency range, neither low- nor high-frequency methods are suitable: low-frequency methods become too costly due to the required high mesh refinement to reduce the numerical pollution effect, and high-frequency methods do not provide local responses. Mid-frequency methods can be divided into hybrid approaches (combining low- and high-frequency methods) and Trefftz approaches (approximating the solution as a linear combination of shape functions that satisfy the equilibrium equations a priori), see an overview in [DOP12]. However, further work is required to make them applicable in a fully industrial context.

Prediction tools could benefit from the considerable wealth of information obtained in flight test data, which is typically used only in the latest design phase. The challenge of this thesis is to determine how to integrate this information into a predictive tool. Some of the issues to be addressed include data compression, feature extraction, and robust modeling. **Machine learning** offers a wide variety of techniques to deal with these different issues.

Among machine learning models, Artificial Neural Networks (ANNs) have been proved successful in predicting vibrations in a wide variety of applications such as aerodynamics, mining, or civil engineering. In the temporal domain, Long Short-Term Memory (LSTM) has been shown to be very effective in predicting the response of complex mechanical systems to a broadband excitation [SCC+22, NB18]. It was also able to predict excess vibration events for engine aircraft [EWHD16]. The LSTM has also been used to predict the vibration environment of electric cars and was found to outperform classical ARMAX (Auto Regressive Moving Average with eXternal inputs) models [DGD+20]. In the frequency domain, ANNs have been shown to successfully predict structural stress [WOdS20]. They have also been applied on turbulent flows to predict noise levels from limited pressure measurements in jets [TGRB20] and at the trailing edge of an airfoil [AF21]. Other applications of ANNs include the prediction of ground vibration due to blasting for mining applications [Sin04], where their performances were found to be superior to that of conventional regression [BP20]. ANNs have also been found to successfully determine ground vibration levels due to the passing of oncoming trains [PDCFN18].

The end goal is to efficiently leverage test data and machine learning models to be able to predict the vibration environment of a complete aircraft, as early as its design phase. To achieve this goal, the following steps are identified:

- The first step is to be able to extract and analyze the voluminous vibration environment data generated at the Flight Test Center. Suitable data representation and reduction methods must be implemented, along with practical tools for incorporating domain rules.
- The second step is to develop a methodology to predict vibration spectra at any point in the flight domain (or feature space for ground tests) using the previously shaped data and machine learning models. This has applications in anomaly detection, where a model is compared to measurements, as any discrepancy between the two would raise a flag and require the attention of an engineer. Another application is to optimize flight testing by identifying flight points that maximize the information contribution.

- The third step is to extend the prediction to any point in the structure of the aircraft, even non-instrumented ones where there is no accelerometer. This requires finding an appropriate structural representation. An application of such a method would be in FTI optimization, where the accelerometer placement is chosen to maximize the information contribution.

One perspective is to extend the prediction to a new aircraft. The general idea is to find an equivalent representation that can be tuned based on domain knowledge. It could be done without experimental data on the new aircraft to have rough estimates that can then be updated as the first tests are conducted. This method could help in the design phase by raising early flags about possible vibration environment problems and in the specification process.

To answer these challenges, the manuscript is organized around three parts, each of them summarized at their end. **Part I** focuses on data extraction and representation. Chapter 1 first introduces the vibration environment, and Chapter 2 details the algorithms implemented for dataset generation. As it uses several data representation techniques, they are introduced in Chapter 3 and then applied in Chapter 4 on flight test data. **Part II** tackles the problem of prediction through the flight domain. In Chapter 5, supervised learning tools are presented, with a focus on regression methods. They are then applied in Chapter 6 to predict the values of kinematic lines, generated by isolated and localized sources, and in Chapter 7 to predict the values of broadband levels. **Part III** is about extending the prediction through a complete aircraft structure. To this end, classical numerical models are first introduced in Chapter 8, along with model reduction through Dynamic SuperElements and Advanced Transfer Path Analysis. These tools are then applied in Chapter 9, where vibration propagation paths are identified using a combination of previously introduced tools and a new sorting algorithm. Finally, Chapter 10 presents graph theory and Graph Signal Processing before applying a graph learning method that allows to learn the underlying structure of data from measurements alone.

This thesis aims to explore innovative ways to answer this problem through a collaboration with Dassault Aviation. As of the industrial collaboration, a focus is made on the methodology and all figures are anonymized, although a maximum of information is kept.

Part I

Data-driven representation of vibration environment

1 - Vibration environment

Contents

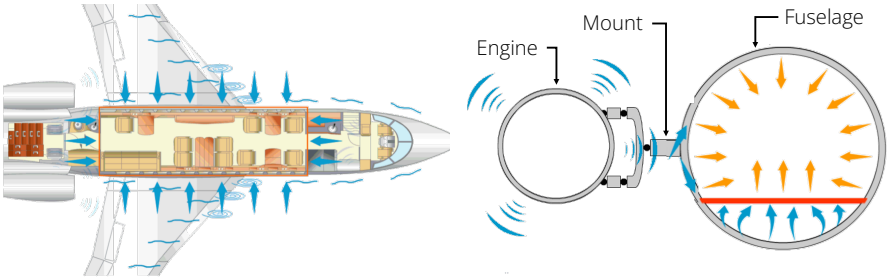
This part focuses on implementing the necessary tools to enable a predictive data-driven framework for vibration environment prediction. To begin, the vibration environment is introduced in this chapter. Then, the methodology developed to generate a reduced-order dataset is presented in the following chapters.

The vibration environment corresponds to the 10–2000 Hz dynamic loads undergone by the structure and on-board equipment during the use of the aircraft. Studying the vibration environment allows to specify the vibrational levels that equipment must be able to withstand throughout their life while continuing to operate satisfactorily. It is crucial to ensure the performance and mechanical strength of equipment [Woo72], and must be done at an early stage of the design for both civilian and military programs. The work presented in the following chapters focuses on civilian applications.

This chapter first presents the different sources of the vibration environment in Section 1.1. Equipment specifications are then introduced in Section 1.2, and flight tests to adjust them are detailed. A side note is given on additional ground tests. Then, the equipment substantiation procedure is presented in Section 1.3, with its different tests and an example of a qualification protocol. Finally, several spectral representations used to study the vibration environment of an aircraft are detailed in Section 1.4

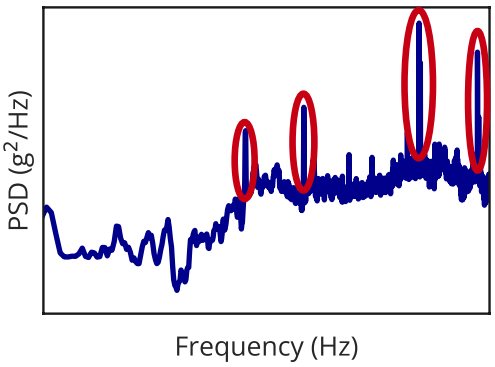
1.1 . Vibration sources

Vibration can originate from external forces, such as aerodynamic field interactions, or from the aircraft itself, such as with any rotating machinery (engines, pumps, and turbines) or on-board system. As reported in [HKM+01], examples of internal excitations include an imbalance of rotating engine parts; misalignment in hardware assembly; additional magnetic, aerodynamic, or hydro-dynamic forces; etc. As illustrated in Figure 1.1, different vibration sources give rise to distinct responses, distinguishable in the frequency domain. Circled in red are what are called **kinematic lines** in this manuscript, originating from rotating machinery and detailed hereinafter. An important part of this manuscript is based on the idea of splitting measured spectra between a **broadband** component and a collection of local excitations, the kinematic lines, see Section 4.1. Here, vibration sources of three types are detailed: aerodynamic sources, engine vibrations, and shocks.



(a) Aerodynamic field

(b) Engines and rotating machinery



(c) Kinematic lines identified in red

Figure 1.1: Measured vibrations are the result of various excitations. This illustration shows the influence of aerodynamic field interaction and engines on a spectrum. Figures 1.1a and 1.1b are courtesy of Dassault Aviation.

1.1.1 . Aerodynamic sources

The interaction[1] between the aircraft skin and the aerodynamic field generates vibrations during flight. These vibrations are broadband as they span the entire considered frequency domain. Their level depends on the aircraft's state and attitude, its configuration, and the flight point. Maneuvers also generate different levels than stabilized flight.

[1] Only the effect of the aerodynamic field on the aircraft skin is considered in this thesis.

In addition to broadband levels, some particular aerodynamic effects can produce noticeable vibrations in some cases and for specific conditions:

- **Buffeting** is an aerodynamic excitation that can originate either from turbulence in the flow stream or the dynamic of an aerodynamic shock around the structure. Generally, it occurs at high angles of attack during maneuvers, for which the airflow is unsteady and turbulent. The generated vibrations are typically rather broadband, between 10 and 50 Hz. They can induce strong levels of vibrations if the excitation frequency is close to an eigenfrequency of the structure due to structural resonances.
- An **aerodynamic whistle** can happen when an airflow stream encounters a peculiar geometry, creating a whistle phenomenon. While it appears only for specific aerodynamic conditions, it can produce very high vibration levels. The frequency of the phenomenon depends on the airflow speed and the local external geometry.

1.1.2 . Engines vibrations

Aircraft engines and all rotating machinery (pumps, turbines, other engines) are made of several rotating parts. The imbalance of these parts creates vibrations at their own rotation frequency, as well as at specific harmonics. These are narrow band levels with a variable frequency with respect to the rotating speed of source equipment. As a result, they are named "kinematic lines" in this manuscript, as they are spectral lines moving in frequency. They are identified in Figure 1.1, encircled in red.

More generally, all rotating machinery in the aircraft produces such levels. Depending on the aircraft considered, the following equipment are examples of vibration sources generating kinematic lines: the APU (Auxiliary Power Unit), the various EDP (Engine Driven Pumps), or the RAT (Ram Air Turbine).

1.1.3 . Shocks

In addition to aerodynamic and rotating machinery, several shocks can occur. While they are not studied in the manuscript, they are essential to specify the vibration environment of an aircraft. Landing is an example of shock that can introduce high levels loads on the landing gears, transmitted to the whole structure. Levels depend on various parameters such as the aircraft configuration, mass and attitude, the landing velocity and position, the pneumatics pressure, and the runway surface quality. The numerous combinations of these parameters add to the complexity of specification-making. If a modal coupling occurs between the landing gear and the equipment, high levels can be reported.

1.2 . Equipment General Specifications

The purpose of specifications is to verify the capacity of equipment to withstand the vibratory environment and operate correctly during their life profile [DO110, MIL08, GAM87].

Specifying the vibration environment of equipment consists of providing the functional and fatigue levels that equipment should withstand. The functional test represents the worst-case vibration (or envelope of worst-case conditions) of the environmental life cycle, whereas the fatigue test subjects equipment to accelerated fatigue levels, representative of an entire life in terms of damage.

Qualification tests are then performed on equipment to demonstrate that they meet the requirements. The substantiation process is detailed in the next section. Non-compliant equipment would have to be modified or redesigned. Determining the vibration environment levels is done in the early design phase of the aircraft but can only be verified during flight tests.

1.2.1 . Preliminary and standard specifications

The purpose of preliminary specifications is to allow the manufacturer to be able to design and produce an equipment that fulfills both regulatory[2] and functional requirements. They must be well-designed to comply with standards while fulfilling feasibility and cost criteria. On civil aircraft, equipment are often COTS (Commercial off-the-shelf). They are qualified to standards and do not require tailoring as the return on experience is satisfactory. On military aircraft, the vibration environment can be specific and severe; therefore, tailoring is of great interest.

More generally, for a new aircraft, preliminary specifications are elaborated using the knowledge of similar aircraft used for similar missions. Standard specifications should be used if no aircraft is similar enough [MIL08]. Standard specifications exist to cover all the types of vibration sources presented before in Section 1.1.

For in-flight vibrations, in-zone specifications are considered. These are obtained by extrapolating the envelope of levels measured by several accelerometers. A safety factor is also taken into account. In the case of a new aircraft with no existing area specifications, experience from previous aircraft is used. Finally, equipment on suspending frame have specific specifications, as vibration isolators attenuate high-frequency vibration inputs to equipment but add low-frequency resonances to the dynamic system.

1.2.2 . Flight-adjusted specifications

Standard specifications are more severe than each individual equipment levels, as they are extrapolated levels. They can be tailored thanks to flight-adjusted specifications [GAM87] [MIL08].

To do so, a dedicated FTI (Flight Test Instrumentation) is required, and uniaxial or triaxial accelerometers are used to measure the levels of equipment. They are located on a rigid structure as near as possible to the fixation points to avoid measuring acceleration due to local modes. Moreover, the measure should be done on a stationary signal for as long as possible. More details on this part are below in Section 1.4. Along with vibration measurements, some other aircraft quantities are measured, giving information about the state and attitude of the aircraft: altitude, control surface position, engine revolutions per minute, etc. All of these parameters that help to describe the general state of the aircraft are called **general parameters** in this manuscript. These parameters are acquired through various sensors at different sample rates.

The whole flight domain should be covered during environmental data-gathering flights. On selected flight points, levels and maneuvers[3] are performed. If the whole flight domain cannot be covered, levels should be extrapolated to its limits. In the case of military aircraft, the choice of loads (e.g., fuel tank position) can heavily change the vibration levels. As environmental data-gathering flights cannot be performed in each possible configuration, the worst configurations are selected

[2] Regulatory requirements can be standards such as the DO-160 [DO110].

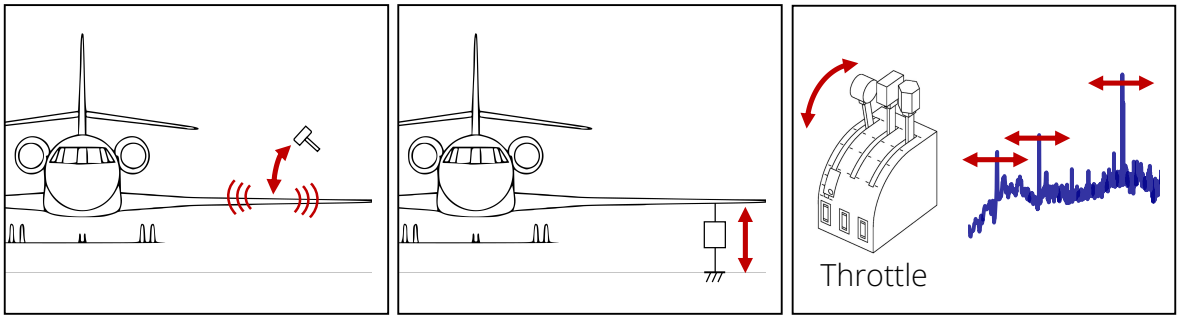
[3] Examples of maneuvers include rolls, speed brake extension, snubber stop, and side-slip.

from knowledge and experience.

From the gathered data, significant phenomena generating levels are identified and understood. Then, adjusted specifications are produced. An example of flight tests conducted to assess the vibration environment of an aircraft is presented in [CFB+02], where flight vibration data from a NASA F-15B equipped with a Flight Test Fixture II is compared to flight-qualification random vibration test standards.

1.2.3 . A note on additional ground vibration tests

In addition to the flight tests presented before, several additional vibration tests can be performed on the ground. They are not used to produce specifications or to qualify equipment but are presented as they provide helpful information. An illustration of 3 tests presented below is given in Figure 1.2



(a) Impact hammer test (b) GVT (Ground Vibration Test) (c) Engine vibration survey test

Figure 1.2: Illustration of different vibration tests performed on the ground.

- An **impact hammer test** or tap test, see Figure 1.2a, consists of using a hammer on the structure to measure its response. As the impact hammer is able to measure the impact force on the structure under test, it allows the computation of a Frequency Response Function (FRF).
- A **GVT (Ground Vibration Test)** measures the frequencies and deformed shapes of the flexible modes of the aircraft, as well as the corresponding modal damping. It is required for civilian certification [CS207] and military specification [AIR79]. A GVT is a critical step that is generally performed a few months before the first flight tests and provides (i) experimental data on which to adjust numerical models, (ii) validation of the overall assembly of the structure, and (iii) valuable data to prepare and monitor flight tests. Multiple shakers are used to vibrate the structure. For aircraft equipped with fly-by-wire[4], structural responses to dynamic excitations of control surfaces can also be measured.
- An **engine vibration survey** can be carried out when the aircraft is on the ground. It consists of exploring the engines' operating regimes while measuring the aircraft response at various points. This can be performed with both engines synchronized or not. This is different from order tracking, as complete spectra are considered.

[4]Numerical flight controls instead of manual ones.

1.3 . Substantiation

Before installing new equipment in the aircraft, they need to be substantiated using a range of tests assuring that they can handle the specification levels. They can be split into functional and endurance tests.

Functional testing is conducted to verify that equipment operate as required while exposed to worst-case operational vibration. Equipment performance must remain nominal [MIL08]. In some cases, a reduced envelope can be used, for example, if an equipment can only be used in a limited flight domain.

Endurance levels represent an entire life cycle through an accelerated fatigue test. It is generally not required to have the equipment functional during the endurance phase of the test, but it is required to have it powered up. Moreover, it needs to function properly before and after the endurance test during the functional test. To be able to reduce the effective test time, Basquin's law is used [GAM87], which approximates the central part of a Wöhler curve[5] as a linear relationship:

$$N\sigma^b = C, \tag{1.1}$$

with N the number of cycles, σ the stress, b Basquin's material parameter, and C the material constant. By doing so, a vibratory fatigue equivalency can be determined between the vibratory fatigue of an entire life cycle and the accelerated fatigue test. Given that points A and B are on the same curve, N_A cycles at σ_A are equivalent to N_B cycles at σ_B, for $N_A > N_B$ and $\sigma_A < \sigma_B$.

Assuming that the equipment can be considered as a linear system, the test time can thus be reduced using the following relationship:

$$\mathbf{s}_2 = \mathbf{s}_1 \left(\frac{t_1}{t_2}\right)^{\frac{2}{b}}, \tag{1.2}$$

where $t_1 > t_2$ are durations and $\mathbf{s}_1 < \mathbf{s}_2$ are Power Spectral Densities, the most common frequency representation of random signals in vibration environment, see details in Section 1.4.1. To avoid vibration levels stronger than the functional level, time is generally reduced such that levels are equivalent to the functional level.

When equipment life profile is not defined, assuming a partition of the life profile between 80% of **current cases** and 20% of **severe cases** is standard practice. Current cases regroup mean speed, high altitude level flight. Severe cases regroup high speed, low altitude, and maneuver flight.

To qualify an equipment, the boundary conditions should be representative of its aircraft integration. Tests are performed using a shaker, with a connection as rigid as possible to avoid any coupling between the equipment and the shaker.

1.4 . Spectral representations

As presented in Section 1.2.2, the vibration environment is measured using accelerometers, which, as their name suggests, measure acceleration in g through time.

[5] Also known as S-N curve.

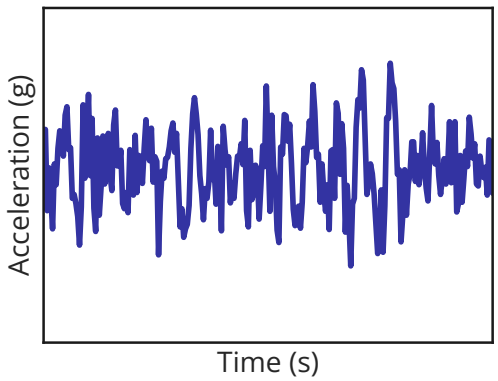


Figure 1.3: Illustration of a temporal signal of 0.05 s sampled at 5000 Hz.

While a temporal signal contains a substantial amount of information, it can be challenging to interpret at such high sample rates, as illustrated in Figure 1.3. This can be rendered more accessible through spectral representations, allowing to estimate the frequency content of the signal. As the vibration environment is random by nature, specific tools need to be used, generally relying on a stationary and/or ergodic assumption. In this section, several representations that are commonly used to assess and specify the vibration environment are presented.

1.4.1 . Power Spectral Density

A PSD (Power Spectral Density) is used to describe the distribution of power in a signal as a function of frequency. It is particularly useful to analyze random signals measured by accelerometers. An illustration is provided in Figure 1.4.

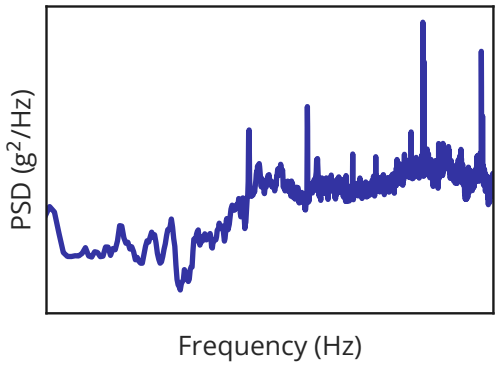


Figure 1.4: Illustration of a Power Spectral Density on a log-log scale.

A PSD is only defined for second-order stationary and ergodic signals [Lal09c]. Second-order stationary means that variance and autocovariance do not change over time. The ergodicity assumption implies that measuring one realization of the signal for a long enough time is sufficient to compute the statistical properties of the entire signal. These two assumptions come from the means of estimating a PSD, as presented below.

To define a PSD, Fourier transforms are usually performed to describe the signal frequency components. The power spectrum is the magnitude squared of the Fourier transform and provides a way to measure the power contained in each frequency component.

Let us denote u a temporal signal. Using Parseval's theorem, its PSD is defined as:

$$s \triangleq \lim_{T \to +\infty} \frac{2}{T} \left| \mathrm{FT}(u) \right|^2 , \tag{1.3}$$

with $\mathrm{FT}(u)$ the Fourier Transform of the temporal signal u and T the total time of the signal u. For an accelerometer signal in g, the resulting PSD is expressed in g^2/Hz.

Another way to calculate a PSD is to use the Wiener-Khinchin theorem, which states that the PSD can be found as the Fourier transform of the autocorrelation function:

$$s \triangleq \left| \mathrm{FT}(r(u)) \right|^2 , \tag{1.4}$$

with $r(u)$ the autocorrelation function of the non-windowed signal u, provided that u is ergodic.

To *estimate* the PSD, the signal u is typically sampled resulting in a discrete vector $\mathbf{u}$. Multiple ways to estimate a PSD exist, as exact calculations are often difficult or impossible due to limitations in data or computational power. Common methods are:

- Periodogram, which considers directly the periodogram or power spectrum of a signal.
- Bartlett's method [Bar48][Bar50], which averages the periodograms taken on multiple segments to reduce the variance of the estimation.
- Welch's method [Wel67], a windowed version of Bartlett's method, using overlapping segments.
- Multitaper method [Tho82], based on periodograms, uses multiple orthogonal tapers[6] to perform independent estimates and reduce the variance of the final estimation.

These methods are non-parametric and thus do not assume that the process has any particular structure. There are also parametric methods, making assumptions about the structure of the underlying stationary stochastic process, and semi-parametric methods, using a non-parametric framework with the additional assumption of model sparsity.

Here, a focus is made on Welch's method. The following parameters are to be set to estimate a PSD using Welch's method: the FFT (Fast Fourier Transform) windows size, overlap, and type.

- FFT windows size determines the spectral resolution of the PSD, see below in Eq. (1.6). Large windows provide better frequency resolution, but the PSD estimate becomes more sensitive to small variations in the signal, resulting in a higher variance.
- The windows overlap is the ratio between the number of samples on a given FFT covered by the following FFT, by the total number of samples of the FFT. Increasing the FFT windows overlap percentage allows for more averaging, which reduces the error variance of the final PSD estimate while increasing the computational complexity.

[6] Also known as Slepian sequences.

- Windowing allows to counter the leakage effect that can occur when applying an FFT on a non-periodic signal due to the discontinuities at the edges. Several types of window functions exist, such as Hanning, Hamming, Bingham, Blackman, FlatTop, and Kaiser-Bessel. Only the Hanning window is used in this manuscript and is defined as:

$$\mathrm{hann}(t) \triangleq 0.5\left(1-\cos\frac{2\pi t}{n}\right), \quad 0 \leq t \leq n, \tag{1.5}$$

with $n+1$ the window length.

The choice of parameters depends notably on the duration of the signal segment on which the PSD is estimated. A minimum number of degrees of freedom must be used to ensure a certain confidence level. From the military norm GAM-EG-13 [GAM87], more than 100 degrees of freedom are to be used to attain a 90% confidence level.

The spectral resolution β of a PSD is the interval between two consecutive points on a PSD. It is defined as:

$$\beta \triangleq \frac{\text{sample rate}}{\text{FFT windows size}}. \tag{1.6}$$

The number of degrees of freedom is defined as:

$$N_{\text{d.o.f.}} \triangleq 2\mu\frac{\beta T - R}{1-R}, \tag{1.7}$$

where R is the overlap proportion. The statistical error is defined as:

$$\varepsilon \triangleq \sqrt{\frac{1-R}{\mu(\beta T - R)}}, \tag{1.8}$$

where μ is a coefficient that depends on the FFT window type. According to the GAM-EG-13 [GAM87], the statistical error should be below 0.14.

1.4.2 . Root Mean Square

The RMS (Root Mean Square) value of a signal is a statistical measure of its magnitude. It is interesting because it takes into account the time history of the signal and its amplitude, which is directly related to the energy content and, therefore, the destructive power of the signal. The broadband RMS value is often studied, and Figure 1.5 shows the cumulative RMS value over the frequency on a given spectrum.

On a temporal signal u discretized into $\mathbf{u}$, it is defined as:

$$\mathrm{RMS}(\mathbf{u}) \triangleq \sqrt{\frac{1}{T}\int_0^T u^2(t)\mathrm{d}t} \equiv \sqrt{\frac{1}{N}}\left\|\mathbf{u}\right\|_2, \tag{1.9}$$

where T is the total time of the continuous form of $\mathbf{u}$, N is the number of samples in the discrete form of $\mathbf{u}$, and $\|\cdot\|_2$ is the ℓ^2 norm. This norm can be defined more generally: let $p \geq 1$, the ℓ^p-norm of a vector $\mathbf{y}$ of dimension d is defined as:

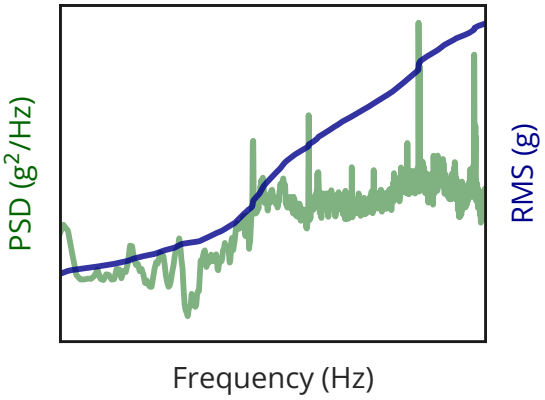


Figure 1.5: Cumulative RMS value (blue) of a PSD (green), using a log-log scale.

$$\|y\|_p \triangleq \left(\sum_{j=1}^{d} |y_j|^p \right)^{1/p}. \tag{1.10}$$

The ℓ^1-norm is the sum of absolute values, and the ℓ^2-norm is the Euclidean distance. When $p \to \infty$, it corresponds to the maximum absolute value of the vector:

$$\|y\|_\infty \triangleq \max_j |y_j|. \tag{1.11}$$

The RMS can also be defined as the square root of the area under the PSD curve, which is equivalent to computing the RMS of the associated temporal signal. The RMS value computed from the PSD $\mathbf{s}$ in the band $b = [f_1, f_2]$ is defined as:

$$\mathrm{RMS}(\mathbf{s}[f_1, f_2]) \triangleq \sqrt{\int_{f_1}^{f_2} s(f) \mathrm{d}f}. \tag{1.12}$$

If the PSD $\mathbf{s}$ is in g^2/Hz, then the RMS value is in g, as the original signal $\mathbf{u}$.

1.4.3 . Spectrogram

A spectrogram is a time-frequency representation of a signal. It is calculated using successive Fourier transforms[7], and allows to visualize the variation over time of a non-stationary signal frequency content.

The generic display is with time on the x-axis, frequency on the y-axis, and amplitude as a color or on a z-axis for 3D plots. Figure 1.6 shows the spectrogram response during a take-off. The speed of the engines first increases, which can be seen by several kinematic lines rising in frequency, then broadband levels appear.

[7] Wavelet transforms can also be used.

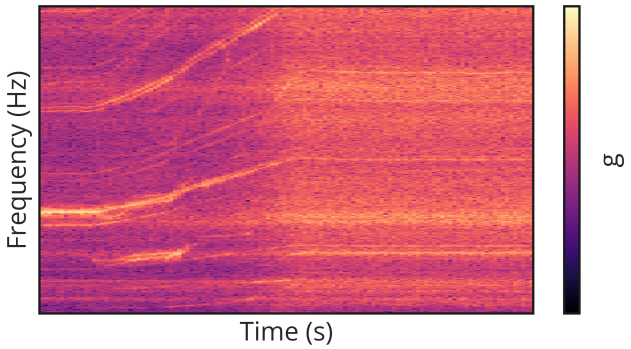


Figure 1.6: Illustration of a spectrogram during a take-off: kinematic lines can be seen moving up in frequency, then broadband levels appear when they reach the max regime. The colorbar uses a log scale.

While spectrograms are a powerful tool for analyzing the frequency content of vibration signals, the trade-off between time resolution and frequency resolution must be carefully set to capture the phenomena of interest. They can also be computationally intensive, especially for long signals.

1.4.4 . Equivalent Response Spectra

When dealing with transient dynamic stresses, PSD estimates have the drawback of averaging the measured levels. This results in lower levels than actually experienced by the structure. To assess the severity of transient vibrations or shocks, which are short-term force pulses, **SRS (Shock Response Spectrum)** can be used [Lal09b]. This method was initially proposed to study the effects of earthquakes on buildings [Bio32]. It is not used later in the manuscript but is presented to provide a comprehensive overview of the vibration environment.

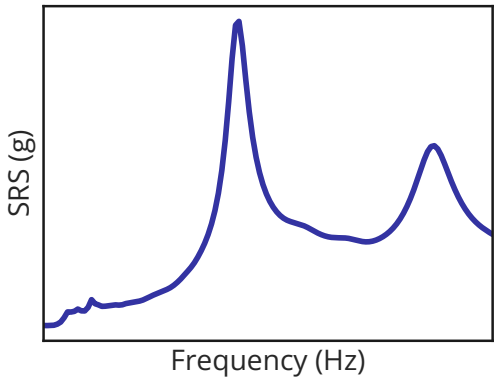


Figure 1.7: Illustration of a Shock Response Spectrum.

An SRS is an equivalent spectrum consisting of the maximum response of the mass of a damped single degree of freedom system of frequency f_n, for $f_n \in [10, 2000]$. This is a strong simplification that assumes decoupled linear systems but can be a valuable tool for studying shocks.

The temporal input is fed to a 1-degree-of-freedom system, and only the maximum response is kept. By doing this for several frequencies in parallel, a spectrum can be produced. An SRS can also be calculated from a PSD at the expense of some assumptions.

The **ERS (Extreme Response Spectrum)** is comparable to the SRS and also used to study vibrations [Lal09d]. It is a primary spectrum, meaning it gives the highest response observed during the vibration or shock.

In the same way as for the SRS, the **FDS (Fatigue Damage Spectrum)** [Lal09a][Lal09d] represent the equivalent fatigue damage experienced by a linear one degree-of-freedom system for each frequency, for a given damping ratio and for a given Basquin's law parameter to represent the Wöhler curve of the material constituting the structure. It is also not used later in the manuscript.

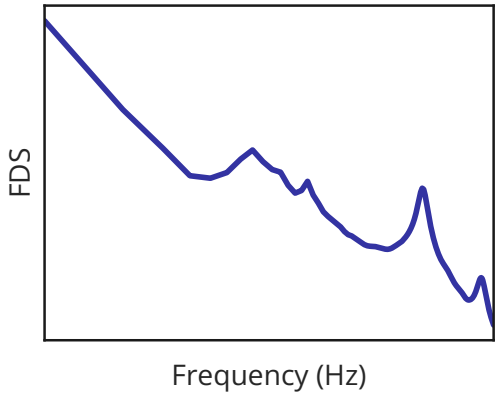


Figure 1.8: Illustration of a Fatigue Damage Spectrum on a log-log scale.

As with the SRS, the FDS can be computed from a temporal signal or from a PSD, in which case several assumptions need to be made on the original signal.

2 - Generating a vibration environment dataset

Contents

Data from the Flight Test Center (FTC) forms the core of this thesis. While extremely valuable and rich in information, its raw volume makes it challenging to feed into machine learning algorithms. As such, specific steps are taken to make it practical in the context of vibration environment prediction.

This chapter begins with a presentation of the available data and the challenges associated with it. Then, the algorithm implemented to generate a dataset that allows to analyze and model the vibration environment from flight test data from a data-driven perspective is detailed. This algorithm relies on several tools: the rule-based ones, such as flight phase identification or kinematic line frequency, are detailed in Section 2.3. The data transformation tools require a comprehensive description and are presented theoretically in Chapter 3 and then applied to experimental test data in Chapter 4.

2.1 . Available test data

The test data from the Flight Test Center is presented in this section with regard to its volume, its structure, and its integrity. This assessment justifies the need for the preprocessing and transformation steps implemented.

2.1.1 . Volume

To study the vibration environment, a set of **accelerometers** is used as presented in Section 1.2.2. As the measurements have to be analyzed up to 2000 Hz, the sample rate of accelerometers should be at least twice this value according to Shannon theorem. In practice, the sample rate is linked to the acquisition system, more specifically to its filters. In the industrial case of Dassault Aviation, the sample rate of accelerometers is set to 5000 Hz. This high sample rate makes the data very large.

In addition, a lot of accelerometers are placed on the aircraft, at least one for each equipment that needs to be specified. In total, a standard business jet aircraft can be instrumented with more than 50 tri-axial accelerometers, thus more than 150 channels.

In addition to the accelerometers, other parameters must be acquired to understand the state and attitude of the aircraft, called **general parameters** in this manuscript, as mentioned in the previous chapter. Everything that needs to be measured is acquired, sometimes twice or thrice if redundancy is required. This gives information at any moment on where the aircraft is in the flight domain, what the state of its engines is, what configuration it is in, and what set of equipment is activated. Although their sampling rate is much lower than that of accelerometers (and varies from parameter to parameter), thousands of general parameters are acquired.

All of this data is acquired for several flights that can last several hours. More than one aircraft is tested. This leads up to a tremendous amount of data. For example, a test campaign of 30 flights and several ground tests takes up 2.6 TB of vibration data and 13 TB of general parameters, without compression.

2.1.2 . Structure and integrity

The data is stored in an in-house binary file format that is specific to the needs of the Flight Test Center. Because the data is in the form of time series, the data structure itself is challenging.

As mentioned above, the general parameters are acquired at different sample rates. Consequently, resampling may be required to align all the information on a consistent time grid.

Data integrity cannot be guaranteed at all times. Engineers validate specific time phases for analysis, ensuring their integrity. However, due to the extensive volume of flight test data, not all datapoints can be validated. One must consider the data cautiously, acknowledging the possibility of inconsistencies or errors.

Anomalies can range, among others, from sensor detachment to slow drift. While some of these anomalies can be easily detected, most are not trivial to identify.

2.2 . Detail of the 3-steps dataset generation protocol

A set of tools is implemented in order to perform the required specific data transformations. They are written using the Python programming language. In this section, the logic of this 3-step algorithm is detailed: data extraction, data cleaning, and dataset generation. At each step, the data is saved[1] so that it can be used at each level of transformation. While this section presents the overall process of generating a dataset, its constituent tools are detailed in the following sections or in the following Chapters for reduction and representation transformations.

2.2.1 . Data extraction

For a given set of sequences S, the values of an accelerometer A and general parameters P are extracted. They are interpolated to a sampling frequency f_s: general parameters of various sample

[1] Parquet is the chosen storage file format. Column-based, it is designed for efficient data storage and processing.

rates are all interpolated to f_s, while for accelerometers (sampled at 5000 Hz), spectral representations are generated each $1/f_s$ second.

For datasets presented later in the manuscript, f_s is set to 1 or 10 Hz.

- Accelerometers time series $\mathbf{u}$ are first transformed into PSDs using Welch's algorithm (see Section 1.4.1) on a window of size w. Algorithm 1 details this step, where N_i is the number of samples in $\mathbf{u}_i$, and $f_s^{\text{Vib}} = 5000$ Hz its sampling frequency. The result is a set of spectra $\mathbf{s}$ for all sequences in S each $1/f_s$ second.
- The chosen set of general parameters P are interpolated at f_s. The result is a set of time series for all sequences in S each $1/f_s$ second.

Algorithm 1 Vibration data extraction

Require: Set of sequences S, Welch's parameters
  for $i \in S$ **do**
    Read $\mathbf{u}_i$
    for $j \in [0, N_i - w]$ each f_s^{Vib}/f_s **do**
      $\mathbf{s}_{ij} \leftarrow \text{Welch}\left(\mathbf{u}_i[j, j+w]\right)$
    end for
  end for

During the process, assertions about data integrity are made to filter out obvious anomalies, ensuring the data is not empty nor incomplete. Metadata are associated with each datapoint, providing information about the aircraft, flight, flight part[2], and sequence starting time.

2.2.2 . Data cleaning and transformation

The data cleaning and transformation process is separated from the extraction part: this allows raw PSD analysis. For both general parameter and vibration data, null or erroneous signals are removed. Several formatting steps are not described here, such as setting the float type of data or transforming metadata into a multi-index. Anomaly detection is performed on vibration data, see Section 2.3.1 for details.

Data transformation is then performed, allowing to apply (i) spectra separation and (ii) spectra reduction. Spectra separation is an important step that is presented in detail in Section 4.1. To summarize, it allows to separate a spectrum $\mathbf{s}$ into its broadband component $\mathbf{b}$ and its kinematic lines $\mathbf{k}$. Then, spectra reduction can be applied only on broadband spectra $\mathbf{b}$ if the interest is in aerodynamic excitations, or on full PSDs $\mathbf{s}$. This step is detailed in Section 4.2, and multiple reduction methods are proposed. For the prediction task of Chapter 7, RMS levels per frequency band is the preferred reduction method, see Section 4.2.2.

Algorithm 2 details this step for vibration data in the case where both spectra separation and reduction are performed. These transformations are denoted with an operator $T\cdot$, and the reduced version of a broadband spectrum $\mathbf{b}$ is denoted $\widetilde{\mathbf{b}}$.

To summarize, this step allows to apply the following transformations:

[2] Each flight part identifies a *consecutive sequence* during a flight.

Algorithm 2 Spectra separation and reduction transformations applied on vibration data

Require: Transformation parameters
  for $i \in S$ **do**
    Read $\mathbf{s}_i$
    if $\mathbf{s}_i$ contains no anomaly **then**
      $\mathbf{b}_i, \mathbf{k}_i \leftarrow T_{\text{Separation}}(\mathbf{s}_i)$
      $\widetilde{\mathbf{b}}_i \leftarrow T_{\text{Reduction}}(\mathbf{b}_i)$
    end if
  end for

- Spectra separation, see Section 4.1, to separate broadband spectra (a smooth representation) from kinematic lines (a sparse representation)
- Spectra reduction, see Section 4.2, for compression and feature extraction, thus modeling and data exploration through clustering
- Kinematic lines energy computation, see Section 2.3.4, to create a dataset containing the energy of kinematic lines generated by a specific rotating machinery, such as engines.

2.2.3 . Dataset generation

The final step is concatenating transformed data into a tabular format to generate a ready-to-use dataset. In this subsection, the notation $\mathbf{s}$ denotes any spectra representation, regardless of any previous transformations. From the previous time series and successive frequency representations, two matrices are generated: a matrix $\mathbf{X}$ with the values of general parameters sampled as f_s (from the set of parameters P on sequences S), and a matrix $\mathbf{Y}$ with the successive spectra $\mathbf{s}$ through time, each $1/f_s$ second (accelerometer A on sequences S).

Due to the previous cleaning process, both matrices may not contain the same number of data points or rows. To ensure that each datapoint in $\mathbf{X}$ has a counterpart in $\mathbf{Y}$, only the intersection of their indices is kept. Finally, preprocessing transformations specific to machine learning are performed on each matrix. They are described in detail in Section 3.1. In this manuscript, where preprocessing is performed, a standard scaler is applied to $\mathbf{X}$, while a min-max scaler or a standard scaler is applied to $\mathbf{Y}$ after applying a log transformation.

The final dataset is thus made of inputs $\mathbf{X} \in \mathbb{R}^{m \times n}$ and outputs $\mathbf{Y} \in \mathbb{R}^{m \times d}$. The columns of matrix $\mathbf{X}$ contain the values of each general parameter $j \in [1, \ldots, n]$ through time[3]. A row $\mathbf{x}_i^\mathsf{T}$ of $\mathbf{X}$ holds the values of all n parameters at time i. The matrix $\mathbf{Y}$ contains as rows the collection of spectra $\mathbf{y}_i^\mathsf{T} = \mathbf{s}_i^\mathsf{T}$ a each time $i \in [1, \ldots, m]$. Visually, it can be represented as:

$$\mathbf{X} = \begin{bmatrix} - & \mathbf{x}_1^\mathsf{T} & - \\ & \vdots & \\ - & \mathbf{x}_m^\mathsf{T} & - \end{bmatrix}, \ \mathbf{Y} = \begin{bmatrix} - & \mathbf{s}_1^\mathsf{T} & - \\ & \vdots & \\ - & \mathbf{s}_m^\mathsf{T} & - \end{bmatrix}. \tag{2.1}$$

[3]Time does not need to be consecutive.

To be consistent with the rest of the manuscript, the dataset can be written as $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,\dots,m}$, and a standard regression problem (see Chapter 5) would be to predict $\mathbf{y}_i \in \mathcal{Y} \subset \mathbb{R}^d$ from $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^n$.

2.3 . Identification

In this section, the identification tools used for the previous algorithms presented before are presented. They consist of anomaly detection, flight phase identification, configuration identification, and most importantly, kinematic lines identification.

2.3.1 . Anomaly detection

Anomaly detection is a complex task beyond the scope of this thesis. However, outlier energy levels are identified. This allows simple anomalies to be quickly removed from the dataset. The low-energy case is symptomatic of a disconnected sensor, while the high-energy case may be symptomatic of a problem related to the acquisition system.

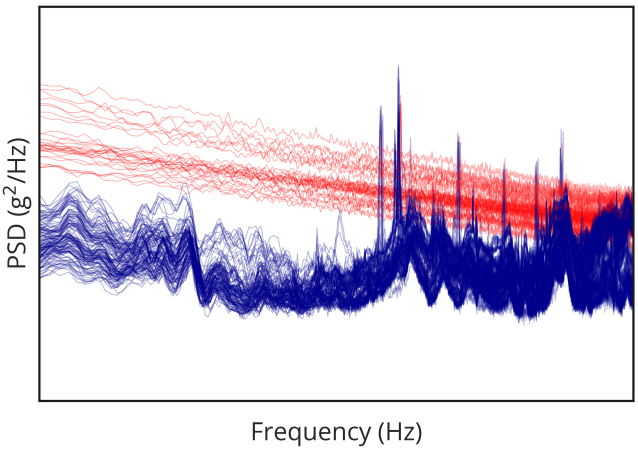


Figure 2.1: Illustration of automatic outlier detection: superposition of several PSDs on a log-log scale, with outliers identified in red.

Outlier identification is illustrated in Figure 2.1, where the anomalies are identified in red. An energy threshold is applied to detect and remove the outliers based on their RMS value on a broad frequency range, filtering out too-low values with λ_1 and too-high values with λ_2:

$$\mathbf{s} \text{ kept if: } \quad \lambda_1 < \mathrm{RMS}(\mathbf{s}\,[f_1, f_2]) < \lambda_2, \tag{2.2}$$

where $[f_1, f_2]$ can either cover the whole frequency range or a specific band. Values of λ_1 and λ_2 are set through domain knowledge.

Sequences identified as faulty can be directly ignored during the extraction, or kept to be analyzed and used to train detection algorithms.

2.3.2 . Flight phase identification

Flight phase recognition, which is a challenging problem [LZV+22], allows for advanced analysis and modeling. For example, it can be used to evaluate how certain maneuvers affect vibration levels or to balance the proportion of flight phases before training a model. It can also be used to select a specific subset of sequences S on which to generate a dataset so that only level flight phases are kept as presented in Section 7.1.

Identifying Level Flight Phases[4] and climbs/descents is implemented here. Other flight phases are not relevant to our needs, although easy to identify, such as turns, or are too difficult to identify and are beyond the scope of this work.

The detection is based on a set of logical rules that should be true on a rolling window of the temporal signal $\mathbf{x}$ of general parameters. Multiple statistics are used, such as the mean $\mu(\mathbf{x})$, the standard deviation $\sigma(\mathbf{u})$, and the maximum $\max(\mathbf{x})$.

For the **stabilized flight phases**, the rules in Table 2.1 must all be true on a 20-second rolling window. The rule on the side slip angle is more strict than for the roll attitude, as a change in the former has more effect on the aerodynamics than the latter, and their amplitudes are not in the same range.

Rule	Implementation
Constant altitude	$\sigma\left(\mathbf{x}\left[t, t+20\text{ s}\right]\right) < 500\text{ ft}$
Constant Mach number	$\sigma\left(\mathbf{x}\left[t, t+20\text{ s}\right]\right) < 0.01$
Constant engine rotational speed	$\sigma\left(\mathbf{x}\left[t, t+20\text{ s}\right]\right) < 0.05\%\text{rpm}$
Centered roll angle	$\sigma\left(\mathbf{x}\left[t, t+20\text{ s}\right]\right) < 1° \;\&\; \left\lvert\max\left(\mathbf{x}\left[t, t+20\text{ s}\right]\right)\right\rvert < 5°$
Centered side slip angle	$\sigma\left(\mathbf{x}\left[t, t+20\text{ s}\right]\right) < 0.1° \;\&\; \left\lvert\max\left(\mathbf{x}\left[t, t+20\text{ s}\right]\right)\right\rvert < 0.5°$

Table 2.1: Stabilized flight phases rules; needs to hold on a 20 seconds window.

The detection of **climbing and descending flight phases** is based on the set of rules presented in Table 2.2, that should hold on a 10-second rolling window, where $\Delta_\tau \mathbf{x}$ denotes a delta of τ seconds on $\mathbf{x}$.

Rule	Implementation
For climbing, threshold of positive altitude variation	$\mu\left(\Delta_{5\text{ s}}\mathbf{x}\left[t, t+10\text{ s}\right]\right) > 50\text{ ft}$
For descending, threshold of negative altitude variation	$\mu\left(\Delta_{5\text{ s}}\mathbf{x}\left[t, t+10\text{ s}\right]\right) < 50\text{ ft}$

Table 2.2: Climbing and descending flight rules; needs to hold on a 10-second window.

As a final note on flight phase detection, it should be noted that the various test equipment can have a significant effect on the vibration environment, so state booleans should be monitored. An extreme example is the use of aileron shakers, where a signal is injected into the ailerons to stress the airframe during certain flight tests. These flight phases should either be removed or accompanied by the appropriate general parameter, such as the intensity of the injected load.

[4] In this manuscript, level flight means *straight-and-level flight*, a specific type of stabilized flight in which the aircraft maintains a constant heading, altitude, and airspeed. It is often referred to as *stabilized flight phases*.

2.3.3 . Aircraft configuration detection

Aircraft configuration refers to the position of its different control surfaces and high lift devices, such as ailerons, flaps, slats, air brakes, and the deployment of landing gear. A change in aircraft configuration changes the aerodynamic flow excitations and induces a different response of the structure. Configuration detection can be used to generate categorical features for machine learning models.

Configurations can be split into two groups: clean and approach configurations. In **clean configuration**, flaps, air brakes, and other surfaces are retracted. In **approach configurations**, multiple of them are deployed to increase lift or reduce the aircraft's speed.

The identification is based on domain rules, where its surface position is associated with a specific configuration. The implementation is similar to the flight phase identification tool (with more rules) and uses conditions on a rolling window.

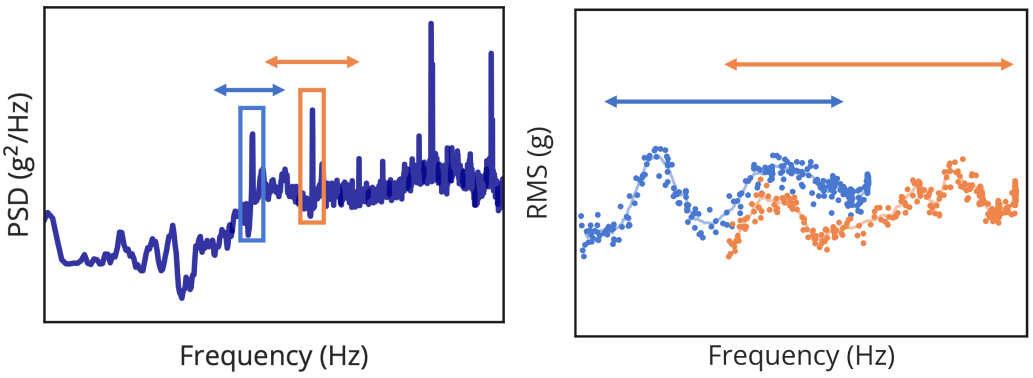
2.3.4 . Kinematic lines frequency, computation of their energy

As presented in Section 1.1.2, kinematic lines are narrowband concentrations of energy by imbalance of rotational machinery, which frequency is linked to the rotational speed.

Since the speed of engines and equipment is known from measurements of general parameters, it is trivial to find the location in frequency of any kinematic line on a PSD. Consider a machine E driven by a turbine N. The frequency f of the harmonic h is given by:

$$f(\mathrm{E}, \mathrm{N}, h) = h \times \frac{100\%\mathrm{rpm}_\mathrm{E}}{60} \times \%\mathrm{rpm}_\mathrm{N}, \tag{2.3}$$

where $100\%\mathrm{rpm}_\mathrm{E}$ is the maximum rotational speed of E, and $\%\mathrm{rpm}_\mathrm{N}$ the rotational speed of the turbine driving the equipment. All rotational speeds are given in revolutions per minute (rpm). Engines kinematic lines can be computed for $\mathrm{E} = \mathrm{N}$, meaning that the turbine N itself is considered. The fundamental frequency is denoted with $h = 1$.



(a) Identification of harmonics frequency (b) RMS levels of harmonics

Figure 2.2: Illustration of the identification of two kinematic lines on a PSD (left), and their computed RMS levels $\boldsymbol{\kappa}$ depending on their frequency (right), both on a log-log scale.

It is thus possible to compute the energy of any given kinematic line. This is done on a PSD $\mathbf{s}$ by

calculating the RMS value, see Eq. (1.12), on a window centered on the kinematic line frequency. For an equipment E, it is defined as:

$$\kappa(\mathrm{E},\mathrm{N},h) \triangleq \mathrm{RMS}\left(\mathbf{s}\left[f(\mathrm{E},\mathrm{N},h)-\frac{\Delta f}{2}, f(\mathrm{E},\mathrm{N},h)+\frac{\Delta f}{2}\right]\right), \tag{2.4}$$

with Δf the window size centered around the kinematic line at frequency f.

A vector containing the values of $\kappa(\mathrm{E},\mathrm{N},h)$ for all frequencies is written $\boldsymbol{\kappa}(\mathrm{E},\mathrm{N},h)$. Later in the manuscript, when E and N are fixed, it is written $\boldsymbol{\kappa}_i^{(h)}$ to specify the value measured at a point i in the structure, or more precisely a given accelerometer channel. In Figure 2.2, the computation of these vectors for multiple harmonics of the engines is illustrated. This is similar to an order analysis.

2.4 . Conclusion on the dataset generation framework

This chapter has presented a framework for generating a vibration environment dataset and allowing a data-driven approach to its analysis and modeling. Its modular implementation allows it to be used in various use cases other than classical analysis. By bypassing data cleaning, a dataset with erroneous measurements can be generated to train anomaly detection algorithms. By choosing one of the vibration data representations described in Chapter 4, its modeling can be approached in different ways.

Several domain knowledge tools are incorporated, allowing anomaly detection, flight phase identification, aircraft configuration identification, and kinematic line frequency identification along with energy computation. In the future, flight phase identification could be extended to specific maneuvers to focus model training. The extraction of kinematic lines frequency and energy is a crucial tool close to order analysis. It can show even greater utility when used with the spectral separation hypothesis presented in Chapter 4.

3 - Unsupervised features extraction

Contents

Machine learning is a branch of artificial intelligence that focuses on developing algorithms and models that can learn from data and make predictions or decisions without being explicitly programmed. It can be divided into several branches, such as unsupervised learning, which is covered in this chapter, and supervised learning, which is covered in Chapter 5. Other branches are not covered in this manuscript, namely semi-supervised learning[1] and reinforcement learning. The former falls between supervised and unsupervised learning by dealing with both labeled and unlabeled data. The latter is a method in which an agent learns to interact with an environment so that its behavior maximizes a reward. Reinforcement learning has applications in physics, particularly in fluid mechanics [GVR+21, VMLH22, RKJ+19].

The goal of unsupervised learning is to find patterns, relationships, and structures in unlabeled data. This can be done to achieve data compression, denoising, and clustering. Since the data does not need to be labeled, it is a powerful tool for large amounts of raw data. However, the quality of its results can sometimes be challenging to interpret due to the lack of labels or *ground truth*.

In the context of the large amount of data generated at the Flight Test Center, machine learning can be useful for various tasks such as data exploration, pattern recognition, anomaly detection, predictive maintenance, and test optimization. It can help improve efficiency, safety, and decision-making. This chapter presents existing tools that can be used on unlabeled test data before applying them in Chapter 4. After an introduction to data preprocessing, methods of data representation and clustering are presented.

Here, and for the rest of this manuscript, a dataset is denoted $\mathcal{D}$, and a parameterized model is denoted $\phi_{\boldsymbol{\theta}}$. The weights of a model are denoted as θ and can be named *parameters*. Any parameter that is not learned is called a *hyperparameter*. Specifically, in the context of unsupervised learning, the dataset consists of m unlabeled data points $\mathcal{D} = \{\mathbf{x}_i\}_{i=1\dots m}$ such that $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^n$.

3.1 . Data preprocessing

[1] Semi-supervised learning is sometimes called weak supervision.

Before introducing any unsupervised learning method, basic transformations need to be explained in detail. They allow certain assumptions to be made[2], and generally facilitate the optimization process. As such, this section is relevant to both unsupervised and supervised learning, see Chapter 5.

Standardization can be applied to get zero mean and unit variance data:

$$\mathbf{x}_{\text{Transformed}} \leftarrow \frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}. \tag{3.1}$$

When the distribution of the data is known to be uniform or when outliers are not a concern, a min-max normalization can be applied. It provides robustness to very small standard deviations of features and preserves zero entries in sparse data:

$$\mathbf{x}_{\text{Transformed}} \leftarrow \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}. \tag{3.2}$$

For skewed data distributions, taking the logarithm of the data helps to normalize the distribution and make it more symmetric:

$$\mathbf{x}_{\text{Transformed}} \leftarrow \log_{10}(\mathbf{x}). \tag{3.3}$$

To apply transformations that rely on statistics, such as standardization or min-max normalization on a matrix $\mathbf{X}$, it is important to be careful of the dimension on which to compute the statistics. Considering m samples and n features, if $\mathbf{X} \in \mathbb{R}^{m \times n}$, the statistics should be taken across samples, thus for each column. The resulting mean vector $\boldsymbol{\mu}(\mathbf{X}) \in \mathbb{R}^n$ is a row vector. An exception in this manuscript is in Section 3.2.1 on Principal Component Analysis, where samples and features dimensions are interchanged.

Finally, encodings are worth mentioning as they allow to transform categorical data into discrete numerical representations using techniques such as one-hot encoding (one binary column is created for each category) or ordinal encoding (one value is associated with each category).

3.2 . Data representation

The choice of representation is crucial to appropriately understand data, as presented in Section 1.4. In addition to highlighting information, a suitable representation can be used for dimensionality reduction, allowing for compression and denoising.

The choice of dimensionality reduction technique depends on the nature of the data at hand, the computational power at disposal, and the end goal. Several methods relevant to the rest of the manuscript are presented here.

3.2.1 . Principal Component Analysis

PCA (Principal Component Analysis) [Pea01][Hot33] allows to reduce the dimensionality of a multivariate dataset by decomposing it into a set of orthogonal and centered components. The components, which are linear combinations of the original features, can be ordered according to their contribution to the total variance of the signal. The number of components to keep can be adjusted

[2] A significant amount of machine learning is based on a Gaussian assumption.

depending on the desired trade-off between the compression ratio and the proportion of variance explained.

To apply PCA to a dataset, an SVD (Singular Value Decomposition) can be performed[3] on standardized data, see Eq. (3.1). It provides an optimal low-rank approximation [EY36], with hierarchically ordered orthogonal components. This means that any r-rank approximation can be obtained by keeping only the leading r singular values and vector, see Eq. (3.5).

Let $\mathbf{X} \in \mathbb{C}^{n \times m}$ be a *centered* matrix, where n is the number of features and m is the number of samples, assuming $n \leq m$. Note that PCA conventions are used in this section, but in the rest of the manuscript, samples are ordered as rows instead of columns like here. Its unique SVD decomposition is:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \tag{3.4}$$

with $\mathbf{U} \in \mathbb{C}^{n \times n}$ a unitary matrix[4] containing the left singular vectors of $\mathbf{X}$, $\mathbf{\Sigma} \in \mathbb{R}_+^{n \times m}$ a diagonal matrix containing the singular values of $\mathbf{X}$, and $\mathbf{V}^* \in \mathbb{C}^{m \times m}$ the conjugate transpose unitary matrix containing the right singular vectors of $\mathbf{X}$. The columns of $\mathbf{U}$ provide an orthonormal basis for the column space of $\mathbf{X}$, and the columns of $\mathbf{V}$ provide an orthonormal basis for the row space of $\mathbf{X}$.

To reduce the data dimensionality to $r < n$, a truncated SVD can be used:

$$\mathbf{X} \approx \widetilde{\mathbf{X}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^*, \tag{3.5}$$

where $\widetilde{\mathbf{X}}$ is the optimal rank-r approximation of $\mathbf{X}$, $\widetilde{\mathbf{U}} \in \mathbb{C}^{n \times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$, and $\widetilde{\mathbf{V}}^* \in \mathbb{C}^{r \times m}$.

In the ℓ^2 sense, the truncated SVD approximation $\widetilde{\mathbf{X}}$ is the best approximation for $\mathbf{X}$. The principal components can be obtained from the right singular vectors $\mathbf{V}$ (or $\widetilde{\mathbf{V}}$), which represent the directions of maximum variance in the data. Indeed, the first principal component $\mathbf{u}_1$ is given as:

$$\mathbf{u}_1 = \underset{\|\mathbf{u}_1\|=1}{\arg\max}\, \mathbf{u}_1^* \mathbf{X}^* \mathbf{X} \mathbf{u}_1, \tag{3.6}$$

which also satisfies:

$$\mathbf{u}_1 = \arg\max \frac{\mathbf{u}_1^* \mathbf{X}^* \mathbf{X} \mathbf{u}_1}{\mathbf{u}_1^* \mathbf{u}_1}, \tag{3.7}$$

which is a Rayleigh quotient[5]. Considering a truncated SVD, the projected version of $\mathbf{X}$ called the score matrix $\mathbf{T}$ can thus be written:

$$\widetilde{\mathbf{T}} = \mathbf{X}\widetilde{\mathbf{V}}. \tag{3.8}$$

The covariance matrix is:

[3] SVD applied on standardized data is the standard way to perform PCA because of the efficiency of SVD algorithms, but it is not the only one.

[4] A square matrix $\mathbf{U}$ is unitary if $\mathbf{U}^*\mathbf{U} = \mathbf{U}\mathbf{U}^* = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. It is analogous to an orthogonal matrix for real numbers.

[5] The maximum possible value of a Rayleigh quotient is the largest eigenvalue of the matrix.

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}^*\mathbf{X}. \tag{3.9}$$

The eigenvalues of the covariance matrix $\mathbf{C}$ can be calculated as the squared singular values: $\lambda_i = \sigma_i^2/(n-1)$, where σ_i is the i-th singular value. The explained variance ratio (EVR) for each principal component can be computed as the ratio of the eigenvalue to the sum of all eigenvalues:

$$\mathrm{EVR} = \frac{\lambda_i}{\sum_{j=i}^{n}\lambda_j}. \tag{3.10}$$

3.2.2 . Non-negative Matrix Factorization

NMF (Non-negative Matrix Factorization) [LS99] is a dimensionality reduction technique that decomposes a non-negative matrix into two non-negative matrices. For instance, it is used in astrophysics where data is often non-negative [BR07][RPZ+18].

Let $\mathbf{X} \in \mathbb{R}_+^{m\times n}$ be a non-negative matrix, where m is the number of samples, and n is the number of features. Note that this is the convention used in the rest of the manuscript, with the exception of the previous section regarding PCA. The NMF decomposition of $\mathbf{X}$ is:

$$\mathbf{X} \approx \mathbf{W}\mathbf{H}, \tag{3.11}$$

with $\mathbf{W} \in \mathbb{R}_+^{m\times k}$ containing the basis vectors or components, and $\mathbf{H} \in \mathbb{R}_+^{k\times n}$ containing the coefficients or weights that combine the basis vectors to reconstruct the original data. Whenever $k < \min(n, m)$, it is not an exact decomposition[6].

$\mathbf{W}$ and $\mathbf{H}$ are obtained by iteratively minimizing the reconstruction error, such that:

$$\min_{\mathbf{W},\mathbf{H}} \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_{\mathrm{F}}^2, \quad \text{s.t. } \mathbf{W} \geq 0, \mathbf{H} \geq 0, \tag{3.12}$$

where $\|\cdot\|_{\mathrm{F}}^2$ is a squared Frobenius norm, which can be seen as a generalization of the Euclidean norm to matrices. The Frobenius norm $\|\cdot\|_{\mathrm{F}}$ is defined as:

$$\|\mathbf{A}\|_{\mathrm{F}} \triangleq \sqrt{\sum_{i,j} A_{ij}^2} = \sqrt{\mathrm{tr}(\mathbf{A}\mathbf{A}^*)}. \tag{3.13}$$

Note that in Eq. (3.12), no regularization terms are included for simplicity, but they are generally used to ensure the values of $\mathbf{W}$ and $\mathbf{H}$ remain low. Moreover, a squared Frobenius norm is used, but could be replaced by another beta-divergence such as the Kullback-Leibler divergence or the Itakura-Saito divergence.

The optimization process can be done using various solvers, such as Coordinate Descend [CP09] or Multiplicative Update [FI11].

[6] Although it should be mentioned for exhaustiveness that an exact decomposition can be attained in polynomial time for some cases.

Considering the fractional residual variance[7], NMF tends to overfit[8] less that PCA [RPZ+18]. NMF decomposition tends to produce local representations of the dataset, which can be useful to describe specific phenomena localized in frequency on a spectrum.

3.2.3 . Autoencoders

Autoencoders are a type of ANN (Artificial Neural Network, see details in Section 5.2.5) that can be used for dimensionality reduction [RHW86]. Briefly, an ANN is a model $\phi_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta} \in \Theta$, mapping an input space $\mathcal{X}$ to an output space $\mathcal{Y}$ such that $\phi_{\boldsymbol{\theta}} : \mathcal{X} \times \Theta \to \mathcal{Y}$. In this section, a parameterized model is simply written ϕ. Autoencoders are especially useful for nonlinear encoding and aim to learn a compressed representation of the input data. They can be written as:

$$\phi(\mathbf{x}) = \phi_{\text{Decoder}} \circ \phi_{\text{Encoder}}(\mathbf{x}) = \hat{\mathbf{x}}, \tag{3.14}$$

with $\hat{\mathbf{x}} \approx \mathbf{x} \in \mathbb{R}^n$ a reconstructed value of $\mathbf{x}$. The model first encodes the data into a lower dimensional space such that:

$$\phi_{\text{Encoder}} : \mathbb{R}^n \times \Theta_{\text{Encoder}} \to \mathbb{R}^k, \tag{3.15}$$

where $k < n$ (in which case the autoencoder is called *undercomplete*). Then, the model decodes the reduced data and maps it back to the original space:

$$\phi_{\text{Decoder}} : \mathbb{R}^k \times \Theta_{\text{Decoder}} \to \mathbb{R}^n. \tag{3.16}$$

To perform this transformation, a cost function evaluating the dissimilarity between the original and reconstructed data is minimized, generally by using gradient descent, see Section 5.3.2. This is done over a complete dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1,\dots,m}$, and a metric such as the Mean Squared Error can be used as in Eq. (3.17). More details on cost functions are given in Section 5.3.1. When the complete mapping is learned, the encoder and decoder can be used separately to reduce and reconstruct data, as would be done using PCA or NMF.

$$C(\boldsymbol{\theta}_{\text{Encoder}}, \boldsymbol{\theta}_{\text{Decoder}}) = \frac{1}{n}\frac{1}{m}\sum_{i=1}^{m} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 . \tag{3.17}$$

An illustration of an autoencoder is given in Figure 3.1. The middle layer of dimension k is called the latent layer, or the bottleneck layer. The name Deep Autoencoder can be used when several hidden layers[9] are used.

While the autoencoder is a relatively simple architecture, several alternatives and variants have been proposed, such as regularized autoencoders [AB14][BYAV13], using regularization to improve sparsity, and variational autoencoders [KW22], introducing elements from variational Bayesian methods.

[7] It is the fraction of variance of the variable to predict which cannot be explained by the features.

[8] Overfitting, see Figure 5.7, is the tendency of an algorithm to fit the data too much, thus reducing its generalization ability. In the case of dimensionality reduction, overfitting can be seen as learning the noise.

[9] A hidden layer is any layer other than the input, output, and latent layer.

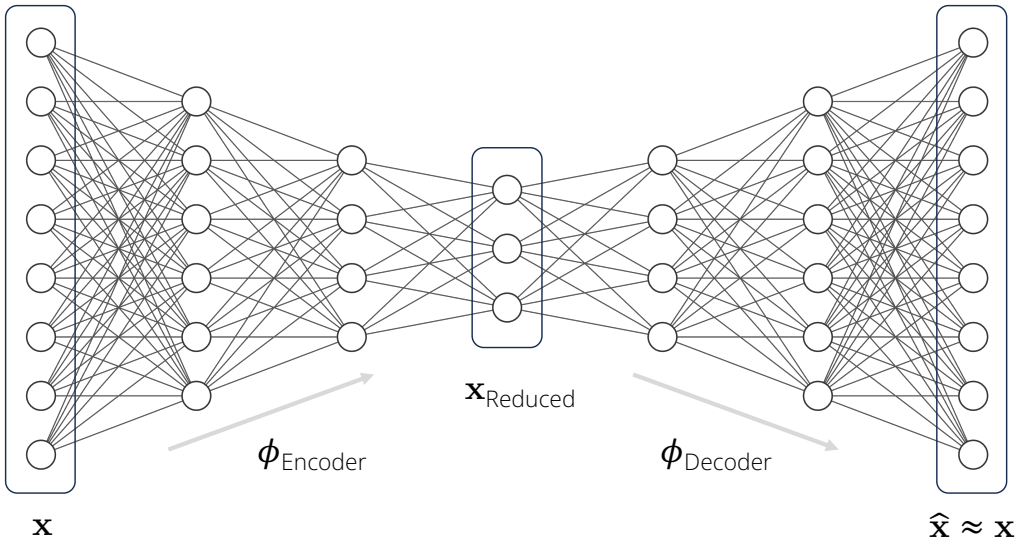


Figure 3.1: Autoencoder architecture example, where a representation is learned to reduce 8-dimensional data to 3 dimensions. Figure partly generated using [LeN19].

3.3 . Clustering

Clustering is a technique used to group similar data points together, which aims to find patterns or structures in data without any predefined labels. In Figure 3.2, an illustration is given where a 2D projection of unlabeled data is separated into 4 clusters.

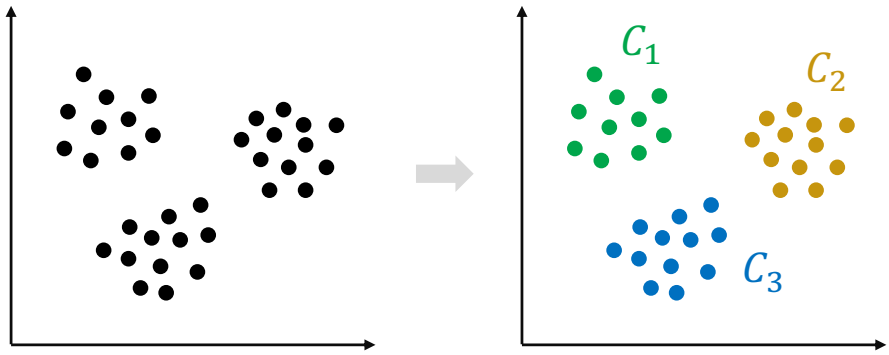


Figure 3.2: Illustration of clustering.

3.3.1 . k-means clustering

One popular clustering algorithm is the k-means clustering algorithm [Llo82]. A dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i,\dots,m}$ is divided into k disjoint clusters C, each described by the mean $\boldsymbol{\mu}_j$ of the samples in the cluster.

The objective of the k-means algorithm is to minimize the sum of squared distances between each

data point and its assigned centroid. This is known as the inertia criterion, or the within-cluster sum of squares:

$$\text{Inertia} \triangleq \sum_{i=1}^{n} \min_{\boldsymbol{\mu}_j \in C} \left(\|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2^2 \right). \tag{3.18}$$

The algorithm works as follows:

Algorithm 3 k-means

Require: $\mathcal{D}$, k
Initialize cluster centroids $\mu_1, \ldots, \mu_k$ randomly.
repeat
Assign each data point to the nearest centroid.
Recalculate the centroids by taking the mean of all data points assigned to each centroid.
until Convergence criterion

The optimal number of clusters k must be determined, which can be a complex problem and has only empirical solutions. One of them is the elbow method. It involves plotting the inertia against the number of clusters and looking for the "elbow" point, which is the inflection point in the plot. The idea is to choose the value of k at the elbow point as it represents a good balance between minimizing inertia and avoiding overfitting.

The initialization of the centroids can strongly influence the performance of the algorithm: while the basic approach is to assign random data points as centroids, other initialization techniques have been proposed, such as k-means++ [AV07], with the idea of spreading the k initial cluster centers.

Finally, it should be mentioned that k-means clustering is usually applied to a dataset where the dimension has been reduced using a previously introduced method such as PCA or an autoencoder. This is done to improve the clustering performance: the algorithm has fewer features to consider, with less noise.

3.3.2 . Other clustering methods

While k-means clustering may be the most straightforward clustering algorithm, other algorithms exist, based on ideas from various fields.

A **self-organizing map** [Koh82], also known as Kohonen map, is a bio-inspired [Tur52, von73] unsupervised learning method based on ANNs (see Section 5.2.5). Using competitive training[10] [RMU86], it has the advantage of preserving the topological properties of the input space. A fuzzy implementation of k-means has been proposed: **fuzzy C-means** [GG89]. Each data point can be assigned to more than one cluster by introducing a *membership function*, which can take values between 0 and 1. **Affinity propagation** [FD07] is based on message-passing, and chooses the number of clusters based on the data provided. Its main drawback is its complexity, rendering it useful only for small and medium-sized datasets. **Hierarchical clustering** is based on successive merging or splitting clusters, and

[10] Different from the nowadays commonly used backpropagation with gradient descent, competitive training puts neurons in competition to try to match input data.

uses tree (or *dendrogram*) representations. It can use an *agglomerative* strategy (bottom-up) or a *divisive* strategy (top-down). **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** [EKX96][SSE+17] is based on the idea that clusters are high-density regions separated by low-density regions. This allows to find clusters of any shape, whereas k-means tends to favor convex shapes. Instead of the conventional centroids of k-means, DBSCAN uses *core samples*, which are samples in areas of high density. **Spectral clustering** techniques such as NCuts [SM00] use the eigenvalues and eigenvectors of a similarity matrix to perform dimensionality reduction. Then, they apply a clustering algorithm, for example k-means, on the reduced data. It is computationally efficient in the case of a sparse affinity matrix. **Gaussian mixture models** assume information about the covariance structure of the data. They can be seen as a generalization of k-means clustering, where data points are seen as being generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Several flavors exists, from Gaussian mixture to Variational Bayesian Gaussian mixture.

4 - Reduced representations and insights in aircraft vibration environment

Contents

The question of representation is at the heart of any data-driven methodology. This chapter details the application of the data representation methods presented in Chapter 2.

First, spectra separation is introduced in Section 4.1, with a presentation of the hypothesis, an evaluation of it through experimental data, and an algorithmic implementation. Second, data reduction techniques are applied and compared by their interpretability and compression factor in Section 4.2. Third and finally, an exploration of the transformed data is performed in Section 4.3. Also based on tools introduced in Chapter 2, such as clustering, it allows to gain insights into the vibration environment of an aircraft. Its results will be used later, especially in the modeling chapters for feature selection.

4.1 . Spectra separation

Predicting a complete vibration spectrum up to 2000 Hz, with its variety of sources and phenomena, is a task that requires complex models that can be difficult to interpret[1].

Here, a method is presented to separate a spectrum into its broadband and kinematic lines components. First, the hypothesis underlying this method is presented and verified. Then, the implementation is described in detail.

[1] Interpretability in machine learning is a broad term with multiple levels of definition. Here, interpretability can be related to the degree of nonlinearity and the number of weights in a model. This is discussed in detail in Section 5.5.

4.1.1 . Background spectrum hypothesis

A spectrum can be seen as the result of a set of excitations and the structure's response. A strong assumption is made. During stationary flight phases such as level-flight[2], the measured response of the structure at any point can be decomposed into:

- A **broadband spectrum** $\mathbf{b}$ which is a smooth representation consisting of a broadband constant shape called the background spectrum $\boldsymbol{\psi}$ multiplied in the frequency domain by a broadband amplitude factor α. The **background spectrum** $\boldsymbol{\psi}$ depends primarily on the position in the aircraft and the measurement axis. It is also affected by the aircraft configuration, which affects the aerodynamic flow[3]. The **broadband amplitude factor** α also depends on position and axis but is mainly affected by a subset of general parameters related to the flight point (mainly the dynamic pressure[4]).
- A **collection of kinematic lines** $\mathbf{k}$ generated by the various rotating machines. For each machine E driven by a turbine N and harmonic h, a kinematic line measured anywhere in the aircraft consists of an excitation $\boldsymbol{\delta}$ and an attenuation factor $\boldsymbol{\eta}$ analogous to a transfer function between the source and the measurement point. The **excitation** $\boldsymbol{\delta}$ depends only on the rotation speed of N and is generated at E. The **attenuation factor** $\boldsymbol{\eta}$ allows to transport this excitation from E to the measurement point, taking into account the reduction of the vibration levels induced by the propagation through the structure, and thus depends only on the structure.

This assumption holds for the mid- and high-frequency ranges (as modal behavior is prevalent in the low-frequency range) and for stationary measurements. As no noise model is used, it is assumed to be contained either in $\mathbf{b}$ or in $\mathbf{k}$. Under this hypothesis, a spectrum $\mathbf{s}$ can be written as:

$$\begin{aligned} \mathbf{s} &= \mathbf{b} + \mathbf{k} \\ &= \alpha\boldsymbol{\psi} + \sum_{\mathrm{E},\mathrm{N},h} \boldsymbol{\eta} \odot \boldsymbol{\delta} \end{aligned} \tag{4.1}$$

where $\odot$ denotes the Hadamard product or element-wise product. Broadband levels generated by the rotating machinery are taken into account by α. Note the difference in modeling between $\mathbf{b}$ and $\mathbf{k}$. For $\mathbf{k}$, a collection of excitations is identified and transferred from the sources to the measurement point. For $\mathbf{b}$, the complexity of the aerodynamic sources makes it such that no separation is performed, and the result at any point is a broadband shape shifted in amplitude. This modeling is close to the one presented in [Maz05] about background estimation of spectroscopy signals applied to chemistry, although the level of background smoothness is not the same.

One way to verify this background hypothesis is to consider only the broadband spectra at various dynamic pressures, thus various flight domain points. All other parameters being equal and the aircraft being stationary, only a broadband shift should be observed. As presented in Figure 4.1, this is indeed what is observed from 300 Hz, where broadband spectra at several dynamic pressures are presented for a given accelerometer.

[2] Note that, as mentioned earlier, level-flight or stabilized flight phase here refers to straight-and-level flight.
[3] E.g., flap extension induces low-frequency vibrations.
[4] The dynamic pressure is defined as $p_{\text{dyn}} = 1/2\rho v^2$ with ρ the air mass density and v the relative air speed.

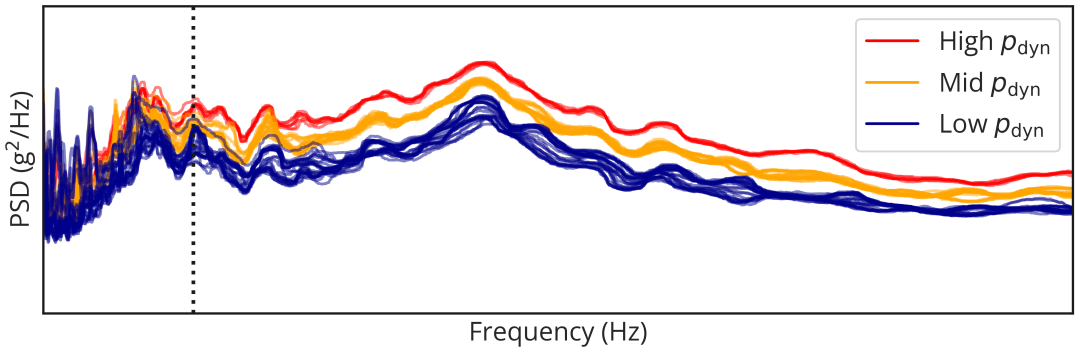


Figure 4.1: Broadband levels amplitude variations depending on the dynamic pressure for a given accelerometer, during straight-and-level flight, with control surfaces at their zero-position. A log scale on the y-axis is used.

Another representation of this is to compute the RMS values on the 300-2000 Hz frequency range with the same setup: stationary aircraft, variations of the dynamic pressure only. Results are presented in Figure 4.2 for different accelerometers. The expected linear dependency is observed. Moreover, the effect of α is showcased: along the x-axis, dynamic pressure variations induce a variation of the broadband levels; the slope of this factor depends on the accelerometer and thus on the position and axis measured.

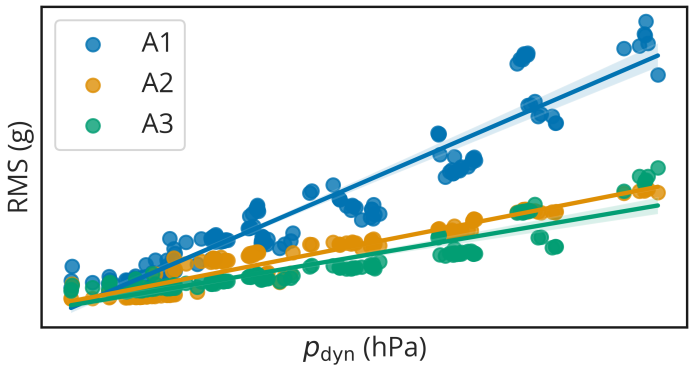


Figure 4.2: Broadband levels amplitude variations in the 300-2000 Hz frequency range depending on the dynamic pressure for 3 accelerometers, during straight-and-level flight, in clean configuration. A1 is an accelerometer close to the engines, while A2 and A3 are far from the engines.

4.1.2 . Broadband estimation algorithm

Following the hypothesis presented previously, a tailored method to separate $\mathbf{b}$ and $\mathbf{k}$ is implemented to answer this non-trivial problem. Figure 4.3 provides an illustration of the separation.

First attempts were based on a rolling ball algorithm [Ste83], mostly used in biomedical image processing, which estimates the background intensity of a grayscale image. In *layman's terms*, an image can be thought of as a surface on which a ball is placed. The background estimation is performed

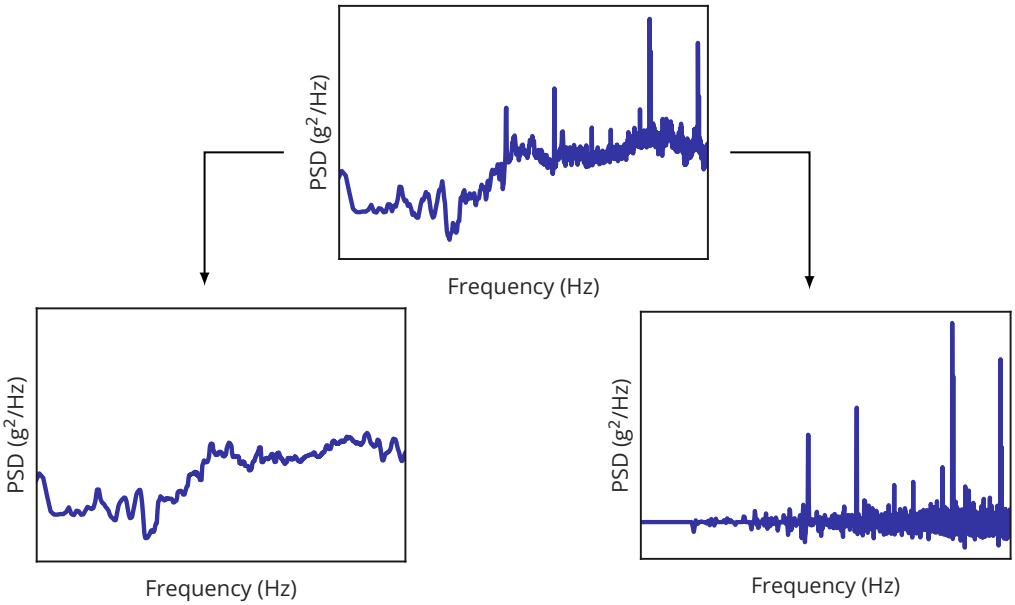


Figure 4.3: Example of a spectrum excitation separation. Original spectrum $\mathbf{s}$ (top), broadband spectrum $\mathbf{b}$ (bottom left), kinematic lines $\mathbf{k}$ and noise ε (bottom right).

by considering the elevation of the ball when rolling along the surface. If the ball has a large diameter, the resulting estimated background is very smooth, and surface "roughness" is mostly ignored, whereas a small ball captures more local irregularities. This method requires careful selection of the ball shape and size (referred to as the *kernel*), and can be computationally intensive.

Other background estimation methods exist, for example based on polynomial approximation [Maz22]. Made for chemical spectra, this method assumes that background spectra are modeled by fourth- or fifth-order polynomial functions. The level of smoothness obtained is too high for our application, resulting in information loss.

The chosen implementation is based on a moving median with a variable window size. A median is used instead of a mean to be robust to "outliers", which in this case are the kinematic lines. The variable window size allows low-frequency information to be preserved while smoothing out mid and high frequencies. As such, it can be seen as close to the rolling ball algorithm on a log-transformed surface. This method works best with stabilized flight phases, as the kinematic lines are assumed to be relatively thin.

The background estimation of a a spectrum $\mathbf{s}$ up to a factor α is given at any frequency by:

$$b\left(\frac{f_1+f_2}{2}\right) = \text{median}\left(\mathbf{s}\left[f_1, f_2\right]\right), \tag{4.2}$$

where the interval $[f_1, f_2]$ corresponds to the window size. It is parameterized depending on the frequency, in Table 4.1. The window size needs to be carefully selected, and an edge effect is introduced by the rolling window but has no negative effect in this case.

Applying this algorithm thus gives $\mathbf{b}$. Subtracting $\mathbf{b}$ from $\mathbf{s}$ gives $\mathbf{k} + \varepsilon$, where ε is the noise. It is

modeled here as a Gaussian noise $\mathcal{N}(0, \sigma)$ of variance σ.

Parameter	Value
Window size from 1 Hz to 20 Hz	β Hz
Window size at 20 Hz	$2 \times \beta$ Hz
Window size at 2000 Hz	$200 \times \beta$ Hz
Window size evolution from 20 Hz to 2000 Hz	Linear

Table 4.1: Smoothing parameters, with β being the spectral resolution, see Eq. (1.6).

4.2 . Vibration spectra low-order representation

This section details the application of reduced representations on data extracted from the flight test database. First, the transition from temporal to frequency domain is presented using a PSD. Second, multiple reductions are applied to the frequency domain representation and compared against each other.

4.2.1 . From temporal to frequency domain

As presented in Section 1.4.1, a PSD allows one to describe the frequency distribution of a random signal's content. Welch's method is applied using the implementation proposed by the Python library SciPy [VGO+20]. A validation against in-house tools used at Dassault Aviation has been performed, showing no differences.

The parameters of PSD calculation using Welch's method are given in Table 4.2.

Parameter	Default value
FFT windows size	4096 or 8192 points
FFT windows overlap	90%
FFT windows type	Hanning

Table 4.2: PSD parameters and proposed default values for this manuscript.

FFT windows sizes are powers of 2 so as to perform the Discrete Fourier Transform using the FFT algorithm in $O(N \log N)$ instead of $O(N^2)$, with N the length of the temporal signal. The default window size on which PSDs are calculated is set to 20 or 30 seconds by default, but is sometimes reduced in the manuscript depending on the application, as for the prediction of vibration spectra during unsteady phases, see Section 7.2.

A 30-second signal sampled at 5000 Hz takes up 150,000 time samples. With the given parameter, a PSD representation of the same signal would have a frequency resolution of $\beta = 4096/5000 = 0.8192$. This means that by considering only the 10-2000 Hz range, only 2430 frequency samples are required to describe this signal, which equals to a compression rate of 1.62%.

The total power of the signal is invariant to the parameters used to compute a PSD: they affect the distribution of power across different frequency bins in the PSD estimate. This means that any given

"spike" can become "narrow and tall" or "broad and short", the RMS value of the total signal remains constant.

4.2.2 . RMS per frequency band

After transforming temporal signals into a frequency representation, several reduction methods can be applied. The first is to compute RMS levels per frequency band. This approach is physically interpretable and adapted to industrial needs. With the exception of some data exploration steps, most of the manuscript uses this representation.

A reduced spectrum is denoted $\widetilde{\mathbf{s}}$, and contains the energy of each band b covering the range $[f_1, f_2]$, such that:

$$\widetilde{s}(b) = \mathrm{RMS}(\mathbf{s}[f_1, f_2]) \approx \sqrt{\beta \sum_{f \in [f_1, f_2]} s(f)}, \tag{4.3}$$

with the RMS being defined in Eq. (1.12) and β the spectral resolution, see Eq. (1.6). The set of bands is close to third-octave bands, which are defined by a center frequency for bands 1 to 32 as:

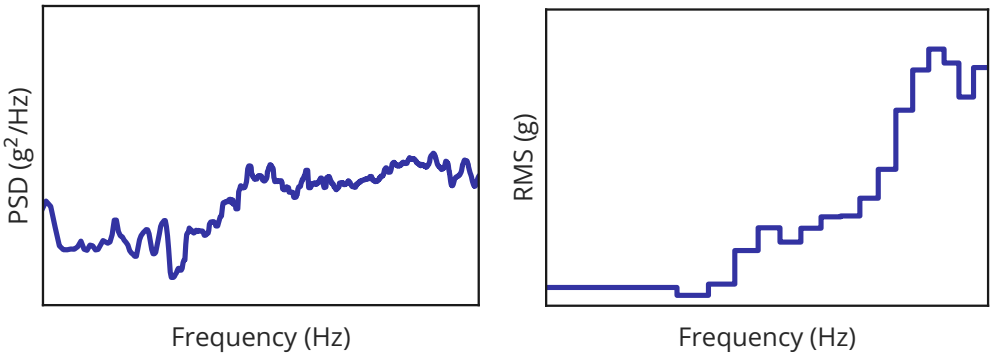
$$f_{\text{center}} \triangleq 10^{0.1n}, \quad \forall\, n \in [12, 43], \tag{4.4}$$

and an upper and lower frequency boundary:

$$f_{\text{upper}} \triangleq f_{\text{center}} \times 10^{0.05}, \quad f_{\text{lower}} \triangleq \frac{f_{\text{center}}}{10^{0.05}}. \tag{4.5}$$

Here, bands are adjusted so they fit the modal behavior of the aircraft, which is understood by performing a GVT, see Section 1.2.3. In total, 17 bands are used, covering the 2-2000 Hz frequency range.

Applying this reduction on broadband spectra $\mathbf{b}$ instead of PSDs $\mathbf{s}$ allows to reduce the variance, as the phenomenon of kinematic lines "jumping" from one band to another is suppressed.



(a) Broadband spectrum (b) RMS levels per band

Figure 4.4: Example of a broadband spectrum (left) reduced into RMS levels per band (right).

In Figure 4.4, the reduction of a broadband spectrum into RMS levels per frequency band is showcased. The choice of bands induces a bias, as variance increases with bands narrowing/refinement.

However, this effect is considered to be non-significant, as presented in Section 7.1.7 where performances of different models are compared against the discretization level.

4.2.3 . Principal Component Analysis

Another reduction method is to apply PCA, which allows the description of a spectrum through a linear combination of its first principal components, see Section 3.2.1. It is applied on broadband spectra, as capturing the very localized energy of kinematic lines would infer with the decomposition[5].

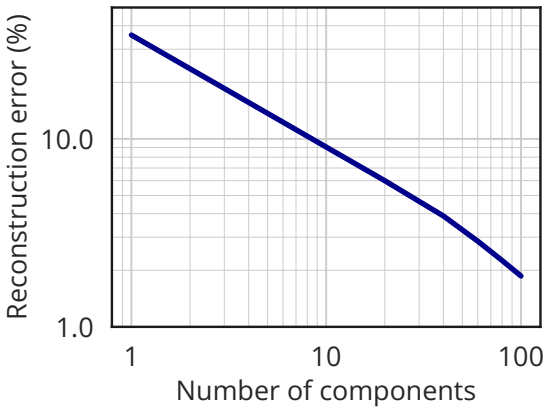


Figure 4.5: Reconstruction error of PCA depending on the number of components used, log-log scale.

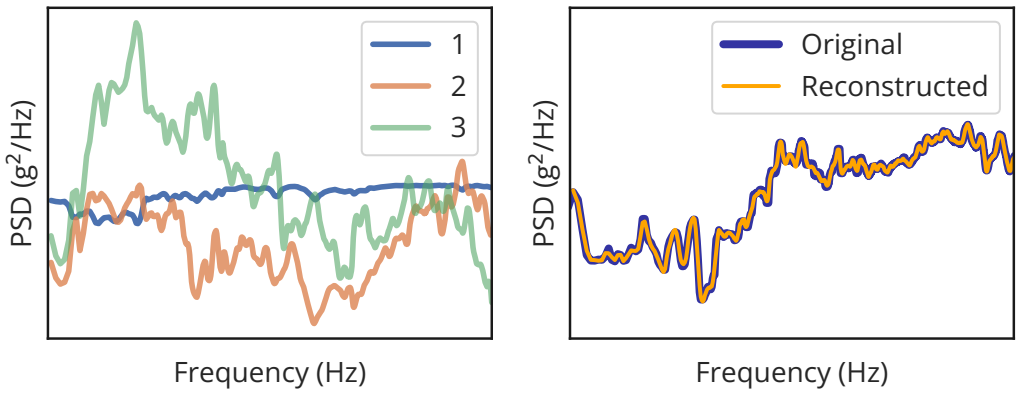
Before performing the transformation, data is log-transformed, see Eq. (3.3), and centered using a standard scaler applied on each column/frequency, see Eq. (3.1). Also, only the relevant frequency range up to 2000 Hz is considered to compute the components: as accelerometers are sampled at 5000 Hz, the resulting spectra have information up to 2500 Hz, but the 2000-2500 Hz samples must be removed.

In Figure 4.5, the reconstruction error decreases with a power relationship with respect to the number of components. The metric used, where a 100% error relates to a 1-order-of-magnitude error, is employed multiple times in the manuscript and introduced in Eq. (5.34). With 100 components, 99.9% of the variance is explained, with a 1.9% reconstruction error, while reducing the data to 4.1% of its original size. The Scikit-learn [PVG+11] implementation is used.

In Figure 4.6a, the first 3 principal components are presented. Component 1 is almost flat and can be associated with the broadband amplitude factor α defined in Section 4.1.1, while other components are more localized in frequency. In Figure 4.6b, the spectrum reconstructed with the first 100 components is superposed to the original spectrum, showing an almost perfect reconstruction. As a side note, it is possible to reconstruct spectra of accelerometers that are not used in the dataset to compute the principal components.

An alternative to applying PCA on broadband spectra is to apply it on RMS spectra. The PCA is thus performed on $\widetilde{\mathbf{b}} \in \mathbb{R}^{17}$. While it could be interesting for extreme compressions, the increase

[5]A solution to avoid the separation between $\mathbf{b}$ and $\mathbf{k}$ would be to learn a dictionary that already contains line components at each frequency.



(a) First 3 principal components (b) Reconstructed broadband spectrum

Figure 4.6: PCA applied on a broadband dataset, using 100 components, log-log scale.

in reconstruction error is not worth the low compression rate benefit compared to only using RMS reduction.

4.2.4 . Non-negative Matrix Factorization

NMF is an alternative decomposition to PCA, taking advantage of the non-negative nature of data, see Section 3.2.2. While originally non-negative, data is log-transformed and scaled using a min-max scaler, see Eq. (3.2), during dataset generation to ease the optimization process.

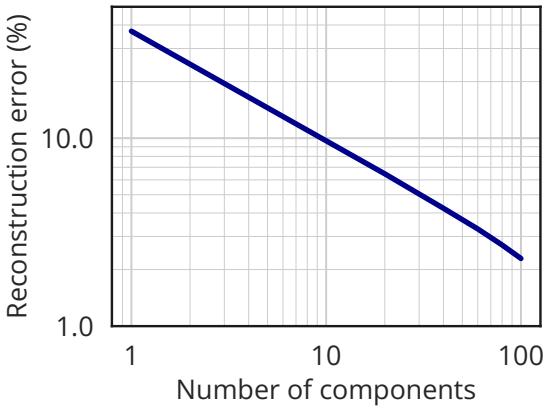
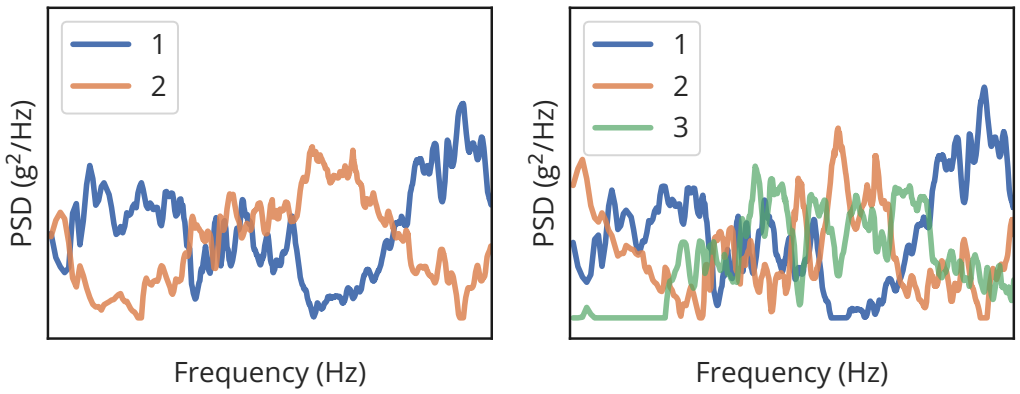


Figure 4.7: Reconstruction error of NMF depending on the number of components used, log-log scale.

In Figure 4.7, NMF is applied on the same broadband dataset as previously in Figure 4.5. The reconstruction error as a function of the number of components is similar to PCA. As for PCA, the Scikit-learn [PVG+11] implementation is used. While it provides interesting components, this method is computationally intensive, as Eq. (3.12) has to be minimized.

An interesting property of NMF is that the non-negative components allow to describe phenomena localized in frequency. Another difference with PCA is that components learned for a decomposition



(a) NMF with 2 components (b) NMF with 3 components

Figure 4.8: NMF components for 2 (left) and 3 (right) components, with the same y-scale.

using 2 modes are not the same as the ones learned for a decomposition using 3 modes, as presented in Figure 4.8.

As the reconstruction error depending on the number of components evolves similarly to PCA, 100 components are used for the reconstruction presented in Figure 4.9.

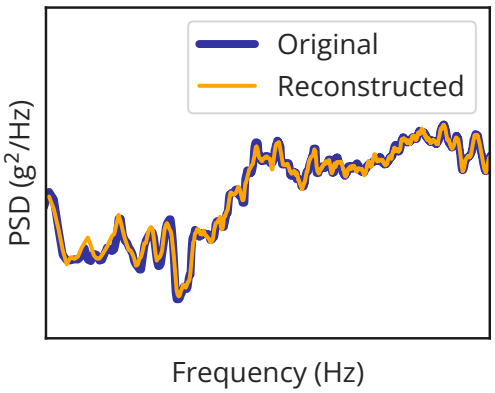


Figure 4.9: Reconstructed broadband spectrum using NMF with 100 components, log-log scale.

Finally, using NMF on RMS levels produces the same kind of results as PCA for greater complexity.

4.2.5 . Autoencoder

A compression algorithm based on an autoencoder has been cursorily evaluated. It is presented here for exhaustiveness, as other representations are sufficient. The significant increase in computation time and interpretation complexity of autoencoders were deemed not worth the effort.

The model's architecture is a feedforward neural network with 2 encoding layers and 2 decoding layers. The number of neurons per layer is given below:

$$1600 \to 400 \to 100 \to 400 \to 1600$$

Each layer is followed by a dropout layer[6] set to 0.1, with the exception of the latent layer. For the encoder layers, ReLU activations are used, while tanh activations are used for the decoder layers. A linear activation is used on the latent layer. The autoencoder is trained for 500 epochs with a batch size of 64, using an Adam optimizer. Details on activation functions and optimizers are given in Chapter 5.

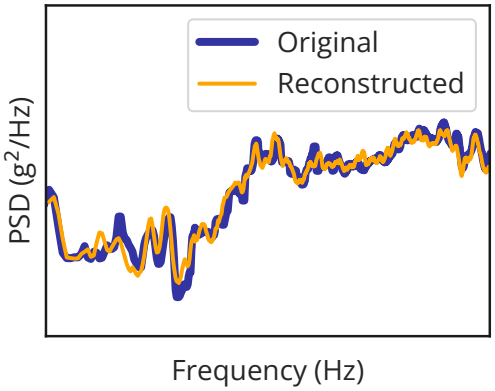


Figure 4.10: Reconstructed broadband spectrum using an autoencoder with a latent space of dimension 100, log-log scale.

An example of reconstruction is provided in Figure 4.10. It is worse than NMF and PCA while capturing the general trend. The average error using a latent space of dimension 100 is around 23%, while being 1.9% for PCA and 2.3% for NMF.

While autoencoders can be an interesting solution for nonlinear dimensionality reduction, they are not helpful for this specific problem. The performance could probably be greatly improved by adjusting the model architecture and its training process, but PCA already provides a significant reduction, and the RMS values per band give an even more compressed representation while still being domain-interpretable. The only use would be to combine the autoencoder with clustering.

Because of the previous results of PCA and NMF, autoencoder compression is not applied to RMS levels data.

4.2.6 . Comparison of reduction methods

In this section, frequency spectra are reduced using RMS levels per frequency band, PCA, NMF, and an autoencoder.

RMS levels per frequency band provides the best compression rate, reducing information to 0.70% of its original size. It is domain-interpretable but does not allow reconstruction of the original signal since the information is lost. **PCA and NMF** offer a compression rate of 4.1% and a reconstruction performance of respectively 1.9% and 2.1%. The components can be difficult to interpret because they are just a mathematical decomposition with no added domain constraints. While PCA is fast, NMF can take much longer to converge. The implemented **autoencoder** also offers a compression rate of 4.1% but with an average error of 23%. While its performance can most likely be greatly improved, this was not considered necessary.

[6] Dropout consists of not using a certain proportion of neurons in order to improve robustness.

As a result, the RMS values per frequency band are used for modeling, especially in Chapters 6 and 7. PCA is used before applying clustering, see Section 3.3.

4.3 . Data exploration

Valuable information can be extracted by examining the data. With the methods implemented for transforming it into a dataset now introduced, a compilation of insights is showcased.

4.3.1 . Correlations

Correlation matrices between general parameters and spectra can be analyzed to extract effects and help to tend towards a better understanding of the vibration environment.

A linear relationship can be captured through **Pearson's correlation coefficient** ρ. Between two random variables $\mathbf{x}$ and $\mathbf{y}$, it is defined as:

$$\rho(\mathbf{x}, \mathbf{y}) \triangleq \frac{\mathrm{cov}(\mathbf{x}, \mathbf{y})}{\sigma(\mathbf{x})\sigma(\mathbf{y})}, \tag{4.6}$$

with $\mathrm{cov}(\,\cdot\,,\,\cdot\,)$ the covariance and $\sigma(\,\cdot\,)$ the standard deviation.

To cover a broader range of relationships, **Spearman's correlation coefficient** r_s can be used. It describes monotonic relationships and corresponds to Pearson's correlation coefficient applied to ranked variables, i.e., variables ordered by their value:

$$r_s(\mathbf{x}, \mathbf{y}) \triangleq \rho(\mathrm{rg}_{\mathbf{x}}, \mathrm{rg}_{\mathbf{y}}), \tag{4.7}$$

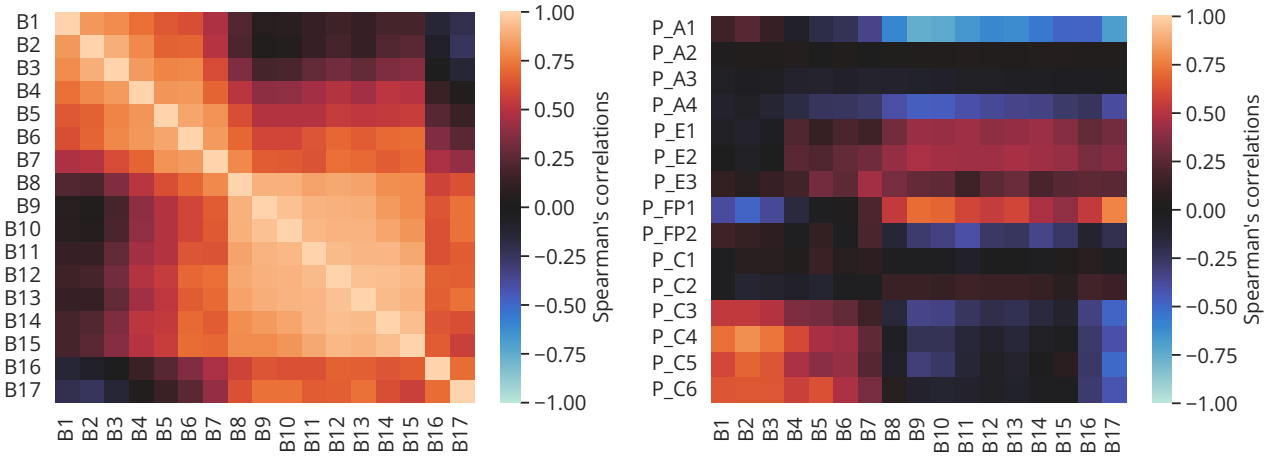
where $\mathrm{rg}_{\,\cdot}$ is a ranked variable.

It must be noted that correlations are only a partial indication of feature influence, as they only consider pairwise relationships and do not account for the influence of other variables[7] A high correlation is different from the importance of a variable to predict another variable and may be produce by non-generalizable phenomena. Moreover, some flight points can be under-represented in the dataset, and several variables are almost categorical, which are difficult to explain using correlations, such as flaps position which take discrete values: $0°$, $10°$, etc. For the continuous variables, flight test constraints make feature distributions highly multi-modal as the flight domain is not covered in a continuous manner, see later in Figure 7.15 for an example.

Correlations between RMS levels are presented in Figure 4.11a. The closer in frequency the bands, the higher their correlation coefficient. Two main groups can be identified, with a separation around 300 Hz. This is coherent with what was presented in Section 4.1.1, with a difference between the low-frequency range and the mid- and high-frequency range. Moreover, it shows the effect of high-lift devices, acting on the complete low-frequency range at once.

Correlations between general parameters and RMS levels are presented in Figure 4.11b. General parameters names are anonymized and refer to the following: attitude P_{A_1}, engines $P_{\mathsf{E}_{1\ldots3}}$, flight point $P_{\mathsf{FP}_{1\ldots3}}$, configuration $P_{\mathsf{C}_{1\ldots6}}$. Low frequencies are influenced mainly by parameters related to the

[7] Partial correlation tries to answer this problem, but it is limited to linear relationships and low orders of control variables.



(a) RMS levels across bands

(b) General parameters and RMS levels

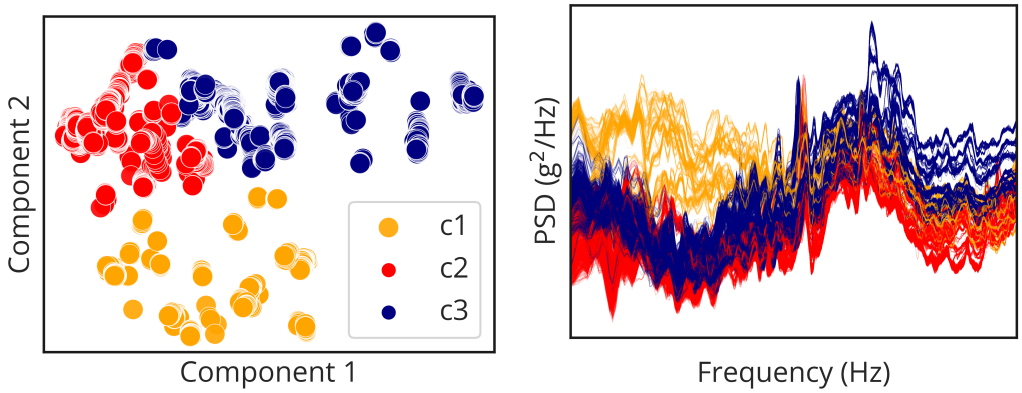
Figure 4.11: Correlation matrices, using Spearman's correlation coefficient.

aircraft configuration, such as $P_{C_{3...6}}$. Mid and high frequencies are mostly influenced by parameters related to aircraft attitude P_{A_1}, engines $P_{E_{1...2}}$, and flight point P_{FP_1}.

The two previous analyses should be conducted on a dataset containing only one accelerometer at a time: a dataset containing multiple accelerometers with opposite responses to certain effects leads to low correlation values.

4.3.2 . Clustering

Clustering, see Section 3.3, is a method that can help discovering hidden structures in the data. It is usually applied on reduced data, for example on PCA components.



(a) PCA projection

(b) Reconstructed spectra

Figure 4.12: Clustering result on measured spectra for one accelerometer. Projection on the first two principal components (left), reconstruction of broadband spectra on a log-log scale (right).

Clustering can be used to **group the measured spectra on a given accelerometer**. In Fig-

ure 4.12, 3 clusters are identified and visualized on the 2 first PCA-components projection, and on the reconstructed spectra. This clustering is done by applying a k-means algorithm on the first 2 PCA components. Using all components does not change the results, as the 2 first components explain most of the variance.

The result of this clustering performed on spectra in $\mathcal{Y}$ can be interpreted by looking at the general parameters space $\mathcal{X}$, as presented in Figure 4.13. This representation is a *parallel coordinates plot*, where all the dimensions are presented next to each other, and each data point consists of a line linking the different dimensions. The colors used here are the same as in Figure 4.12. The 3 clusters can be interpreted as follows:

- In blue, high dynamic pressures and clean configuration[8].
- In red, low dynamic pressures and clean configuration.
- in orange, low dynamic pressures and approach configuration[9].

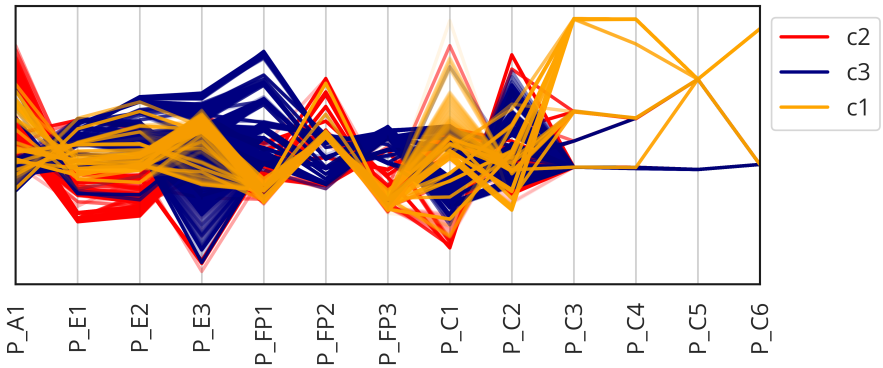


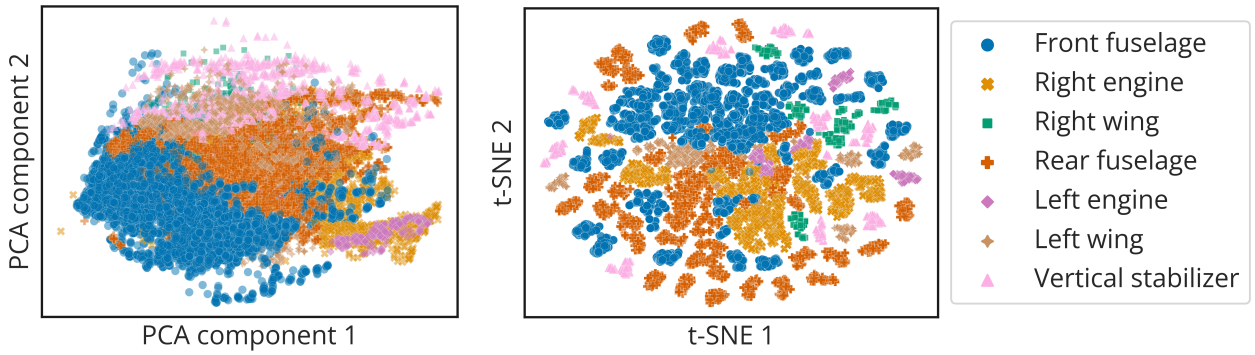
Figure 4.13: Parallel coordinates plot of general parameters, colored by cluster.

It must be noted that results highly depend on the accelerometer considered. For example, accelerometers closer to the engines are much less affected by the configuration. As a result, the choice of the number of clusters becomes challenging, as the goal is to categorize a continuous variation. In such case, an elbow plot has no elbow.

Clustering can also be applied to **group the measured spectra across accelerometers**, and infer which sensors are similar. This can be useful to optimize the FTI by identifying similar sensors that could be redundant.

In Figure 4.14a, each point is a spectrum reduced to PCA components. Multiple points can belong to the same accelerometer. The color corresponds to the standard aircraft zone division. It can be seen that accelerometers on the front of the aircraft have a low coefficient for the first PCA component, while this coefficient is high for accelerometers on the rear of the aircraft. This first PCA component is related to the total RMS value, which is indeed higher near the engines, at the rear of the aircraft. Next to the accelerometers on the rear side are the accelerometers on the engines.

[8]Clean configuration: high lift devices, air brakes, and landing gear are retracted.
[9]Approach configuration: high lift devices, air brakes, and/or landing gear are deployed.



(a) PCA projection

(b) t-SNE projection

Figure 4.14: Projection on two dimensions, using the two first PCA components (left) and t-SNE applied on the PCA components (right).

On Figure 4.14b, the same data points are visualized using a t-SNE (t-distributed Stochastic Neighbor Embedding) projection [vdMH08]. This stochastic method is used to visualize high-dimensional data and is based on random walks on neighborhood graphs. Here, the first 100 PCA components are projected on a 2D space. As with Figure 4.14a, front and rear accelerometers are mostly clustered, and accelerometers on engines are close to the rear ones. Moreover, the many small groups correspond to spectra of the same accelerometer. This representation gives more information and is maybe more readable.

4.4 . Conclusion on data representation and exploration

This chapter has introduced the strong hypothesis of spectrum separation, which states that a spectrum in the frequency domain can be represented as a broadband component with an amplitude shift and a collection of kinematic lines. Experimental results confirm this hypothesis, and an algorithm is implemented to perform the separation. It is compared to the rolling ball algorithm [Ste83] and a baseline estimation method [Maz22], and shows to be better suited for our use-case, although the rolling ball algorithm could be tried again.

Several reduced representations of spectra are compared and can be applied either to the raw spectra or to their broadband counterparts generated by the separation algorithm. RMS levels per frequency band are presented as the best representation to model broadband levels, being robust and domain-interpretable while offering the best compression rate. Finally, data exploration is performed using the tools presented so far: dataset generation from Chapter 2, data representation from Chapters 3 and 4, and clustering from Chapter 3. A correlation study is presented, especially between general parameters and frequency bands. This helped the modeling process in identifying the core features to use as model input. Then, clustering is applied and shows results consistent with domain knowledge and can be another tool to identify features. Finally, a parallel coordinates plot and a t-SNE projection are presented as an additional way to explore and visualize the data.

Summary of Part I

This part presents the transformations applied to experimental data from the Flight Test Center. This framework, which allows the generation of different types of vibration environment datasets, is the core of the **data-driven methodology** described in Part II and Part III.

First, an introduction to the **vibration environment** is given in Chapter 1. It describes what the vibration environment is, how it is generated, and why it needs to be studied in an industrial aerospace context. Several standard signal representations are provided, including some that are used in the methodology implemented in this manuscript.

Then, in Chapter 2, an overview of the general **dataset generation process** is presented. It includes identification techniques based on domain knowledge, which are described in detail. It also allows to apply several **dataset representation** techniques, which are presented in Chapter 3 together with clustering methods for feature extraction. In Chapter 4, the methods introduced in the previous chapter are applied to produce **reduced representation of vibration environment data** and to generate insights through data exploration. The chosen representation techniques used in the rest of the manuscript are justified here.

Part II

Prediction of new flight points

5 - Supervised learning: regression for predictive vibration environment

Contents

To predict the vibration environment of an aircraft, the first step is to make predictions at any point of the flight domain (or feature space) for any instrumented point. This is the objective of Part II. To this end, the tools of supervised learning and more specifically regression are introduced in this chapter. Then, Chapters 6 and 7 present an implementation of these tools for vibration environment prediction.

While Chapter 3 introduced unsupervised learning, this chapter proposes an introduction to supervised learning. As an illustrative example, let us consider the case of handwritten numbers recognition[1]. The classic approach is to take a set of inputs (images of handwritten numbers) and explicitly program an algorithm to produce a set of outputs (the identified numbers in the set of inputs). This task is complex, notably due to the diversity of handwritings. Supervised learning proposes to use the inputs and outputs to *learn* the algorithm linking them, provided that a certain algorithm structure is implemented as a backbone.

[1] This task belongs to the subfield of *Computer Vision*.

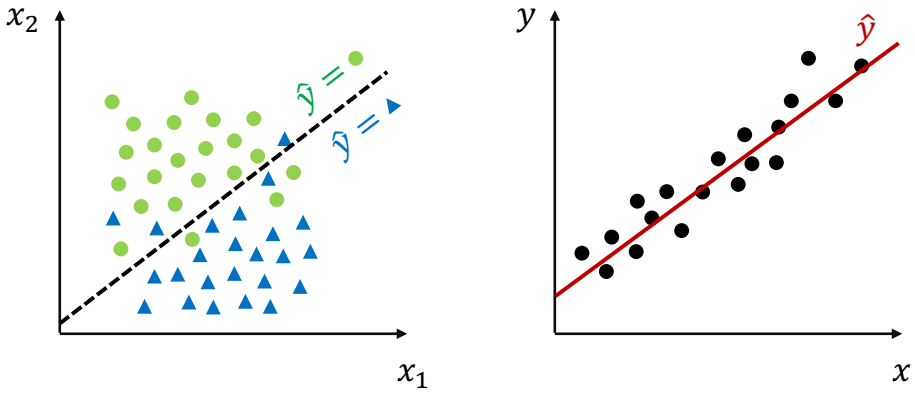
In the context of data generated at the FTC, supervised learning can be used to implement predictive models, for example, to predict vibration levels from the state of the aircraft. The difference between regression and classification is introduced in Section 5.1 along with notations. Several models are presented in Section 5.2, and the optimization process to learn the weights of parameterized models is detailed in Section 5.3, with implementation details in Section 5.4. The problem of model interpretability is tackled in Section 5.5, where the SHAP method is detailed.

5.1 . Notations, regression and classification tasks

Here and for the rest of the manuscript, a dataset is denoted as $\mathcal{D}$ and a model as $\phi \in \Phi$. In the case of a model parameterized by $\boldsymbol{\theta} \in \Theta$, it is denoted $\phi_{\boldsymbol{\theta}}$. In the context of supervised learning, the dataset is comprised of m labeled datapoints $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{1,\dots,m}$ such that $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^n$ and $\mathbf{y}_i \in \mathcal{Y} \subset \mathbb{R}^d$. A model prediction is denoted $\hat{\mathbf{y}}$. The link between inputs and outputs for parameterized models is performed such that:

$$\phi_{\boldsymbol{\theta}} : \mathcal{X} \times \Theta \to \mathcal{Y}. \tag{5.1}$$

Depending on the nature of the output space, the learning task can be a classification or a regression task, see Figure 5.1 for an illustration.



(a) Classification (b) Regression

Figure 5.1: Illustration of the classification and regression tasks.

Classification is used when the target variable is categorical and thus can take on a limited number of distinct values. The goal of classification is to assign input data to one of the predefined classes or categories. For example, classifying vibrational levels as within the specified range or not, $\mathcal{Y} = \{0, 1\}$, based on general flight parameters.

Regression is used when the target variable y is continuous, and thus can take on any numerical value within a defined range. The goal of regression is to predict the value of the target variable based on the input features. For example, predicting the values of vibrational levels, $\mathcal{Y} \subset \mathbb{R}_+$, based on general flight parameters.

5.2 . Models

Defining the class of models Φ is an empirical task performed through domain knowledge, experience, and trial and error. In this section, several models implemented in the following chapters are presented. Some of them, like Artificial Neural Networks, can also be used for unsupervised learning tasks, as presented previously in Section 3.2.3 about autoencoders.

Models parameters are denoted with the letter θ and contain *weights* (slopes) and *biases* (intercepts). Thus, the input vector $\mathbf{x}$ starts with a 1 to account for the bias, such that $\mathbf{x} = [1, x_1, \ldots, x_n]^\mathsf{T}$. While $\boldsymbol{\theta}$ is defined as a vector in Θ, it can take the form of a matrix, denoted as $\boldsymbol{\Theta}$. A bijection is then assumed to exist between $\boldsymbol{\theta}$ and $\boldsymbol{\Theta}$ by reorganizing its elements.

5.2.1 . Linear regression

One of the simplest models to implement is linear regression[2], which assumes a linear relationship between the input features and the target variable.

In multivariate linear regression, there are multiple input features, and the relationship between features and the target variable is represented by a hyperplane in a higher-dimensional space. In the multivariate and multi-output case, it is defined by:

$$\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}) = \hat{\mathbf{y}} = \boldsymbol{\Theta}\mathbf{x}, \tag{5.2}$$

with $\mathbf{x} = [1, x_1, \ldots, x_n]^\mathsf{T}$ the input vector containing n features, $\mathbf{y} = [y_1, \ldots, y_d]^\mathsf{T}$ the output vector of dimension d, and $\boldsymbol{\Theta}$ the matrix of coefficients, containing the bias as an additional column.

Linear regression has the advantages of simplicity, interpretability, and efficiency. However, it assumes a *linear* relationship between the input feature(s) and the target variable(s), which may not always hold in real-world scenarios. In such cases, more complex regression techniques may be used. Overall, it is a powerful tool for modeling and predicting continuous variables, and it serves as a foundation for more advanced regression techniques.

5.2.2 . Polynomial regression

Polynomial regression can be seen as an extension of linear regression where polynomial terms are introduced to capture nonlinear relationships. This allows for modeling more complex patterns in the data.

In the univariate case where x is used to predict y, a p-order polynomial regression is defined as:

$$\phi_\theta(x) = \hat{y} = \sum_{i=0}^{p} \theta_i x^i, \tag{5.3}$$

with $\theta_0 \ldots \theta_p$ the coefficients. Similarly to the linear regression, it can be written in the multivariate multi-output case similarly to Eq. (5.2):

$$\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}) = \hat{\mathbf{y}} = \boldsymbol{\Theta}\mathbf{x}_{\text{poly}}, \tag{5.4}$$

where $\mathbf{x}_{\text{poly}}$ is the vector concatenating the different polynomial orders of $\mathbf{x}$ up to order p, such that:

[2] Although powerful and with maybe one of the most comprehensive literature.

$$\mathbf{x}_{\text{poly}} = \begin{bmatrix} 1 \\ \mathbf{x} \\ \mathbf{x}^2 \\ \vdots \\ \mathbf{x}^p \end{bmatrix}. \tag{5.5}$$

This is a linear basis expansion [HTF09]. By introducing polynomial terms, polynomial regression can capture more complex relationships between the input features and the target variable.

5.2.3 . XGBoost

Decision trees use binary rules to compute the target value. A tree consists of a hierarchy of nodes, each of which can be split into sub-nodes. Decision-tree splitting is usually based on Mean Squared Error. A random forest [Bre01] prediction can be obtained by combining the decisions of an ensemble of decision trees applied to a random selection of features in a random subset of the training data. They are both illustrated in Figure 5.2.

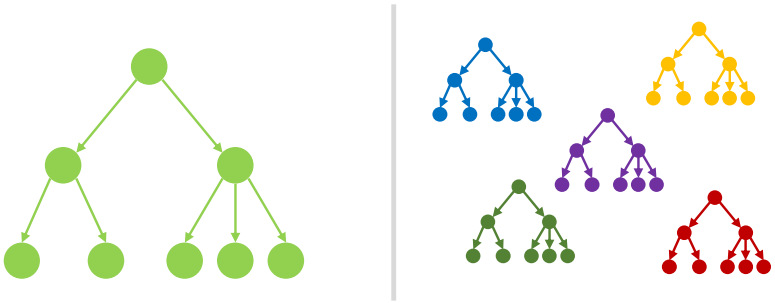


Figure 5.2: Decision tree (left) and random forest (right) illustration.

XGBoost (eXtreme Gradient Boosting) [CG16] is a library implementing tree-based gradient boosting techniques with parallel computing. It is used in many applications such as ground vibration prediction [NBBC19] and vehicle driving risk prediction [SWL+19]. The idea is to perform gradient boosting, where an ensemble of weak predictors are combined in an additive way such that:

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + f_t(x), \tag{5.6}$$

where f_t is a classification and regression tree (CART). It can be defined as:

$$f_t(x) = w_{q(x)}, \quad w \in \mathbb{R}^T, \quad q : \mathcal{X} \to \{1, 2, \ldots, T\}, \tag{5.7}$$

where w is the vector on leaves and $q(x)$ is a function that attributes each datapoint x to a specific leaf on the current tree t. The complexity of a tree can be defined as:

$$\omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{5.8}$$

where $\gamma > 0$ and $\lambda > 0$ are hyperparameters. Finally, the cost function is:

$$C(\boldsymbol{\theta}) = \sum_{i=1}^{N}(y - \hat{y})^2 + \sum_{t=1}^{T}\omega(f_t), \tag{5.9}$$

which aims to minimize the MSE as a cost function[3], while a regularization term on trees complexity $\omega(f_t)$ is included.

XGBoost is known for its high predictive accuracy for interpolation and its ability to handle complex relationships. However, one of its major drawbacks is that predictions are based only on a sum of values attached to tree leaves: no transformation is applied. As a result, it cannot perform out-of-sample extrapolation where it can only predict a constant.

5.2.4 . Gaussian process

A Gaussian process consists of learning the process that generates all the functions $\mathbf{f}$ that are the origin of the observed data, instead of learning the parameters of a function $\mathbf{f}(\mathbf{x}) = \hat{\mathbf{y}}$ that could have generated the observed data. It is an non-parametric model. To limit the space of functions, regularity constraints are applied: if $\mathbf{x}_1$ and $\mathbf{x}_2$ are close, then it is assumed that $\mathbf{y}_1$ and $\mathbf{y}_2$ are also close. A distribution on $\mathbf{x}$ is defined on an arbitrary set of points $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and each point $\mathbf{y}_1, \ldots, \mathbf{y}_m$ is a random variable. The joint probability $p(\mathbf{y}_1, \ldots, \mathbf{y}_m)$ is a multivariate normal distribution.

To constrain the space of functions and quantify regularity, an $(m \times m)$ covariance matrix is defined, quantifying the similarity of points based on their distance. The calculation of this covariance matrix is done using a kernel function or covariance function, which takes two points in space as input and returns their similarity based on a notion of distance. Given a kernel $k(\mathbf{x}, \mathbf{x}')$, the covariance matrix is constructed as follows:

$$\boldsymbol{\Sigma}(\mathbf{x}, \mathbf{x}') = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}'_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}'_m) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_m, \mathbf{x}'_1) & \cdots & k(\mathbf{x}_m, \mathbf{x}'_m) \end{bmatrix}. \tag{5.10}$$

This matrix must be positive definite. In addition to the covariance matrix, a mean function $\mu(\mathbf{x})$ is needed to fully characterize the multivariate normal distribution. Finally, as in [RW06], a function $\mathbf{f}$ that could have generated the data is described by:

$$\mathbf{f}(\mathbf{x}) \sim \boldsymbol{\phi}\left(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right). \tag{5.11}$$

There are various types of Gaussian process kernels, each with its own characteristics and applications. They can be combined using sum and product operators. Below is a short presentation of commonly used kernels.

The **RBF (Radial Basis Function) kernel** is based on the ℓ^2-norm, or Euclidean distance. The RBF kernel is smooth and infinitely differentiable, making it suitable for modeling smooth functions.

$$k_{\text{RBF}}\left(\mathbf{x}, \mathbf{x}'\right) \triangleq \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right), \tag{5.12}$$

[3] More details on cost functions in Section 5.3.

where l is the length scale parameter that controls the smoothness of the function.

The **Matérn kernel** is a family of kernels that includes the RBF kernel as a special case. It introduces a parameter that controls the smoothness of the function being modeled. The Matérn kernel is more flexible than the RBF kernel and can handle functions with different levels of smoothness.

$$k_{\text{Matérn}}\left(\mathbf{x}, \mathbf{x}'\right) \triangleq \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{l} \left\|\mathbf{x} - \mathbf{x}'\right\|_2\right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}}{l} \left\|\mathbf{x} - \mathbf{x}'\right\|_2\right), \tag{5.13}$$

where K_{ν} is a modified Bessel function, $\Gamma(\cdot)$ is the gamma function, and ν is a parameter that controls the smoothness of the function. As $\nu \to \infty$, the Matérn kernel converges to the RBF kernel.

The **linear kernel** measures the similarity between two points as a linear function of their coordinates. It assumes a linear relationship between the input and output variables:

$$k_{\text{Linear}}\left(\mathbf{x}, \mathbf{x}'\right) \triangleq \left\langle \mathbf{x}, \mathbf{x}' \right\rangle. \tag{5.14}$$

The **polynomial kernel** measures the similarity between two points as a polynomial function of their coordinates. It captures nonlinear relationships between the input and output variables:

$$k_{\text{Polynomial}}\left(\mathbf{x}, \mathbf{x}'\right) \triangleq \left(\left\langle \mathbf{x}, \mathbf{x}' \right\rangle + c\right)^{d}, \tag{5.15}$$

where c is a constant and d is the degree of the polynomial.

The **periodic kernel**, or Exp-Sine-Squared kernel, is used to model functions with periodic behavior. It measures the similarity between two points based on their periodic distance:

$$k_{\text{Periodic}}\left(\mathbf{x}, \mathbf{x}'\right) \triangleq \exp\left(-\frac{2\sin^2\left(\pi \left\|\mathbf{x} - \mathbf{x}'\right\|_2 / p\right)}{l^2}\right), \tag{5.16}$$

where l is the length scale of the kernel and p is the periodicity of the kernel.

The **constant kernel** assigns a constant value to the similarity between any two points:

$$k_{\text{Constant}}\left(\mathbf{x}, \mathbf{x}'\right) \triangleq C \quad \forall\, \mathbf{x}, \mathbf{x}'. \tag{5.17}$$

The **white noise kernel** assigns a constant variance to the similarity between identical points:

$$k_{\text{White}}\left(\mathbf{x}, \mathbf{x}'\right) \triangleq \begin{cases} w & \text{if } \mathbf{x} = \mathbf{x}', \\ 0 & \text{otherwise} \end{cases} \tag{5.18}$$

where w is a noise level to set.

5.2.5 . Artificial Neural Network

An ANN (Artificial Neural Network) is a model comprised of different layers: an input layer, one or several hidden layers, and an output layer. Each layer has a collection of neurons, which are basic computational units taking in an input, transforming it, and producing an output, see Figure 5.3. The value of a neuron is defined as:

$$a \triangleq g(\langle \boldsymbol{\theta}, \mathbf{x} \rangle), \tag{5.19}$$

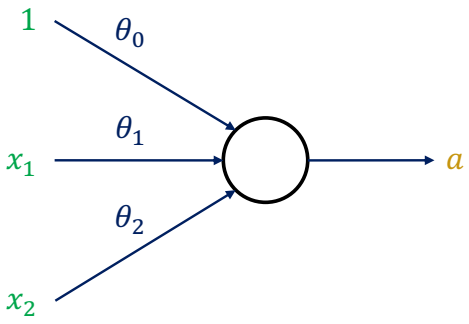
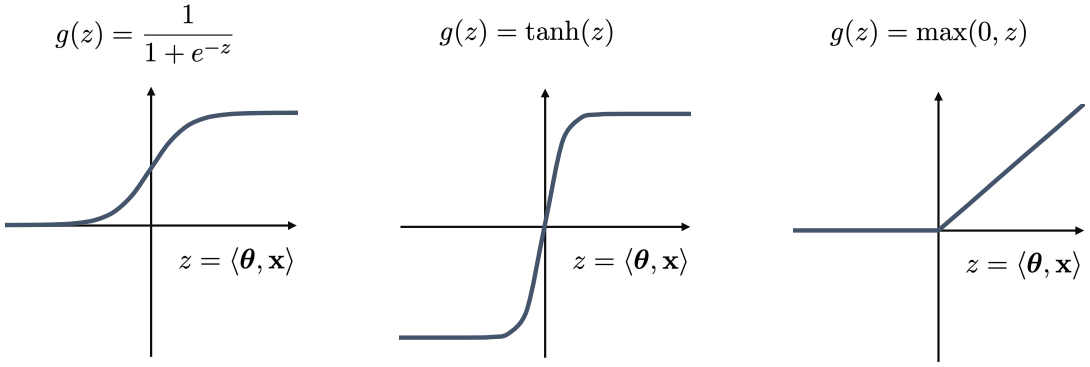


Figure 5.3: Illustration of a neuron.

with $\mathbf{x}$ the input value, $\boldsymbol{\theta}$ the weights, and g the activation function.

The activation function serves the purpose of introducing a nonlinearity, allowing the modeling of complex relationships. Several activation functions exist, such as sigmoid, tanh, or ReLU (Rectified Linear Unit), and are illustrated in Figure 5.4.



(a) sigmoid (b) $\tanh$ (c) ReLU

Figure 5.4: Activations functions

Each layer can be seen as a multi-dimensional space, with its dimension being the number of neurons in that specific layer. The value of all neurons in a layer l can be defined as:

$$\mathbf{a}^{(l)} \triangleq \mathbf{g}^{(l)}(\boldsymbol{\Theta}^{(l)}\mathbf{a}^{(l-1)}) = \mathbf{f}^{(l)}(\mathbf{a}^{(l-1)}), \tag{5.20}$$

with $\mathbf{a}^{(l)}$ the vector of neurons values at layer l (called activation or hidden state), $\boldsymbol{\Theta}^{(l)}$ the matrix of weights and biases, and $\mathbf{g}^{(l)}$ the activation function. Denoting $\mathbf{f}^{(l)}$ the transformation from layer $l-1$ to layer l, the general equation of an L-layer ANN can be written as:

$$\phi_{\boldsymbol{\theta}}(\mathbf{x}) = (\mathbf{f}^{(L)} \circ \ldots \circ \mathbf{f}^{(3)} \circ \mathbf{f}^{(2)})(\mathbf{x}) = \hat{\mathbf{y}}, \tag{5.21}$$

where $\phi_{\boldsymbol{\theta}}$ is parameterized by $\boldsymbol{\theta} = (\boldsymbol{\Theta}^{(2)}, \ldots, \boldsymbol{\Theta}^{(L)})$.

This specific configuration where each neuron in a layer is connected to all neurons in the next layer is called a **Feedforward Neural Network**, see Figure 5.5.

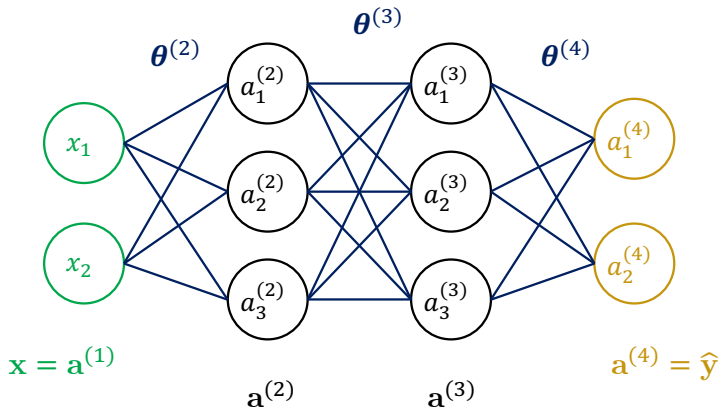


Figure 5.5: Illustration of a Feedforward Neural Network with 2 hidden layers.

Each new layer is built on the previous one and adds a level of complexity. As the number of neurons in a given layer defines its dimension, the number of layers in an ANN is relative to its complexity level. The deeper the model, the more complex the relationships that can be modeled.

Let us give an illustration of how layers relate to complexity level. Consider the regression problem of predicting house prices, where inputs are measurements such as surface, number of rooms, location, etc. The first layer can build concepts based on those inputs, such as the overall size of the house, by using surface and number of rooms. The second layer builds concepts built on top of the first layer. For example, the value of one neuron could be linked to how upscale the neighborhood is: this is a better *feature* (independent variable) than the initial raw measurement, with a higher complexity level. Note, however, that the actual meaning of values is most of the time hard to interpret. A brief overview of interpretability methods is given in Section 5.5, with a detail on the SHAP method in Section 5.5.4.

5.2.6 . Recurrent Neural Network

Multiple architectures of ANNs exist to encode specific relationships in the data. One of them is the **RNN (Recurrent Neural Network)**, which encodes temporal relationships, see Figure 5.6. This architecture allows to capture dependencies and patterns in sequences, making it useful for tasks such as language modeling, speech recognition, translation, and time series forecasting. Regarding forecasting, RNNs can surpass classical methods such as autoregressive models in the case of high dimensional data ($d > 10$), even when they are adapted to the multivariate case, see Vector Autoregressive model [ziv06, Lüt13].

For each time step (t) a cell produces an activation, passed to the cell at the next time step, and an output. The activation is defined as:

$$\mathbf{a}^{(t)} \triangleq \mathbf{g}_a \left(\mathbf{\Theta}_a \left[\mathbf{a}^{(t-1)}, \mathbf{x}^{(t)} \right] \right), \tag{5.22}$$

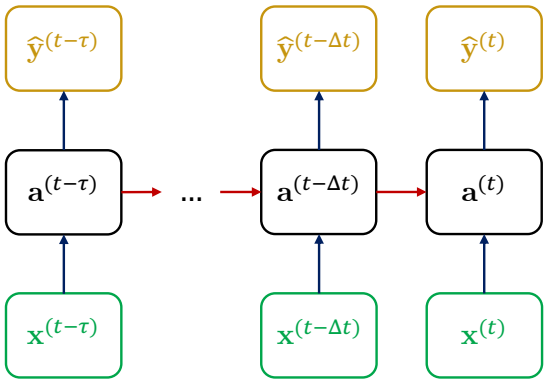


Figure 5.6: Illustrations of a Recurrent Neural Network, considering the time from $t-\tau$ to t each Δt.

where $\boldsymbol{\Theta}_a \left[\mathbf{a}^{(t-1)}, \mathbf{x}^{(t)}\right] \triangleq \boldsymbol{\Theta}_{aa}^{(t)}\mathbf{a}^{(t-1)} + \boldsymbol{\Theta}_{ax}^{(t)}\mathbf{x}^{(t)}$ with $\boldsymbol{\Theta}_{aa}$ and $\boldsymbol{\Theta}_{ax}$ being parameters contained in $\boldsymbol{\Theta}_a$. The output is defined as:

$$\hat{\mathbf{y}}^{(t)} \triangleq \mathbf{g}_y\left(\boldsymbol{\Theta}_{ya}^{(t)}\mathbf{a}^{(t)}\right). \tag{5.23}$$

An issue with a basic RNN architecture is its limitations in capturing long-term dependencies, notably because of the vanishing and exploding gradient problem [Hoc91, BSF94, PMB13]. To address this, more advanced architectures have been developed.

LSTM (Long Short-Term Memory) [HS97] introduces memory cells and gating mechanisms to better capture and preserve long-term dependencies in sequential data. Gates control the flow of information into and out of the memory cells, allowing the LSTM to remember or forget information over time selectively. A gate is defined as:

$$\boldsymbol{\Gamma}_\cdot \triangleq \boldsymbol{\sigma}\left(\boldsymbol{\Theta}_\cdot \left[\mathbf{a}^{(t-1)}, \mathbf{x}^{(t)}\right]\right), \tag{5.24}$$

where $\boldsymbol{\sigma}$ is the sigmoid activation function.

The forget gate $\boldsymbol{\Gamma}_f$ determines which information from the previous state should be kept, the input gate $\boldsymbol{\Gamma}_i$ determines which values from the current state should be updated, and the output gate $\boldsymbol{\Gamma}_o$ determines which information should be used for the hidden state $\mathbf{a}^{(t)}$.

The memory cell acts as a long-term memory storage unit, enabling the LSTM to capture and propagate information across multiple time steps. It is defined as the combination of previous information passed into the forget gate, and current information passed into the input gate, such that:

$$\mathbf{c}^{(t)} \triangleq \boldsymbol{\Gamma}_f \odot \mathbf{c}^{(t-1)} + \boldsymbol{\Gamma}_i \odot \widetilde{\mathbf{c}}^{(t)}, \tag{5.25}$$

where $\odot$ denotes the Hadamard product, or element-wise product, and $\widetilde{\mathbf{c}}^{(t)}$ are the candidate values for the current state and are defined as:

$$\widetilde{\mathbf{c}}^{(t)} \triangleq \mathbf{tanh}\left(\boldsymbol{\Theta}_c \left[\mathbf{a}^{(t-1)}, \mathbf{x}^{(t)}\right]\right). \tag{5.26}$$

The output of the cell is the hidden state, computed from the cell state:

$$\mathbf{a}^{(t)} \triangleq \mathbf{\Gamma}_o \odot \mathbf{tanh}\left(\mathbf{c}^{(t)}\right). \tag{5.27}$$

Many variants of LSTMs have been developed, such as LSTMs augmented by *peephole connections* [GS00], or Depth Gated RNN [GG15]. **GRU (Gated Recurrent Unit)** [CvG+14] is another type of RNN architecture that also addresses the vanishing and exploding gradient problem and captures long-term dependencies. It simplifies the LSTM architecture and has shown comparable performance to LSTMs while using fewer parameters. GRU combines the input and forget gates into a single update gate $\mathbf{\Gamma}_u$. Additionally, it introduces a reset gate $\mathbf{\Gamma}_r$ that controls the flow of information from previous time steps. Finally, the cell state and hidden state are combined:

$$\mathbf{a}^{(t)} \triangleq (1 - \mathbf{\Gamma}_u) \odot \mathbf{a}^{(t-1)} + \mathbf{\Gamma}_u \odot \widetilde{\mathbf{a}}^{(t)}, \tag{5.28}$$

where $\widetilde{\mathbf{a}}^{(t)}$ are the candidate values for the current hidden state, and are defined as:

$$\widetilde{\mathbf{a}}^{(t)} \triangleq \mathbf{tanh}\left(\mathbf{\Theta}_a \left[\mathbf{\Gamma}_r \odot \mathbf{a}^{(t-1)}, \mathbf{x}^{(t)}\right]\right). \tag{5.29}$$

Apart from LSTM and GRU, there are other variants of RNNs that have been developed to address specific challenges or improve performance in different scenarios. Bidirectional RNNs process the input sequence in both forward and backward directions, allowing the network to capture information from past and future contexts simultaneously. This is particularly useful in tasks where the current prediction depends on context around the information being processed, such as speech recognition and sentiment analysis.

Attention Mechanisms enhance the capability of RNNs to focus on relevant parts of the input sequence by assigning different weights to different time steps or input elements. They have been widely used in tasks such as machine translation, image captioning, and text summarization.

Transformer Networks [VSP+17] is a type of architecture that relies solely on self-attention mechanisms, eliminating the need for recurrent connections. They have been highly successful in natural language processing tasks, such as machine translation and language understanding, due to their ability to capture long-range dependencies and parallelize computations effectively[4].

5.2.7 . Ensembles and Deep ensembles

Ensemble models combine the predictions of multiple individual estimators to make a final prediction. The idea behind ensemble modeling is that by combining the strengths of different models, the overall performance and accuracy can be improved compared to using a single model. There are several types of ensemble models, including:

- **Bagging**, short for bootstrap aggregating, involves training multiple models on different subsets of the training data, typically using the same algorithm. Each model is trained independently, and their predictions are combined through averaging or voting to make the final prediction. Random Forest is a popular example of a bagging ensemble model.

[4] For demonstration purposes, this paragraph has been generated by a transformer.

- **Boosting** is a technique where multiple weak models are trained sequentially, with each subsequent model focusing on the samples that the previous models misclassified. The predictions of all the models are combined using weighted voting or averaging. Gradient Boosting Machines (GBM) and AdaBoost are common boosting ensemble models.
- **Stacking** involves training multiple models on the same dataset and then using another model, called a meta-model or blender, to combine their predictions. The meta-model is trained on the predictions of the individual models, learning to make the final prediction based on their outputs. Stacking can be more complex and computationally intensive but often leads to improved performance.

While these techniques are more computationally expensive than non-ensemble models, they offer improved accuracy, robustness, and generalization. They can also be used with ANNs[5], for example, using deep ensembles. A deep ensemble [LPB17] is an ensemble of neural networks allowing to quantify model uncertainties. It was recently shown that this method produces better results than other uncertainty quantification methods [FHL20]. As with classical ensembles, they require much more computational power than their single counterpart. Finally, more recent approaches exist, such as model soups [WIG+22], proposing to average the weights of multiple models fine-tuned with different hyperparameter configurations to improve accuracy and robustness.

5.3 . Optimizing weights and hyperparameters

After introducing several classes of models, this section describes the optimization process through which the model parameters are learned.

5.3.1 . Cost function

The first step to learn the weights of a model is to define an objective. This is done through the use of a *cost function* $C(\boldsymbol{\theta})$, also known as an objective function, which quantifies the average difference between predictions $\hat{\mathbf{y}}$ of a model and the actual values to predict $\mathbf{y}$. A cost function averages the error over all training points, while a *loss function* computes the error for a single data point.

Different types of cost functions are used depending on the problem at hand. For regression problems, the two most common cost functions are based on the ℓ^1 and ℓ^2-norms: the MAE (Mean Absolute Error) and the MSE (Mean Squared Error). For multi-output problems where there is m datapoints of dimension d, they can be written as:

$$\mathrm{MAE} \triangleq \frac{1}{d}\frac{1}{m}\sum_{i=1}^{m} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_1 \,, \tag{5.30}$$

$$\mathrm{MSE} \triangleq \frac{1}{d}\frac{1}{m}\sum_{i=1}^{m} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 \,. \tag{5.31}$$

While MAE treats all errors equally, MSE penalizes larger errors more heavily as they are squared. Note that this multi-output case is not conventional, and a choice of giving the same weight to all

[5] Whether they are feedforward, recurrent, or other types of neural networks.

output dimensions is made in this definition. Other metrics can be used to evaluate the results, but are not used as a cost function because of their properties. For example, this is the case of the MAPE (Mean Absolute Percentage Error):

$$\mathrm{MAPE} \triangleq \frac{1}{d}\frac{1}{m}\sum_{j=1}^{d}\sum_{i=1}^{m}\left|\frac{y_{ij}-\hat{y}_{ij}}{y_{ij}}\right|. \tag{5.32}$$

While this metric is easy to interpret as it gives a percentage error, it has multiple problems. First, it is not symmetric, meaning that interchanging y and $\hat{y}$ gives a different error and favors under-prediction as a result. Second, it can give infinite or extremely high values for small absolute differences if the actual value is close to zero, which can lead to misleading results. Finally, it favors under-prediction.

As a solution to the limitations of MAPE, other metrics have been introduced, such as the log of the accuracy ratio, which does not favor under predictions and which expected value is the geometric mean [Tof15]. Extended to the multi-output case, it is defined here as the MLAR (Mean Log Accuracy Ratio):

$$\mathrm{MLAR} \triangleq \frac{1}{d}\frac{1}{m}\sum_{j=1}^{d}\sum_{i=1}^{m}\log\left(\frac{\hat{y}_{ij}}{y_{ij}}\right). \tag{5.33}$$

In this manuscript, two metrics are introduced for model evaluation. The first one is named **MALE (Mean Absolute Log Error)**, and is defined as:

$$\mathrm{MALE} \triangleq \frac{1}{d}\frac{1}{m}\sum_{j=1}^{d}\sum_{i=1}^{m}\left|\log_{10}\frac{\hat{y}_{ij}}{y_{ij}}\right| \times 100. \tag{5.34}$$

It can be considered as a mix of the MAE (as it can be rewritten using an ℓ^1-norm) and the MLAR. With this metric, a 1-order-of-magnitude error amounts to a 100% error. The second metric introduced is an alternative to the previous one but made to evaluate the maximum error across all samples and dimensions:

$$\mathrm{maxALE} \triangleq \max_{i,j}\left|\log_{10}\frac{\hat{y}_{ij}}{y_{ij}}\right| \times 100. \tag{5.35}$$

To improve the generalization of a model, a technique is to keep the magnitude of the weights small by regularization methods, for instance, by adding a penalty based on the weight magnitude to the cost function. This is done through the use of ℓ^1-penalty (LASSO, or Least Absolute Shrinkage and Selection Operator), ℓ^2-penalty (ridge/Tikhonov), or both (elastic net). An example of elastic net is given below:

$$C(\theta) = \frac{1}{d}\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{y}_i-\hat{\mathbf{y}}_i\|_2^2 + \lambda_1\|\boldsymbol{\theta}\|_1 + \lambda_2\|\boldsymbol{\theta}\|_2^2, \tag{5.36}$$

where λ_1 and λ_2 are regularization parameters.

5.3.2 . Optimization process

To minimize the cost function, an optimization process is performed. The goal is to find the best weights $\boldsymbol{\theta}^\star$ such that:

$$\boldsymbol{\theta}^\star \in \underset{\boldsymbol{\theta}\in\Theta}{\arg\min}\, C(\boldsymbol{\theta}). \tag{5.37}$$

The most commonly used optimization algorithm is *gradient descent*, a first-order optimization algorithm[6].

Gradient descent iteratively adjusts the model's parameters by following the direction of the steepest descent of the cost function. Depending on how many data points are used to compute the gradients, it can be called stochastic gradient descent (one sample), mini-batch gradient descent (a mini-batch sample), or batch gradient descent (all samples). The batch size is the number of samples used in one iteration of gradient descent. An epoch is one or multiple steps of gradient descent, after which all samples have been seen by the algorithm. The update rule is:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} C, \tag{5.38}$$

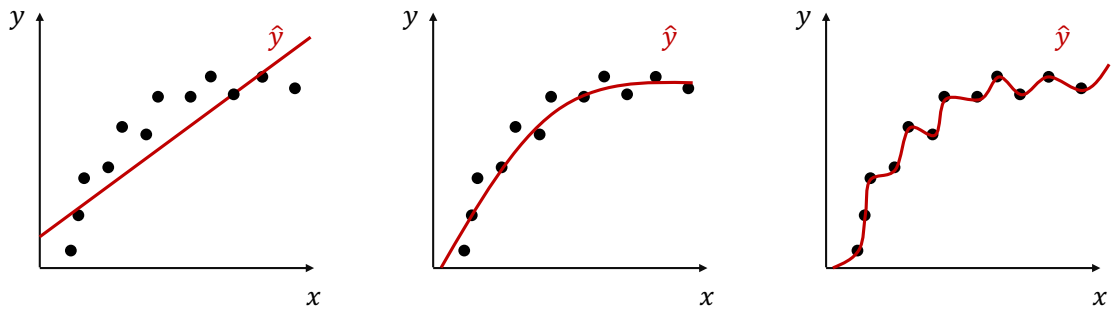
with $\nabla_{\boldsymbol{\theta}} C$ the partial derivative of C with respect to $\boldsymbol{\theta}$, and α the learning rate. The learning rate determines the step size at which the parameters are updated during the optimization process. If too small, convergence may be too slow, if too large, convergence may never happen. The learning rate can be adapted during the optimization process.

Multiple improvements in gradient descent have been proposed. **Momentum-based gradient descent** introduces a momentum term that accumulates the gradients of previous iterations and adds it to the current update. This helps to accelerate convergence, especially in the presence of sparse gradients or noisy data. **Adagrad** [DHS11] adapts the learning rate for each parameter based on the historical gradients. It performs larger updates for infrequent parameters and smaller updates for frequent parameters. Adagrad is effective in handling sparse data but may suffer from a diminishing learning rate over time. **RMSprop** is an extension of Adagrad that addresses the diminishing learning rate problem. It introduces a decay factor to control the accumulation of historical gradients, allowing the learning rate to adapt more effectively. **Adam (Adaptive Moment Estimation)** [KB17] combines the ideas of momentum and RMSprop. It maintains both the first-order moment (the mean) and the second-order moment (the uncentered variance) of the gradients. Adam is widely used and performs well in practice due to its adaptive learning rate and momentum. **Nadam** [Doz16] is an extension of Adam that incorporates the Nesterov accelerated gradient (NAG) method. NAG looks ahead in the direction of the momentum update before computing the gradient, which helps to improve convergence.

Learning a model parameters is subject to the trade-off between bias and variance. As illustrated in Figure 5.7, high bias leads to *underfitting* while high variance leads to *overfitting*. Here, bias refers

[6]Second-order optimization algorithms such as Newton's method are less used as they require to compute the Hessian matrix, which can be computationally expensive, especially for large datasets.

to the error introduced by the model's assumptions or simplifications. A high-bias model tends to oversimplify the relationships between the features and the target variable. Variance refers to the error introduced by the model's sensitivity to the fluctuations in the training data. A model with high variance tends to capture the noise and random fluctuations in the training data. In both under and overfitting cases, the model will tend to generalize poorly.



(a) Underfitting (b) Balanced trade-off (c) Overfitting

Figure 5.7: Illustration of the bias-variance trade-off.

To find the right balance between bias and variance, multiple techniques can be used, such as regularization (see at the end of Section 5.3.1), which helps lowering the variance.

5.3.3 . Evaluating performances

The end goal of a trained model is to apply it to new data samples, unused for training, which is dependent on its *generalization* ability. To evaluate this, a proportion of the dataset is kept away and not used in training. In general, a dataset is split into:

- A training set, used to train the model and adjust its parameters through gradient descent.
- A validation set, used to evaluate models and adjust their hyperparameters. This is the *model selection* process.
- A test set, used for the final evaluation. This set is necessary as the model is biased towards the validation set.

The proportion of data in the training-validation-test sets is an empirical choice, varying depending on the problem and the available data. As a general rule of thumb, a split into 80%-10%-10% is a good starting point.

The initial choice of the validation set can bias the hyperparameters optimization. To avoid this, **cross validation** can be used, for example, k-fold cross validation illustrated in Figure 5.8. First, the whole dataset is divided into a training-validation set and a test set. Then, the training set is split into k folds. For each split, the model is trained on $k-1$ folds and validated on the last fold. When a model is chosen, it can finally be evaluated on the test set. It has the benefit of providing a less biased estimate of model performances at the cost of an increased computational time.

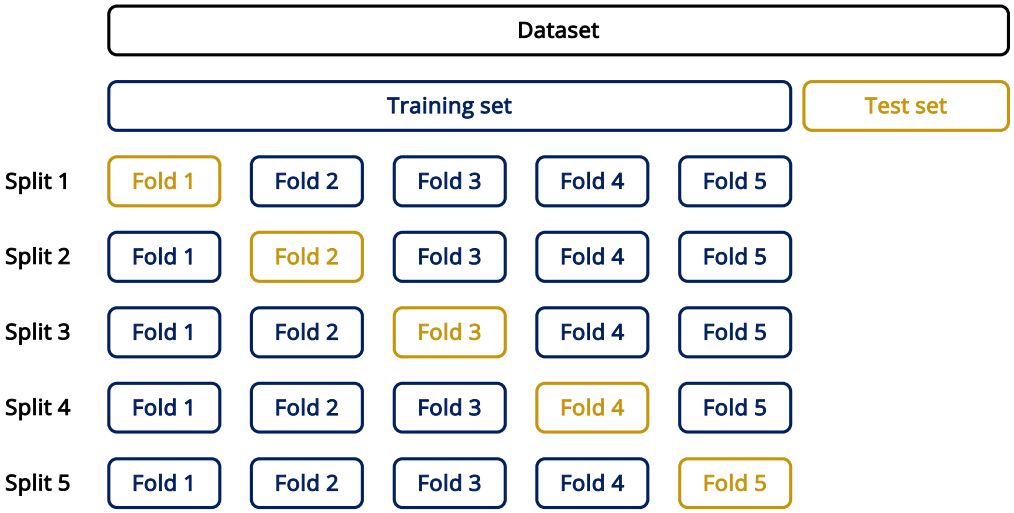


Figure 5.8: Illustration of the cross-validation process using 5-folds.

In classical machine learning the different sets are assumed to be i.i.d., independent and identically distributed. However, in reality, this assumption is often false, and the challenge is to generalize to distribution shifts, see Out-of-Distribution generalization [LSH+23]. For example, in the prediction problem presented in Chapter 7, a way to evaluate models is to perform extrapolation along the dynamic pressure variable, thus perform a prediction under a distribution shift. In this context, such extrapolation is justified solely by its alignment with domain-specific knowledge; otherwise, it would be considered imprudent

5.4 . Practical implementation

Now that the theoretical aspects have been covered, the question arises as to how this knowledge can be translated into practical application. Machine learning is greatly iterative by nature, as it involves refining and improving models based on feedback and discoveries. Figure 5.9 illustrates this iterative process.

The overall process begins with selecting, extracting, and cleaning a subset of data from a potentially large database. For a regression problem, the choice of features and output variables is crucial in defining the problem to be solved. Then, the data must be transformed to find a suitable representation. This step often involves feature engineering and preprocessing techniques.

Once the data is prepared, a model architecture is selected, and its weights are learned through training. However, each step in this process can lead to discoveries and insights that may challenge or invalidate previous choices. This iterative nature of machine learning requires constant evaluation, experimentation, and adaptation to improve the model's performance and address any limitations or shortcomings.

Lately, AutoML (Automated Machine Learning) has emerged as a powerful tool in the field of machine learning [HZC21]. By automating the iterative process of feature engineering, model selection, and hyperparameter tuning, AutoML can reduce the burden of manual trial and error, saving time

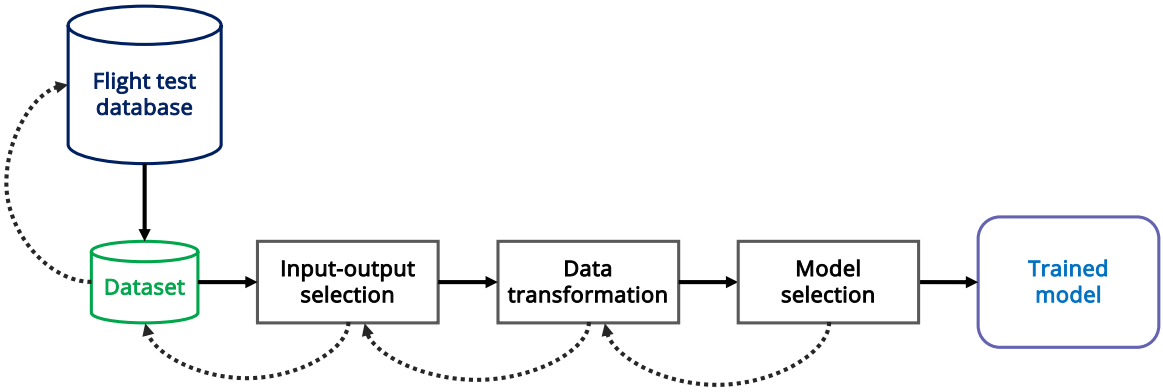


Figure 5.9: Illustration of the process to obtain a trained model from a flight test database.

and resources.

While this iterative process should be able to identify the best-performing model on the given dataset, there is no guarantee that the features used in a black box model for prediction are actually predictive. The dataset could be biased, and certain features could be used to make good predictions that would not generalize. In addition to a careful selection of the dataset and how it is partitioned, interpreting model predictions is an essential step in being able to trust its ability to generalize.

5.5 . Interpretability: understanding a model's predictions

The emergence of the field of explainable AI stems from the need to understand the predictions made by machine learning models. In an aeronautical context, interpretability is critical and could be subject to standards regulation in the near future.

Multiple definitions of interpretability have been given. In [Mil19], it is defined as « the degree to which a human can understand the cause of a decision. », while in [KKK16] it is defined as « the degree to which a human can consistently predict the model's result. ». In both cases, the definition is broad but offers a baseline for what is expected from an interpretation method.

According to [RB18], the quality of explanation methods and individual explanations can be assessed through several properties. **Explanation methods** have properties such as expressive power, translucency (how much the method relies on analyzing the model's parameters), portability, and algorithmic complexity. **Individual explanations** have properties such as accuracy, fidelity, consistency (between models trained on the same task), stability (between similar samples), comprehensibility, certainty, degree of importance, novelty, and representativeness.

In [Mil19], several **human-centered criteria** to define a "good" explanation are presented. An explanation should be contrastive [Lip90] by answering the question, "Why this prediction instead of another?". It should focus on the primary cause (selectiveness), be domain-oriented, and focus on the abnormal. Indeed, people tend to focus on abnormal causes to explain events [KT81]. Thus, a prediction should highlight any feature with abnormal values, even when there are other features with the same amount of influence but with standard values. Finally, an explanation should be truthful (having a high fidelity), consistent with prior beliefs of the explainee (otherwise, it tends to be ignored

through confirmation bias [Nic98]), and probable.

In this section, multiple *model-agnostic methods* [RSG16a] detailed in [Mol22] are presented, allowing consistent interpretation of different models on the same dataset without being tied to a specific architecture. A focus is made on SHAP, used in Chapter 7 to explain model predictions.

5.5.1 . An overview of local model-agnostic interpretation methods

Model-agnostic interpretation methods can be separated into global methods, aiming to describe how features affect the prediction on average, and local methods, aiming to explain individual predictions. Here, local methods are presented, but parallels are made with global methods.

An **ICE (Individual Conditional Expectation) plot** [GKBP15] shows the effect of a feature on the prediction for each sample. One line is drawn per sample, and one plot is given for each feature. While this method is local, taking its average gives the **PDP (Partial Dependence Plot)**, a global method [Fri01].

Counterfactual explanations are in the form of "if event A had not occurred, then event B would not have occurred". They allow to describe the smallest change to the feature values that changes the prediction to a predefined output. Several implementations exist, such as [WMR18] and [DMBB20]. They are contrastive and selective, thus human-friendly, and allow for third-party auditing, as no access to the model is required, only to its predictions. However, they are prone to providing multiple contradictory explanations.

From the authors of LIME, see below, the **anchors** method [RSG18] proposes an explanation based on scoped rules called anchors, representing local, "sufficient" conditions for predictions. While limited to classification models, it provides easy-to-interpret explanations and works even when model predictions are nonlinear or complex. However, it has many hyperparameters to set and requires to perform many inferences as other perturbation-based explainers.

Finally, there are Shapley values and SHAP, based on Shapley values. Since this is the selected method to interpret our models, a detailed description is given in the following sections. First, LIME is introduced as it is closely related.

5.5.2 . LIME

Local surrogate models propose to explain individual predictions by using another interpretable model. **LIME (Local Interpretable Model-agnostic Explanations)** [RSG16b] is a concrete implementation of this idea. An interpretable model (such as a linear regression or a decision tree) is used to learn a local approximation of the original black box model to be explained.

Let us define an input $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}' \in \{0,1\}^{n'}$ a binary vector, and $\phi : \mathbb{R}^n \to \mathbb{R}$ the model to be explained. The explanation provided by LIME is written as:

$$\zeta(\mathbf{x}) \triangleq \underset{g \in G}{\arg\min}\, C(\phi, g, \pi_x) + \Omega(g), \tag{5.39}$$

where $g \in \{0,1\}^{n'}$ is the explanation model, $\Omega(g)$ is a measure of complexity[7], $\pi_x(\mathbf{z})$ is a proximity measure[8] between $\mathbf{z}$ and $\mathbf{x}$, and $C(\phi, g, \pi_x) + \Omega(g)$ is a cost function measuring of how faithful g is

[7] This measure of complexity acts as a regularization term, as introduced in Eq. (5.36).

[8] The proximity measure is defined as a Gaussian kernel here.

in approximating ϕ in the neighboorhood $\pi_x(\mathbf{z})$. Moreover, G is the set of linear models $g(\mathbf{z}') = \theta_g \mathbf{z}'$, and the cost function is defined as:

$$C(\phi, g, \pi_x) \triangleq \sum_{z,z'} \pi_x(\mathbf{z})(\phi(\mathbf{z}) - g(\mathbf{z}'))^2. \tag{5.40}$$

This method has the advantage of providing contrastive explanations (thus human-friendly), and the fidelity of the surrogate model informs on the reliability of its explanations. However, it is challenging to choose how "local" the surrogate model should be, and explanations can lack robustness, where similar inputs give rise to different explanations [AJ18]. Finally, this method is prone to adversarial attacks, allowing to hide biases in the explanation while still using them for prediction [SHJ+20].

5.5.3 . Shapley Values

Shapley values [Sha51] allow to explain a prediction by representing the average marginal contribution of each feature across all possible feature combinations. This concept comes from cooperative game theory, where Shapley values quantify how much each player (feature) contributes to a game (prediction), when different players collaborate in different combinations.

To showcase the computation of Shapley values, the feature contribution in a linear model is first defined. Let us denote ζ_{ij} the feature contribution of feature j for example i. This is how much the j-th feature contributed to the prediction $\hat{y}_i = \phi_{\boldsymbol{\theta}}(x_i)$ compared to the average prediction for the entire dataset:

$$\zeta_{ij}(\hat{y}_i) \triangleq \theta_j x_{ij} - \mathrm{E}\left[\theta_j X_{:j}\right], \tag{5.41}$$

where $\mathrm{E}\left[\theta_j X_{:j}\right]$ is the estimation of the mean effect for feature j. The sum of contributions for an example i is the difference between the prediction at x_i and the average predicted value across all m examples:

$$\begin{aligned}
\sum_{j=1}^{n} \zeta_{ij}(\hat{y}_i) &= \sum_{j=1}^{n} \left(\theta_j x_{ij} - \mathrm{E}\left[\theta_j X_{:j}\right]\right) \\
&= \sum_{j=1}^{n} \theta_j x_{ij} - \sum_{j=1}^{n} \mathrm{E}\left[\theta_j X_{:j}\right] \\
&= \phi_{\boldsymbol{\theta}}(\mathbf{x}_i) - \mathrm{E}\left[\phi_{\boldsymbol{\theta}}(\mathbf{X})\right], \quad \forall\, i \in [1, \ldots, m]
\end{aligned} \tag{5.42}$$

The Shapley values allow to transpose this to any model. Let F be the set of all features, and $S \subset F$ all feature subsets. The Shapley value ζ_j assigns an importance value to each feature j that represents the effect on the model prediction of including that feature. To compute this effect, two models are trained, one with a subset S that does not include j, and one for subset $S \cup \{j\}$. The difference between the two is computed for all possible subsets that do not contain j as $\phi_{S\cup\{j\}}(\mathbf{x}_{S\cup\{j\}}) - \phi_S(\mathbf{x}_S)$. The Shapley values are defined as the weighted average of all these possible differences:

$$\zeta_j \triangleq \sum_{S \subseteq F \setminus \{j\}} w_m \cdot \left(\phi_{S\cup\{j\}}(\mathbf{x}_{S\cup\{j\}}) - \phi_S(\mathbf{x}_S)\right), \tag{5.43}$$

where the weight w_m corresponds to the fact that there are $|F|$ features to be analyzed, and for each choice of j, there are $\binom{|S|}{|F|-1}$ different subsets of size $|S|$:

$$w_m \triangleq \frac{|S|!\,(|F|-|S|-1)!}{|F|!} = \frac{1}{|F|}\frac{1}{\binom{|S|}{|F|-1}}, \tag{5.44}$$

where $|S|$ and $|F|$ are the cardinality of sets S and F.

The Shapley values are an **additive feature method**. Let h_x be a mapping funtion such that $\mathbf{x}' = h_x(\mathbf{x}')$ where $\mathbf{x}$ is the input and $\mathbf{x}'$ a binary vector (or simplified input). An additive feature attribution method has an explanation model that is a linear function of binary variables:

$$g(\mathbf{z}') = \zeta_0 - \sum_{i}^{n} \zeta_i z'_j, \tag{5.45}$$

where $\mathbf{z}' \in \{0,1\}^n$ with n the number of simplified input features and $\phi_j \in \mathbb{R}$.

Summing the effects of all feature attributions approximates the output $\phi(\mathbf{x})$ of the original model. An additive feature method verifies the following properties:

- Local accuracy: $g(\mathbf{x}') = \phi(\mathbf{x})$ when $\mathbf{x} = h_x(\mathbf{x})$
- Missingness: $x'_j = 0 \implies \zeta_j = 0$
- Consistency: Let $\phi_x(\mathbf{z}') = \phi(h_x(\mathbf{z}'))$ and $\mathbf{z}'_{-j} \iff \mathbf{z}'_j = 0$. For any two models ϕ and ϕ', if $\phi'_x(\mathbf{z}') - \phi'_x(\mathbf{z}'_{-j}) \geq \phi_x(\mathbf{z}') - \phi_x(\mathbf{z}'_{-j})$ then $\zeta_j(\phi', x) \geq \zeta_j(\phi, x)$.

In addition to the properties of additive feature attribution, Shapley values satisfies multiple properties such as additivity, symmetry, efficiency (feature contributions add up to the difference of prediction for x and the average), and dummy (a feature that does not contribute to the predicted value has a Shapley value of 0). They are detailed in [Sha51].

In practice, Shapley values are estimated instead of calculated exactly, as all possible coalitions (sets) of feature values would have to be evaluated with and without the j-th feature. Each time, the model would have to be retrained. An approximation with Monte-Carlo sampling has been proposed [ŠK14].

Shapley values guarantee that the prediction is fairly distributed among the features and offer contrastive explanations. Both of these advantages are not offered by LIME. However, they require a lot of computing time, do not provide sparse explanations, and are unable to predict the effect of a change in features on the model prediction. They also require access to the function and the dataset.

5.5.4 . SHAP

SHAP (SHapley Additive exPlanations) [LL17] is a method based on Shapley values, representing them as an additive feature attribution method, similar to LIME. Its Python implementation is used in this manuscript.

KernelSHAP is the first proposed implementation of the SHAP method. It is a kernel-based estimation approach for Shapley values inspired by local surrogate models. It offers a solution to Eq. (5.39),

consistent with the properties of local accuracy, missingness, and consistency of an additive feature method. The kernel is defined as follows:

$$\pi'_x(\mathbf{z}') \triangleq \frac{(|F|-1)}{\binom{|\mathbf{z}'||F|}{|S|}\,(|F|-|\mathbf{z}'|)}, \tag{5.46}$$

and should be used with $\Omega(g)=0$ and $C(\phi,g,\pi'_x)=\sum_{\mathbf{z}'}\pi'_x(\mathbf{z}')\left(\phi(h_x^{-1}(\mathbf{z}'))-g(\mathbf{z}')\right)^2$.

As with other interpretable methods, it quantifies the contribution or importance of each feature but does not assess the quality of the prediction itself.

SHAP offers all the advantages of Shapley values, such as its theoretical foundation, its contrastive explanations, and the fair distribution of prediction among the feature value. In addition to Shapley properties (Efficiency, Symmetry, Dummy, and Additivity), SHAP adds local accuracy, missingness, and consistency. While KernelSHAP is computationally expensive, a fast implementation is proposed for tree-based models, TreeSHAP [LEL19].

However, as with LIME, SHAP can be used to hide biases [SHJ+20]. Finally, KernelSHAP ignores feature dependence contrary to TreeSHAP, but TreeSHAP can produce unintuitive feature attributions.

6 - Kinematic lines prediction

Contents

As presented in Chapter 1 regarding the vibration environment, an aircraft is excited by several sources, such as the aerodynamic broadband excitations and the kinematic lines. Studying all of them at once can be a complex task, inducing problems of interpretation regarding the model used and its results. This chapter focuses on kinematic lines, especially those generated by engines. They are the more energetic, so equipment need to be defined relative to them. In addition, they allow the study of propagation through the structure, which helps extend predictions to non-instrumented points in the aircraft structure.

Firstly, the RMS values of an engine's kinematic lines are studied and reconstructed using a non-parametric model. The data are from ground tests, which allows the aerodynamic interactions to be ignored. This model is helpful for modeling the noise and injecting it into numerical models.

Secondly, a way is proposed to transfer this prediction from ground tests to airborne tests. It is a first step in the approximation of broadband levels and is based on kinematic lines RMS ratios, which can be seen as transfer functions or frequency response functions in our output space. By measuring them on the ground, they could be adjusted to flight phases and used to predict a non-instrumented point, reducing FTI (Flight Test Instrumentation) costs.

6.1 . Kinematic lines prediction during ground tests

The objective of this section is to train a model to reconstruct the local RMS value of kinematic lines depending on the source excitation. Local means it is inherently linked to a specific accelerometer, but the methodology can be applied to any instrumented point in the structure. After presenting how the dataset is generated, the prediction methodology is detailed. Then, results are analyzed for one accelerometer and extended to multiple ones.

6.1.1 . Generating a kinematic lines dataset from vibration survey tests

The generated dataset contains the kinematic lines frequency and RMS levels for the fundamental harmonic of the low-pressure turbine (also denoted N_1) of the left and right engines. It corresponds

to multiple survey tests performed for one aircraft. It is generated using the steps presented in Chapter 2, by first computing short PSDs of 1 second, with FFT windows of 2048 points[1]. Short PSDs are not conventional, but they allow to tackle variations in rotational speed at the cost of an increased variance in the spectra. Moreover, this limits the width of kinematic lines: if their frequency changes during the PSD computation, they spread out because of the averaging effect.

Kinematic lines frequencies are identified, and their RMS value is computed, as defined in Eq. (2.4). In this Chapter, they are simply denoted $\boldsymbol{\kappa}$, as the vector holds the RMS values for all frequencies of a given harmonic. Associated with the RMS values are the rotational speed of engines (in $\%\mathrm{rpm}$) as general parameters. An illustrative diagram of the extraction process is given in Figure 6.1.

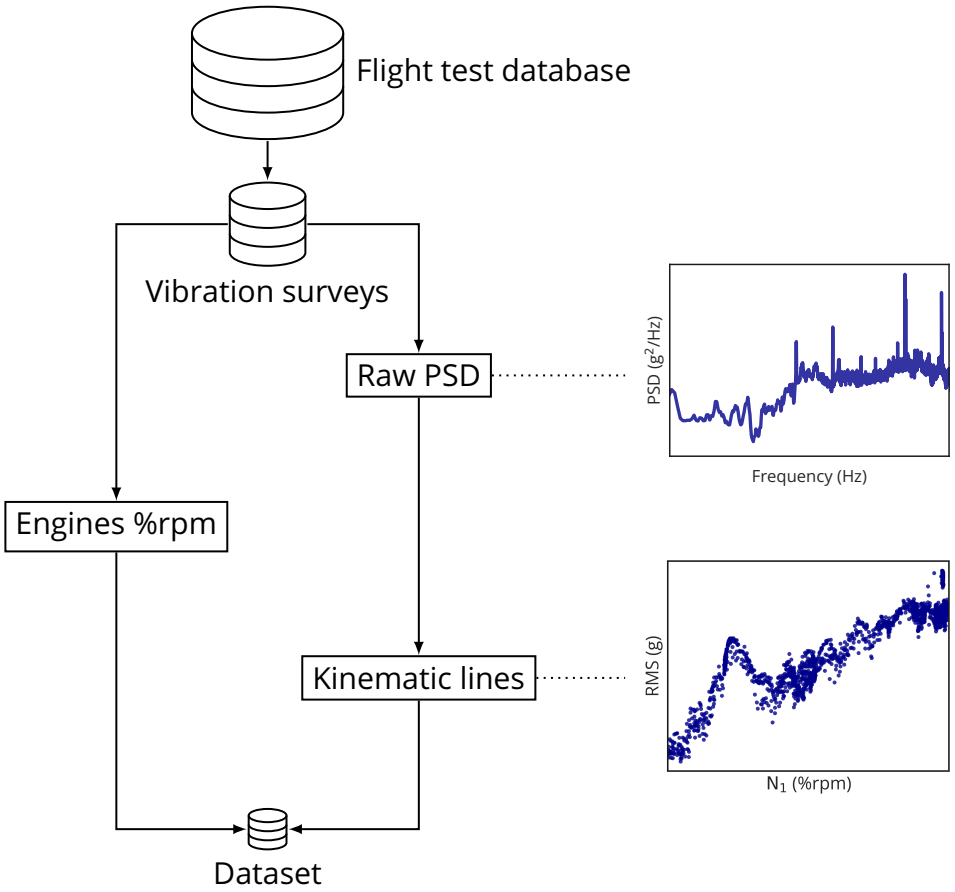


Figure 6.1: Illustration of the dataset generation process: RMS levels of kinematic lines are computed along their frequency and associated with engines rotational speed.

To consider only a single source, sequences where the left engine is up and running and the right engine is idling are used. They are illustrated in Figure 6.2, where the evolution of the low-pressure turbines through time is presented. The final dataset contains $m = 3162$ data points, each corresponding to the RMS value of a kinematic line and the associated engine regime over one second.

[1] This means that one PSD is averaging 8 FFT windows, as the overlap is set to 90%.

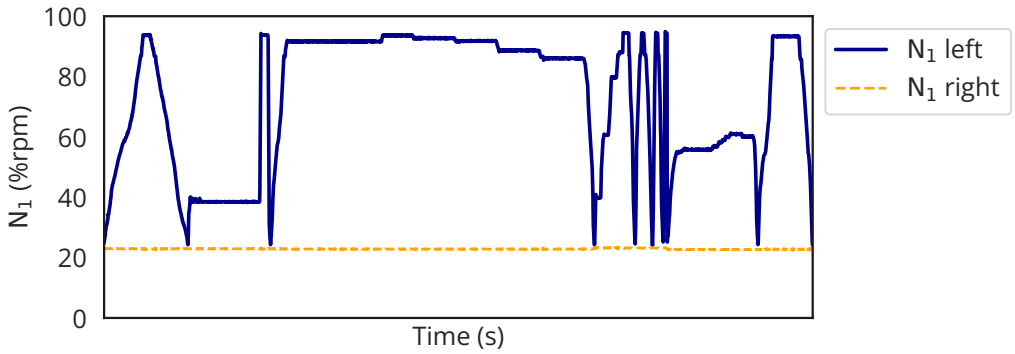


Figure 6.2: Evolution of the left and right N_1 through time.

Broadband levels are not removed so that the model can predict the actual levels measured on the ground. However, subtracting broadband spectra is a task rendered simple by the algorithm proposed in Section 4.1, and can be used if the goal is to use this model as a building block of a more general prediction model.

6.1.2 . Modeling RMS values of kinematic lines using a Gaussian process

In this regression problem, the goal is to predict the RMS value[2] of any kinematic line, which is a scalar value y. It corresponds to an element in $\boldsymbol{\kappa}$, of length m. To do so, the rotational speed of the left engine is given, also a scalar value, denoted x. It is equal to the frequency of the kinematic line up to a factor, see Eq. (2.3).

The model used is a Gaussian process, introduced in Section 5.2.4. As it is non-parametric, it is denoted ϕ. One model is used for each harmonic h and accelerometer i (more precisely an accelerometer channel, corresponding to an instrumented position in the aircraft and an axis). The modeling task, whether or not spectra separation is applied, can be written from Eq. (2.4) and Eq. (4.1) as:

$$\hat{y} = \phi(x) = \kappa_i^{(h)}(x) = \mathrm{RMS}\left((\boldsymbol{\eta}_{ik}^{(h)} \odot \boldsymbol{\delta}_k^{(h)} + \mathbf{b}_i)\,[f - \Delta f/2, f + \Delta f/2] \right), \tag{6.1}$$

where $\boldsymbol{\eta}_{ik}^{(h)}$ is the attenuation factor of harmonic h between the engine k and the measurement point i, $\boldsymbol{\delta}_k^{(h)}$ is the excitation of harmonic h generated at the engine k, $[f - \Delta f/2, f + \Delta f/2]$ is the frequency interval centered around the frequency corresponding to the rotational speed x, and $\mathbf{b}_i = \alpha_i \boldsymbol{\psi}_i$ is the energy due to the broadband, considered negligible here as the window is centered around the kinematic line. In this case, the equipment E and the turbine N that drives it are both the left engine. If spectra separation is applied, see Section 4.1.1 for details on the hypothesis, then $\mathbf{b}_i = \mathbf{0}$ in Eq. (6.1).

The Gaussian process is based on the following composite kernel:

$$k\left(\mathbf{x}, \mathbf{x}'\right) = k_{\mathsf{RBF}}\left(\mathbf{x}, \mathbf{x}'\right) \times k_{\mathsf{Constant}}\left(\mathbf{x}, \mathbf{x}'\right) + k_{\mathsf{White}}\left(\mathbf{x}, \mathbf{x}'\right). \tag{6.2}$$

The constant kernel is used here as no multiplicative constant is included in the RBF kernel definition, see Eq. (5.12), and the white noise kernel allows to take into account variability between tests.

[2] It is log transformed during the preprocessing step.

The evaluation methodology consists of splitting the dataset into a training set and a test set containing multiple engine vibration survey tests. One engine vibration survey test is used for the training[3] (324 points), and 3 engine vibration survey tests are used for the test set (2838 points). Samples are assumed to be i.i.d. (independent and identically distributed) with respect to the environmental conditions such as temperature and wind: this dataset split is performed to assess the model robustness to these conditions.

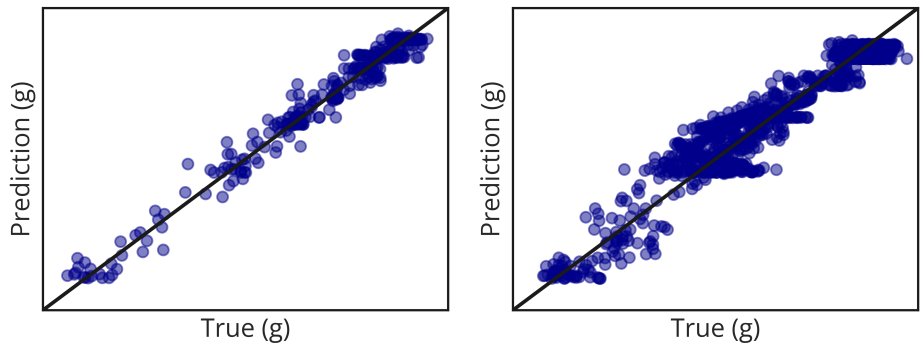
Such as model is however not supposed to be able to extrapolate in the frequency domain, as this depends on the structure through engine excitations $\boldsymbol{\delta}_k^{(h)}$ and attenuation factor $\boldsymbol{\eta}_{ik}^{(h)}$.

As a side note, another approach has been tested: decomposing spectra into NMF components, and predicting the coefficients values to be able to reconstruct a complete spectrum[4]. This approach was abandoned as the NMF reconstruction error adds up to the modeling error, and because here only the kinematic lines RMS levels need to be predicted, not complete PSDs.

6.1.3 . Results: RMS prediction and amplitude reconstruction

Results are detailed here and exhibit good generalization to unseen engine vibration survey tests. First, they are detailed for a specific accelerometer; then, they are presented for various accelerometers across the aircraft structure.

The predictions against true values are presented in Figure 6.3. The training dataset is captured well, and the residuals are well distributed. The noise in the data can be seen with the distribution of residuals around the unitary slope line. The test dataset exhibits some trend that could maybe be captured by adding another RBF kernel, but is otherwise well distributed.



(a) Training set, $R^2 = 0.97$ (b) Test set, $R^2 = 0.89$

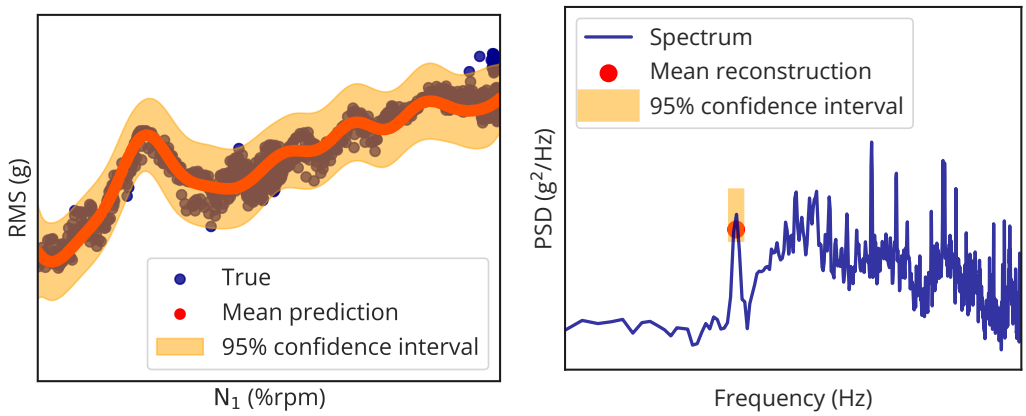
Figure 6.3: Predictions against true values for training set and test set, with a unitary slope line. The coefficient of determination R^2 denotes the square of Pearson's correlation.

Figure 6.4 shows the actual predictions against the true values, for the test set. In Figure 6.4a, the RMS mean prediction and a 95% confidence interval are plotted against the measured RMS values. The trend is well captured, and the confidence interval contains all measurements with the exception of outliers at the maximum rotational speed. In Figure 6.4b, a snapshot of amplitude reconstruction

[3]The training set is used to find the hyperparameters using k-fold cross validation.

[4]In practice, one Gaussian process is used per component in this alternative approach.

for a given rotation speed is provided, and superposed to the original PSD. The amplitude reconstruction is performed by assuming a squared kinematic line with a width of β Hz. The frequency of the reconstructed value is perfectly aligned with the frequency of the actual kinematic line, and the confidence interval includes the peak, as required.



(a) RMS prediction (b) Amplitude reconstruction

Figure 6.4: Prediction of RMS values in the test set using a log scale on the y-axis (left), and amplitude reconstruction on a spectrum using a log-log scale (right).

This method is evaluated on 24 accelerometer channels corresponding to 8 tri-axial accelerometers. They are located on the left engine mounts, a critical spot where the vibrations flow from the engine to the structure. This is done to see how well the method generalizes to different accelerometers, since they exhibit different responses depending on their position in the aircraft structure.

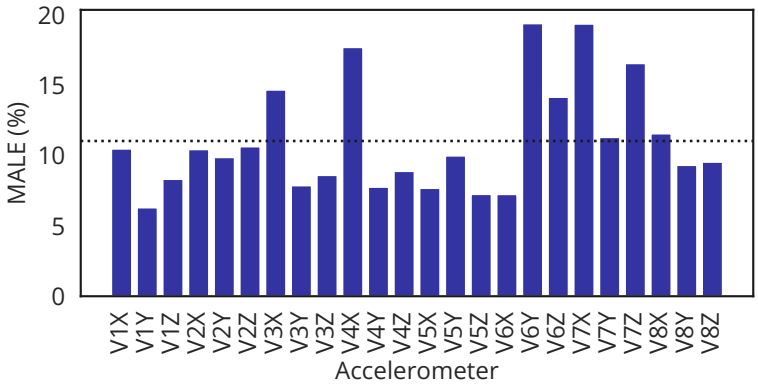


Figure 6.5: Average prediction error for various accelerometer channels on the left engine mount. The dotted line is the mean across accelerometers.

Results are reported in Figure 6.5, where the average prediction error on the test set is reported for accelerometer channels on the left engine mount. The metric used is the MALE, see Eq. (5.34), which gives a percentage of order of magnitude error. Across accelerometer channels, the mean error is

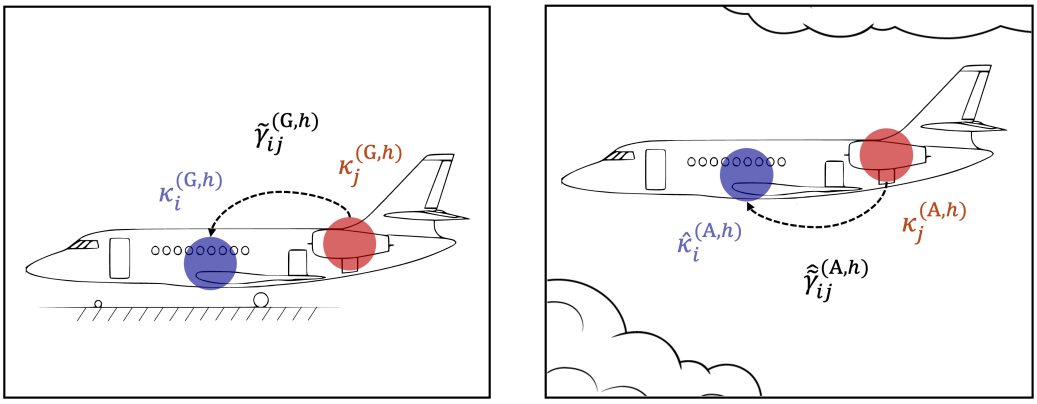
11% with a standard deviation of less than 4%. This is a good prediction performance, considering the noise present in the data and the fact that only the mean prediction is considered, not the confidence interval.

All the results previously shown are given for the fundamental kinematic line of the left engine. They were also assessed for other harmonics, and showed comparable results for visible harmonics.

To further develop this model, it could be interesting to see how well it generalizes between aircraft of the same type. This would help to quantify the variability in production quality. Doing this on test aircraft is projected to give significant differences.

6.2 . Transposition from ground test to flight test

In the previous section, the prediction of the value of a kinematic line at any instrumented point in the structure while the aircraft is on the ground is presented. An illustration is provided in Figure 6.6, showcasing the notations used in this section. Here, two concepts are detailed: how to use a **Kinematic Lines RMS Ratio** (abbreviated here as KLR) to displace an RMS value between instrumented points in the structure, and how to use a **transposition function** to adjust a KLR so that it models a flying aircraft. By measuring, on the ground, the RMS values of kinematic lines at any points i and j in the structure, and by having access to the RMS values at point j during flight, the objective is to reconstruct the values at point i by applying the KLR adjusted by the transposition function between i and j.



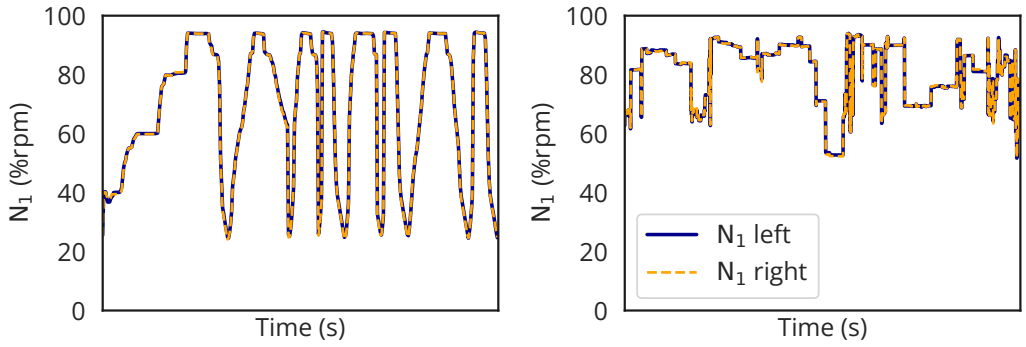
(a) Ground test (b) Flight test

Figure 6.6: Illustration of the transposition from ground to flight between points i and j.

6.2.1 . Methodology: KLR and transposition function

The dataset used for this section is similar to the one used previously in Section 6.1, and its extraction process is the same as in Figure 6.1. Different accelerometers are considered: they are spread across the aircraft, see Figure 6.8 where the 7 triaxial accelerometers are presented, including VMOT, which is in the left engine.

In contrast to the previous section, both engines are synchronized. This is done to provide a ground vibration surveys dataset more consistent with the flight test dataset. The final dataset is



(a) Ground test data

(b) Flight test data

Figure 6.7: Evolution of the left and right N_1 through time, for ground (left) and flight test data (right).

comprised of $m = 47360$ data points, including 1584 data points of ground engine vibration survey tests and 45 776 points of flight tests.

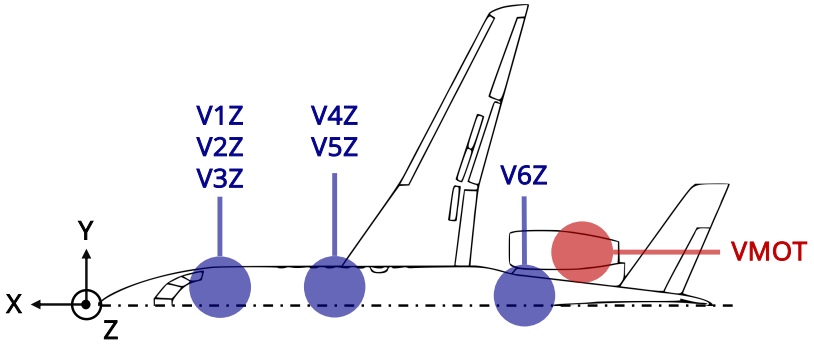


Figure 6.8: Position of the accelerometer channels considered in this study (NB: a left-handed coordinate system is used).

First, let us define the **KLR** γ_{ij} between two points i and j: it is the ratio between the RMS value $\boldsymbol{\kappa}_i$ at point i and the RMS value $\boldsymbol{\kappa}_j$ at point j, see Eq. (2.4). $\boldsymbol{\kappa}$ and $\boldsymbol{\gamma}$ are vectors where each component corresponds to a frequency. When this ratio is calculated on harmonic h using ground test data, it is denoted with a superscript (G, h):

$$\boldsymbol{\gamma}_{ij}^{(\mathrm{G},h)} \triangleq \frac{\boldsymbol{\kappa}_i^{(\mathrm{G},h)}}{\boldsymbol{\kappa}_j^{(\mathrm{G},h)}}. \tag{6.3}$$

The KLR allows to compute the value of a point in the structure by knowing the value of another point in the structure. The system is assumed to be linear and time-invariant.

Note that the KLR $\boldsymbol{\gamma}$ should not be confused with the attenuation factor $\boldsymbol{\eta}$. While $\boldsymbol{\eta}$ allows to displace a PSD engine excitation to another point in the structure, $\boldsymbol{\gamma}$ allows to transposes any RMS excitation between points in the structure. The difference can be explicited by using the same modeling as in Eq. (6.1), which is based on the broadband hypothesis detailed in Eq. (4.1). Let us consider

i and j two points in the structure, and k is the source generating the kinematic line considered. The relationship between γ at a given frequency f and $\boldsymbol{\eta}$ can be written as:

$$\gamma_{ij}^{(h)}(f) = \frac{\mathrm{RMS}\left((\boldsymbol{\eta}_{ik}^{(h)} \odot \boldsymbol{\delta}_k^{(h)} + \mathbf{b}_i)\,[f - \Delta f/2, f + \Delta f/2]\right)}{\mathrm{RMS}\left((\boldsymbol{\eta}_{jk}^{(h)} \odot \boldsymbol{\delta}_k^{(h)} + \mathbf{b}_j)\,[f - \Delta f/2, f + \Delta f/2]\right)}, \tag{6.4}$$

where $\boldsymbol{\delta}_k$ is the excitation at the source k, $\boldsymbol{\eta}_{ik}$ is the attenuation factor between the source k and the measured point i, and $\mathbf{b}_i$ is the broadband energy, considered negligible as the RMS window is centered around the kinematic line. The RMS, see Eq. (1.12), is applied on the interval $[f - \Delta f/2, f + \Delta f/2]$ centered around f.

For this dataset, k denotes the engines, and as the ratios are computed between the left engine (measured by VMOT) and the other 6 accelerometers, $j \approx k$.

In practice, to avoid modeling measurement error, a smoothed-out version of the KLR $\boldsymbol{\gamma}_{ij}$ is used. While it introduces an irreducible error to the model performances, it allows to generalize better. This is done by performing a moving average, and the result is denoted as $\widetilde{\boldsymbol{\gamma}}_{ij}$.

While this formalism holds for a given harmonic h, the results section shows that the concatenation of KLRs for different harmonics superpose well. In Section 8.1, this representation is presented for different accelerometers, showing that it can provide information about structural attenuation.

Second, let us define a **transposition function** defined through a model $\boldsymbol{\phi}_{\boldsymbol{\theta}}$, allowing to transpose a KLR $\boldsymbol{\gamma}_{ij}^{(\mathsf{G})}$ to flight conditions. The RMS values at a point i can be written as:

$$\hat{\boldsymbol{\kappa}}_i^{(\mathsf{A})} = \hat{\widetilde{\boldsymbol{\gamma}}}_{ij}^{(\mathsf{A})} \boldsymbol{\kappa}_j^{(\mathsf{A})}, \tag{6.5}$$

where the superscript (A) denotes airborne data, $\widetilde{\cdot}$ denotes a smoothed-out KLR and $\hat{\cdot}$ denotes a prediction resulting from the transposition function.

By domain knowledge, the transposition function $\boldsymbol{\phi}_{\boldsymbol{\theta}}$ is assumed to be dependent on the flight point, the distance to the source (here the engines), and the frequency. As a first attempt to answer the problem at hand, a simplified model is proposed. It is based on the longitudinal distance d_{ik} of i to the source k, the dynamic pressure p_{dyn}, and the frequency f:

$$\hat{\widetilde{\boldsymbol{\gamma}}}_{ij}^{(\mathsf{A})} = \boldsymbol{\phi}_{\boldsymbol{\theta}}\left(\widetilde{\boldsymbol{\gamma}}_{ij}^{(\mathsf{G})}\right) = \left(1 + \theta \times p_{\mathsf{dyn}} \times d_{ik} \times \boldsymbol{\sigma}\right) \odot \widetilde{\boldsymbol{\gamma}}_{ij}^{(\mathsf{G})}, \tag{6.6}$$

where $\boldsymbol{\sigma}$ is the sigmoid function depending on the frequency. It acts as a smooth boolean step function and allows to apply $\boldsymbol{\phi}_{\boldsymbol{\theta}}$ only if $f > 300$ Hz. In Section 6.2.2, a more physically-grounded alternative is proposed.

6.2.2 . Results and discussion on transposition performance

The smoothing of the KLRs $\widetilde{\boldsymbol{\gamma}}_{ij}$ introduces an error in the reconstruction. This error can be modeled as an additive Gaussian noise, and is originating from the measurements and acquisition procedure, in addition to natural variability. In Figure 6.9, the KLRs $\boldsymbol{\gamma}$ of several harmonics are superposed with their smoothed-out version $\widetilde{\boldsymbol{\gamma}}$. An interesting result is how well the smooth KLRs of different harmonics superpose, giving information about structural attenuation.

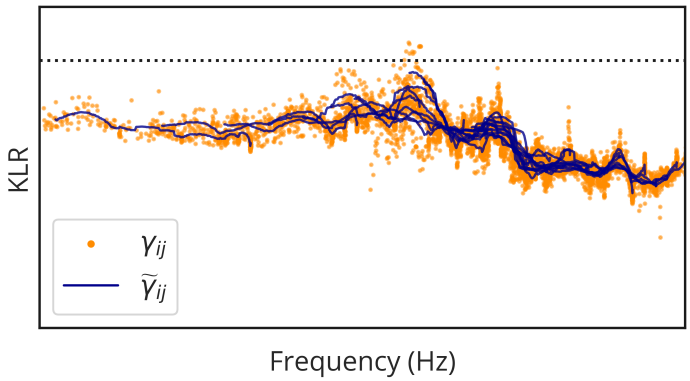


Figure 6.9: KLRs and smoothed-out KLRs from j = VMOT to i = V4Z, for several harmonics, log-log scale.

The smoothed-out KLR $\widetilde{\gamma}$ is then used to reconstruct RMS values, and superposed with the actual measurements in Figure 6.10. The result is a 10.7% reconstruction error across the 6 accelerometers in the structure, which is coherent with the expected noise present in the data.

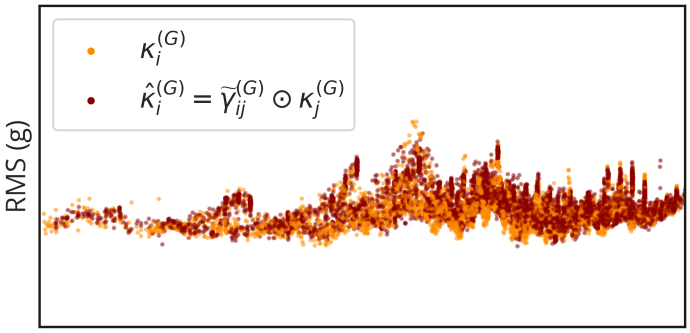


Figure 6.10: Reconstruction of $\boldsymbol{\kappa}_{\text{V4Z}}^{(\text{G})}$ using the smoothed-out KLRs from j = VMOT to i = V4Z, for several harmonics, log-log scale.

After estimating the irreducible noise error, the transposition from ground to airborne can be performed. In Figure 6.11, the value of $\boldsymbol{\kappa}_i^{(\text{A})}$ is predicted at V4Z using the transposition function applied on the ground KLR, and superposed to the measurements. While noisy, the prediction captures well the measurement trend. For low frequencies, the transposition function has no effect, and prediction is close to measurements. For higher frequencies, the transposition allows to correct the predicted $\hat{\boldsymbol{\kappa}}_i^{(\text{A})}$ from the gray points (using $\widetilde{\gamma}_{ij}^{(\text{G})}$) to the orange points (using $\hat{\widetilde{\gamma}}_{ij}^{(\text{A})}$), closer to the blue points (measurements).

Across all 6 accelerometers, the transposition error averages at 30.4% using $\hat{\widetilde{\gamma}}^{(\text{A})}$. If no transposition function is used and the ground KLR $\widetilde{\gamma}^{(\text{G})}$ is used, the error increases to 49.9%.

To improve the methodology, focusing on the transposition function is the priority, through three

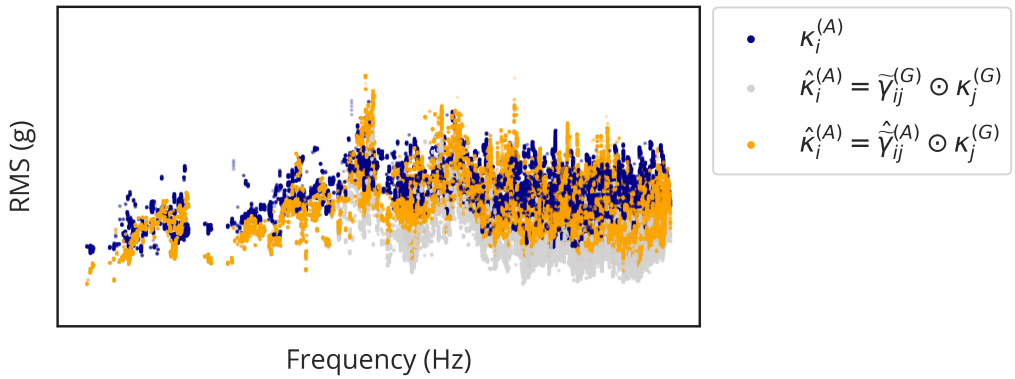


Figure 6.11: Prediction of $\boldsymbol{\kappa}_i^{(\mathrm{A})}$ at V4Z using the transposition function on the smoothed-out KLRs from $j =$ VMOT to $i =$ V4Z, for several harmonics, log-log scale.

aspects. First, a model similar to the one presented in Chapter 7 could be used to represent the aerodynamic effect, instead of the single dependence to the dynamic pressure. Second, to replace the distance to the source, a representation of the structure should be used. This is the subject of Part III. Third, the reverberation effect of the ground needs to be taken into account. The adjusted could then be written:

$$\hat{\boldsymbol{\kappa}}_i^{(\mathrm{A})} = \tilde{\boldsymbol{\gamma}}_{ij}^{(\mathrm{G})} \odot \boldsymbol{\phi}_{\text{Reverberation}} \odot \boldsymbol{\kappa}_j^{(\mathrm{A})} + \boldsymbol{\phi}_{\text{Aero}}, \tag{6.7}$$

with $\boldsymbol{\phi}_.$ the effects to model.

Other adjustments are to:

- Use a point on the left engine mount instead of the left engine itself for j (to bypass the complex effect of engine mount and directly consider the input at the structure)
- Consider the interaction between left and right engines, and be able to handle multiple sources
- Consider the energy across all 3 axes of accelerometers instead of only the Z-axis.

Moreover, the broadband levels have not been removed from the dataset: the model can be used to predict the actual levels measured on the ground. However, it is possible to apply the separation algorithm detailed in Section 4.1.2 before computing the kinematic lines energy. Then, this could be combined with a broadband level to reconstruct complete spectra, in flight for example. The following sections detail models that could be used to this end.

A perspective of this method is to compute the KLR from j to i while i is not instrumented. A transfer function can be estimated between two points that are not instrumented using an impact hammer test, as presented in Section 1.2.3. If a KLR can be estimated this way, then it would be possible to predict the RMS values at any non-instrumented point i during flight by using: (i) the value of instrumented point j during the flight, (ii) the KLR estimated on the ground, and (iii) the transposition function.

Finally, this method could be based on numerical models at the design stage and iteratively include experimental data when the first tests are performed. The value of instrumented point j during the flight can be first provided by the engine manufacturer. The attenuation factor between the engines and the considered point can be estimated using a numerical model, as for the transposition function.

6.3 . Conclusion on the prediction of kinematic lines

This chapter offers solutions for predicting kinematic lines generated by engine(s), allowing to model excitations generated by a localized source. A Gaussian process is used to predict the values during ground tests. The error of the mean prediction is at 11% on average across accelerometers, and the noise is in the 95% confidence interval, showing robustness to environmental test conditions. A study on aircraft of the same model can be performed to assess production variability. This model can be used to inject values into numerical models.

The second idea of this chapter is to propose a methodology to transpose the RMS values of kinematic lines from ground to flight tests. To this end, KLRs (Kinematic Lines RMS Ratios) are introduced, and can be seen as transfer functions or as normalized levels depending on the end use. They are used again in Section 8.1, and provide information on the structure. In addition to KLRs, a transposition function is introduced. While it is primarily a methodological section, an actual implementation is provided as an example and offers better results than without transposition function. By using a finer model, this method could enable the prediction of non-instrumented points during flight tests.

7 - Broadband levels prediction

Contents

In the previous chapter, the problem of predicting excitations generated by a single source has been addressed. Here, the vibration environment prediction problem is taken from the other side: all sources are considered including aerodynamic ones, and the objective is to predict broadband levels from general parameters (altitude, speed, high-lift device positions, engines rotational speed, etc.) at any flight point including stabilized flight phases, climbs and descents, and maneuvers.

Stabilized flight phases are the focus of the first section, based on work published in [FNM+23a]. Then, the methodology is extended to all flight phases in the second section, based on work presented in [FNM+23b]. Multiple models are compared, and model complexity is justified by improved performances. The questions of generalization ability and interpretability are also tackled.

The formalism of this chapter is the following. The objective is to predict an output $\hat{\mathbf{y}} \in \mathcal{Y} \subset \mathbb{R}^d$ from an input $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$, using a model $\boldsymbol{\phi}_{\boldsymbol{\theta}} \in \Phi$ parameterized by $\boldsymbol{\theta} \in \Theta$. The dimension of the output space remains at $d = 17$ throughout this chapter, while the dimension of the input space depends on the number of general parameters considered.

7.1 . Prediction during stabilized flight phases

Prediction of stabilized flight phases is the first problem tackled in this chapter. The work presented in this section has been published [FNM+23a] and is adapted here to fit the manuscript.

7.1.1 . Dataset

In what follows, data from the flight-test campaign of a civil jet aircraft is considered. Twenty-three flight tests are used, which amounts to 65 hours of flight. Each flight test produces a large number of time series, each of which is associated with a sensor. Among the considered sensors, vibration environment accelerometers are sampled at 5000 Hz, and the dataset to be handled is therefore very large.

The aircraft does not spend equal time at all flight points, making the measurement sparse along the flight domain. Moreover, measurements are contaminated with noise and possibly faulty sensors. The first step of data processing therefore consists of identifying usable sequences and discarding unreliable data, based on the work presented in Section 2.3.1.

A focus is made on the measurements along the vertical axis of 5 triaxial accelerometers, represented in Figure 7.1. This set is chosen to be representative of the whole aircraft as these sensors are known to be influenced by different vibration sources: V4Z and V5Z are heavily influenced by the engines, while V1Z and V2Z are influenced mainly by aerodynamic field interaction, V3Z being a middle ground influenced by both.

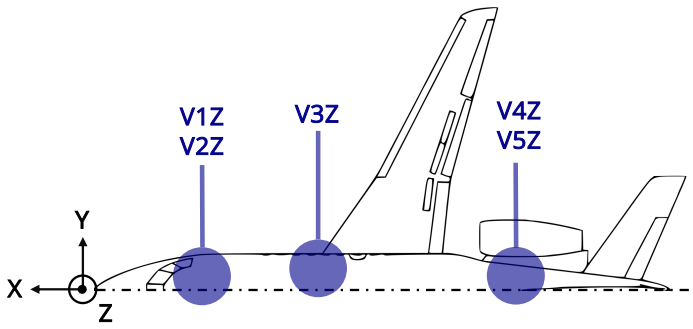


Figure 7.1: Position of the 5 accelerometer channels considered in this study (NB: a left-handed coordinate system is used).

The preprocessing pipeline is presented in Figure 7.2, and allows to produce a sufficiently diverse yet compact dataset to train a regression model.

The approach is focused on stabilized flight phases, where the aircraft conditions can be considered steady, which eases flight phase identification and avoids non-stationary phases. A total of 342 sequences, accounting for 7.5 flight hours, are identified based on a set of logical rules presented in Section 2.3.2.

This allows to perform PSD estimation, followed by signal separation. For each sequence, sliding PSDs of 20 seconds are computed with a step of 1 second, using FFT windows of 8192 points with a 90% overlap. Only the broadband energy is considered and the kinematic lines are removed using the algorithm presented in Section 4.1.2. A broadband spectrum is thus defined as:

$$\mathbf{b} = \alpha\boldsymbol{\psi}, \tag{7.1}$$

with α the broadband amplitude factor and $\boldsymbol{\psi}$ the background spectrum.

Smoothed-out spectra are then compressed by expressing them as RMS levels per frequency band, which is consistent with how vibration specifications are generally given. They are denoted as

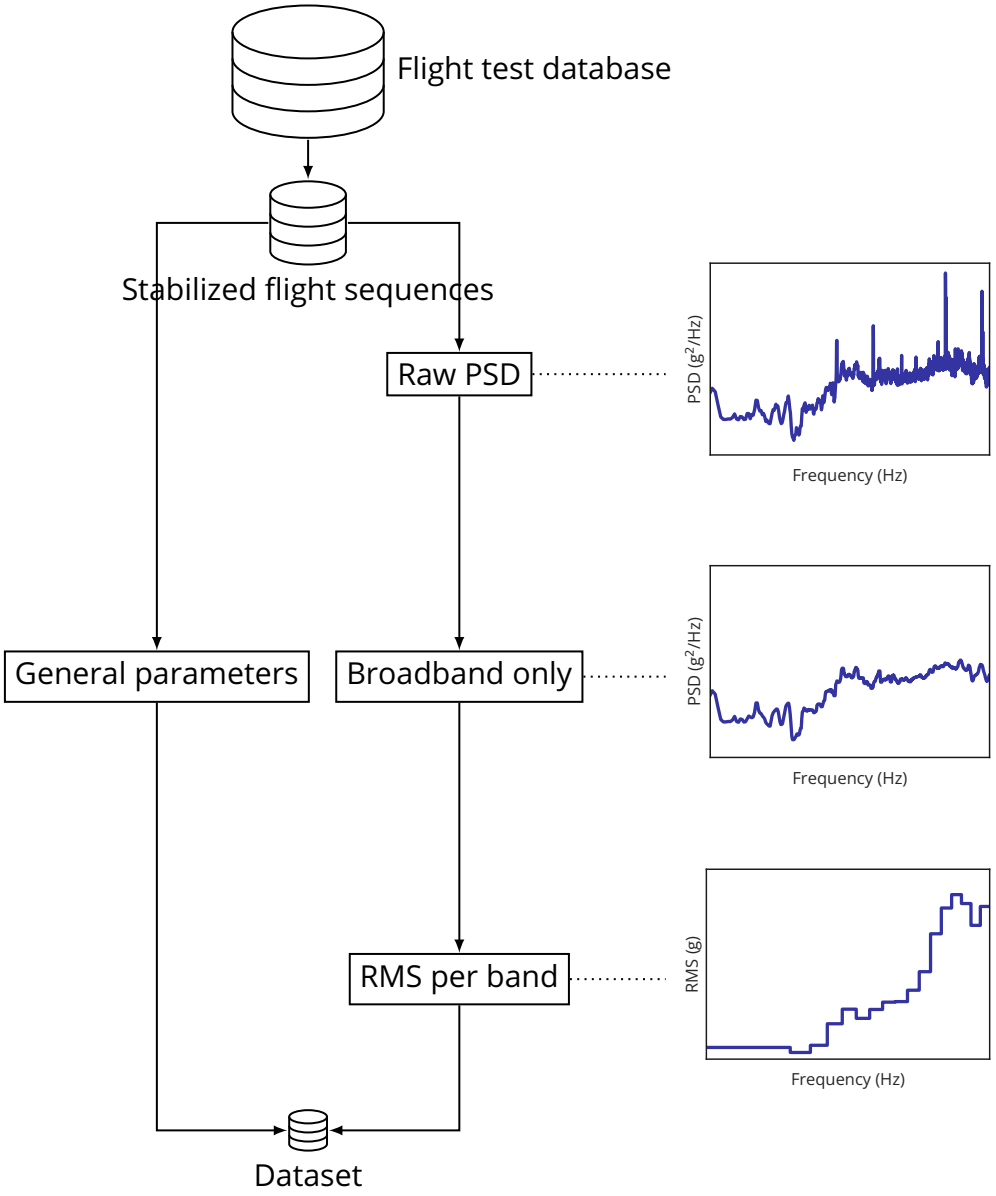


Figure 7.2: Diagram of the preprocessing pipeline, generating a dataset from the flight test database.

$\widetilde{\mathbf{b}}$, as they are a reduced representation of $\mathbf{b}$. A total of $d = 17$ frequency bands are used, as presented previously in Section 4.2.2. A sensitivity study of models error to the width of RMS bands is given in Section 7.1.7.

Finally, as spectra can take values spreading over several orders of magnitudes, they are log-transformed, see Eq. (3.3). A min-max scaler is then applied, which constrains the data to lie in the $[0, 1]$ range using the min and max values across all $i = 1 \dots m$ samples, see Eq. (3.2).

The compression is such that 20 s of temporal data sampled at 5000 Hz are compressed into a frequency representation of $d = 17$ scalars, amounting to 0.02% of the original size.

7.1.2 . Features selection process

From the generated dataset, the goal is to train a predictive model to estimate spectra $\mathbf{y}$ from an input $\mathbf{x}$, containing a set of n general parameters or *features*. A constraint is that the model should be agnostic to equipment position in the aircraft, meaning that the same n features must be used to make prediction whether or not the same set of sources is influential. This allows to quickly train a new model without having to change the architecture.

Among all the monitored parameters, roughly 50 are pre-selected via domain knowledge since most parameters are unrelated to the vibration environment. From this set of 50 parameters, only $n = 13$ of them are kept by removing low variance, redundant, and non-influential parameters. They express the state of the aircraft relative to several categories: attitude P_{A_1}, engines $P_{\mathsf{E}_{1\ldots3}}$, flight point $P_{\mathsf{FP}_{1\ldots3}}$, configuration $P_{\mathsf{C}_{1\ldots6}}$.

The selection is done by studying the correlations between features and frequency bands, and by performing clustering. Such a study is presented in Section 4.3. Several features show weak correlations with the outputs, but are yet retained through domain knowledge. This is because they are deemed to have a certain predictive power not encapsulated by the correlation value. Before using the selected set of features, each feature vector $\mathbf{x}$ is standardized using the empirical mean μ and standard deviation σ across all m data points, see Eq. (3.1), which allows to ease the training process by forcing all variables to be in the same range of values.

7.1.3 . Model implementation

The *multi-output multivariate regression* problem at hand can be written as follows:

$$\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}) = \widetilde{\mathbf{b}} = \hat{\mathbf{y}}. \tag{7.2}$$

Two baseline models $\boldsymbol{\phi}_{\boldsymbol{\theta}}$ are implemented: linear regression, see Section 5.2.1, and XGBoost, see Section 5.2.3. They are compared against an ANN ensemble, see Sections 5.2.5 and 5.2.7. Higher order regression was also tested, but showed to give significantly worse results not detailed here.

Each ANN of the ensemble has the same architecture. To ease the hyperparameters optimization process, each hidden layer is set to the same structure. Model stability is fostered by using batch normalization, i.e., centering and rescaling of layers' weights. Each layer performs the following transformations:

$$\text{Dense} \rightarrow \text{BatchNormalization} \rightarrow \text{Activation}.$$

In order to select the optimal configuration, tests were carried out for architectures of 2 to 6 layers. The final architecture is made of 3 hidden layers, each of them having 30 neurons and using a ReLU (Rectified Linear Unit) as an activation function.

A set of 8 models is used for the ensemble, with identical hyperparameters but different initial conditions. They are trained on the same dataset for 5000 epochs, each with a batch size of 32, using an Adam optimizer.

The prediction of the ANN ensemble is considered to be a Gaussian distribution parameterized by a mean μ and standard deviation σ over all predictions $\hat{\mathbf{y}}$ of the 8 networks:

$$\boldsymbol{\phi}(\mathbf{x}) \sim \mathcal{N}(\mu(\hat{\mathbf{y}}), \sigma(\hat{\mathbf{y}})). \tag{7.3}$$

Standard deviation measures the degree of confidence in predictions, as large values indicate that models give different predictions that should be considered carefully.

To find the hyperparameters, Bayesian optimization is first performed with the Python implementation from Optuna [ASY+19] to identify a region of interest in the hyperparameters space, where a grid search is then conducted. To use the available data efficiently and limit information leakage, k-fold cross-validation is used, with $k = 5$ folds. To have a non-biased estimate of model performance, a final test is performed on a separate test set, yet unseen and never used for the optimization.

7.1.4 . Evaluation methodology

For each data point i of dimension d, model predictive performances are assessed using the MALE metric defined in Eq. (5.34), where a 100% error amounts to a 1-order-of-magnitude error. Note that the data is already log-transformed, so the log transformation is not applied in the metric. Only results for the unseen test data are presented, which are data points that are not used for model training. Two protocols that correspond to different ways of splitting the dataset are used. For each protocol, all models are trained and evaluated on the same data.

The first protocol corresponds to a standard random splitting of the dataset (constituted of $m = 9430$ data points), where 90% of the data is used for training ($m_{\text{train}} = 8487$ data points) and 10% for test ($m_{\text{test}} = 943$ data points). The goal is to assess whether the model is expressive enough to represent the data.

The second protocol aims to evaluate the physical consistency of the model, through its ability to model the dynamic pressure. The dynamic pressure, here labeled P_{FP_1}, is directly linked with the broadband aerodynamic loads on the aircraft; the RMS levels are expected to increase linearly with it, all other things being equal. In order to check this, the training dataset in the second protocol contains only samples with a dynamic pressure value lower than a given threshold, and the model is tested on the remaining data points, corresponding to a higher dynamic pressure. A threshold of $p_{\text{dyn}} = 100$ hPa is chosen, which yields respective values for the training and the test dataset of $m_{\text{train}} = 7232$ and $m_{\text{test}} = 2198$. Figure 7.3 shows the train-test split for the dynamic pressure and another flight point-related parameter.

This second protocol may initially seem unusual, as no i.i.d. hypothesis holds for the training and test sets. It is however based on domain knowledge, as a linear dependence on this variable is expected. Thus, such a dataset split allows to specifically evaluate the ability of the model to learn this relationship, as the training set outputs corresponding to a low dynamic pressure – are influenced by many other variables, and the test set outputs mainly depend on the dynamic pressure.

7.1.5 . Results of the classical evaluation protocol

The predictive performances of each model for the classical evaluation protocols are detailed here. First, an example of prediction for a specific data point is presented, then the error distribution over frequency bands for a specific accelerometer, and then the global results for all accelerometers.

7.1.5.1 . Prediction example for one flight point

First, one flight point randomly taken from the test set is considered to compare the performance of each model. Figure 7.4 shows a typical output, i.e., one spectrum prediction for an input data point, associated with the sensor V3Z which is in the middle of the aircraft, thus representative of equipment

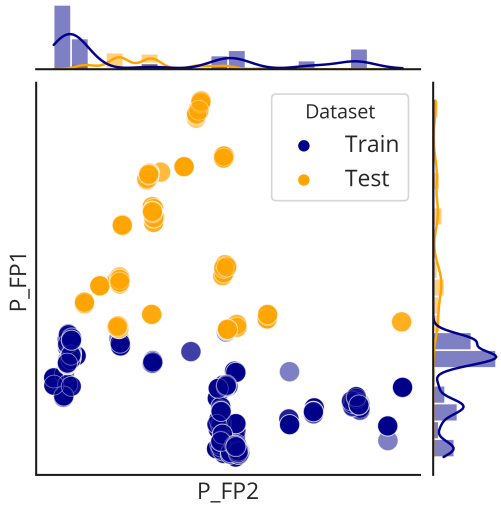
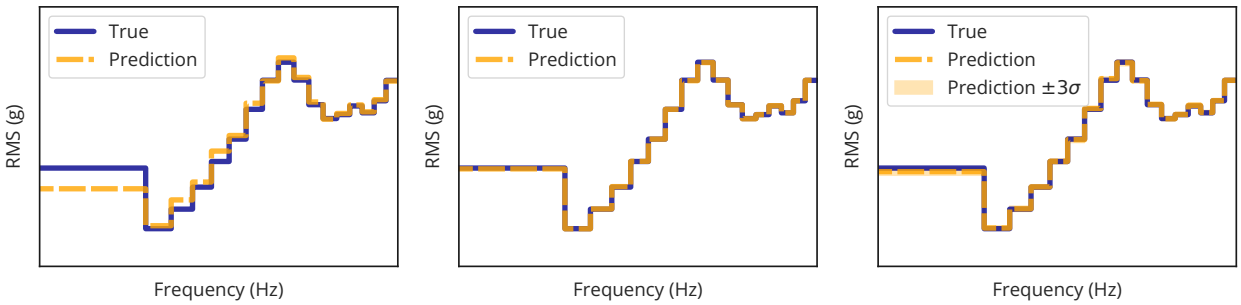


Figure 7.3: Scatter plot with marginal distribution of flight point parameters P_{FP_1} and P_{FP_2} using the extrapolation protocol.

subjected to various sources. In the case of the ensemble model, a confidence interval corresponding to 3 standard deviations is also provided. All predictions fit almost perfectly the measurements, with some little variability. In this specific example, the linear regression underestimates the first frequency band, which is highly influenced by aircraft configuration and thus hard to predict through a linear regression.



(a) Linear regression (b) XGBoost (c) ANN ensemble

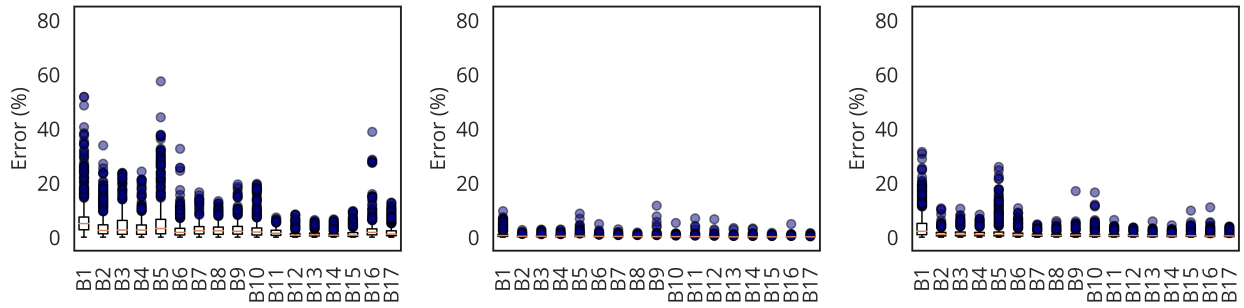
Figure 7.4: Predicted spectrum of a given data point for the accelerometer V3Z, classical evaluation, using a log-log scale.

7.1.5.2 . Error distribution over frequency bands for an accelerometer

To better characterize models errors, the mean error over each frequency band j is considered over all m data points i.

Figure 7.5 shows the boxplot of this error for each frequency band on the accelerometer V3Z across all three models. It is first noticeable that all models have good performances, the end of

distributions[1] being consistently under 20%, and even less most of the time. However, many outliers can be seen, which correspond to under-represented flight points. Even though the outliers are far from the distribution, these errors are acceptable for the XGBoost and ANN ensemble models. Finally, The bands B_1 and B_5 are consistently harder to predict, which is notable for the linear regression and ANN ensemble.



(a) Linear regression (b) XGBoost (c) ANN ensemble

Figure 7.5: Boxplot of the error per band for the accelerometer V3Z, classical evaluation.

7.1.5.3 . Global error

Finally, the overall predictive performance of each model is summarized in Table 7.1. For each accelerometer, the average of the MALE is computed across every test data point and frequency band, and the last column provides the average across all 5 accelerometers.

	Accelerometer					
	V1Z	V2Z	V3Z	V4Z	V5Z	Mean
Linear regression (%)	3.9	5.1	2.8	2.6	3.1	3.5
XGBoost (%)	**0.3**	**0.4**	**0.3**	**0.3**	**0.4**	**0.3**
ANN ensemble (%)	1.3	1.4	1.1	0.9	1.3	1.2

Table 7.1: Mean error for each accelerometer depending on the model, classical evaluation. Bold denotes the best results for a given accelerometer; the lower the better.

XGBoost offers the best performances with this test protocol, followed by the ANN ensemble and then the linear regression. The error is relatively similar across all the accelerometers, even though V2Z at the front of the aircraft seems slightly more challenging to predict. Since engines are located at the aircraft's rear, this sensor is mainly influenced by the aerodynamic field and the flight configuration.

[1] Ends of distributions are defined using the first quartile (Q1), the third quartile (Q3), and the interquartile range (IQR). The lower end of the distribution is defined as the largest value that is less than or equal to Q1 - 1.5 $\times$ IQR, and the higher end of the distribution is defined as the smallest value that is greater than or equal to Q3 + 1.5 $\times$ IQR. Any value outside of this range is considered to be an outlier.

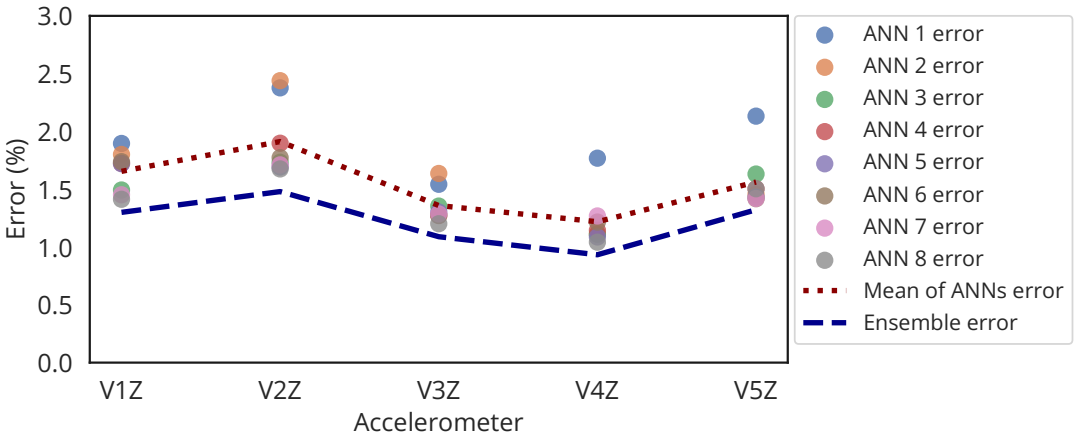


Figure 7.6: Mean error for each accelerometer depending on the neural network, classical evaluation.

The performance of each ANN of the ensemble is given in Figure 7.6. The variability between ANNs depends on the sensor, but is relatively low in this case. As expected, the ensemble error is lower than the average error of all the ANNs by approximately 30% on average. Here, the ensemble error is even lower than any single ANN at all times. By studying the confidence interval of predictions, it is found that 93.2% of predictions are within a $\pm 3\sigma$ interval, and 96.9% of predictions are below $\mu + 3\sigma$, which is deemed to be conservative. A large variability between individual predictors indicates that the flight point is difficult to predict for these models and that the predicted values should be taken carefully. Uncertainty quantification can be obtained with a single gradient boosting model [MPU21], but was not used here.

The good performances of all models show that the set of inputs is sufficient to predict the outputs. In fact, as the measurement error can be roughly estimated to be around 5 to 10%, taking into account the whole data acquisition system, it is likely that overfitting is present. Learning a dataset without generalization ability can be useful to generate spectra instead of querying a database. This can be used for anomaly detection, where the validity of a new measurement can be determined by comparing it with its predicted value in the model. In any case, overfitting is not possible in the second evaluation protocol, as the training set and the test sets are disjoint in parameter space.

7.1.6 . Results for the extrapolation evaluation protocol

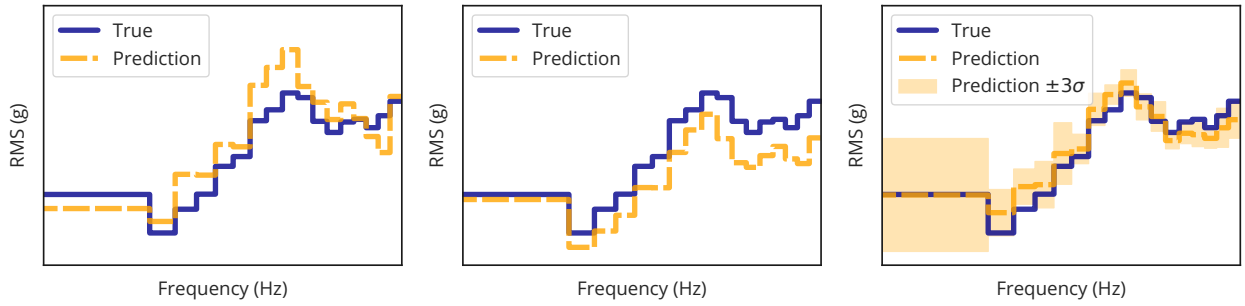
The classical evaluation protocol described above relies on the hypothesis that variables are independent and identically distributed (i.i.d.), and gives a measure of how well a model can interpolate. Here however, there is knowledge about the physical process that generated the data, namely, there is a linear dependence between the energy spectra and the dynamic pressure. The goal of the new evaluation protocol is to determine whether the model is able to extrapolate, which is generally not possible for neural networks, but should be possible in some specific cases [XZL+21]. To this end, all data where the dynamic pressure is below 100 hPa is used for training, and all the remaining data is used for testing. The value of 100 hPa corresponds to the mid-range value of the dynamic pressure.

7.1.6.1 . Prediction example for one flight point

Again, one flight point randomly taken from the test set is considered and used to compare the performance of each model, see Figure 7.7. One spectrum prediction for an input data point is shown, associated with the sensor V3Z. The data point corresponds to a dynamic pressure of 210 hPa, more than twice the maximal value of the training set.

For linear regression, under-prediction is observed in the lower frequency band, but substantial over-prediction is present at almost all frequencies except in the highest range. In particular, the peak of the spectrum over the frequency range is overestimated by a factor of three.

In contrast, above the first frequency band, XGBoost consistently under-predicts the spectrum by a nearly constant factor of 2. Best results are observed for the ensemble model, which slightly over-predicts the energy for frequencies lower than the peak frequency, and slightly under-predicts it for higher frequencies. The true spectrum lies within the confidence interval of $\pm 3\sigma$.



(a) Linear regression (b) XGBoost (c) ANN ensemble

Figure 7.7: Predicted spectrum of a given data point for the accelerometer V3Z, extrapolation evaluation, using a log-log scale.

7.1.6.2 . Error distribution over frequency bands for an accelerometer

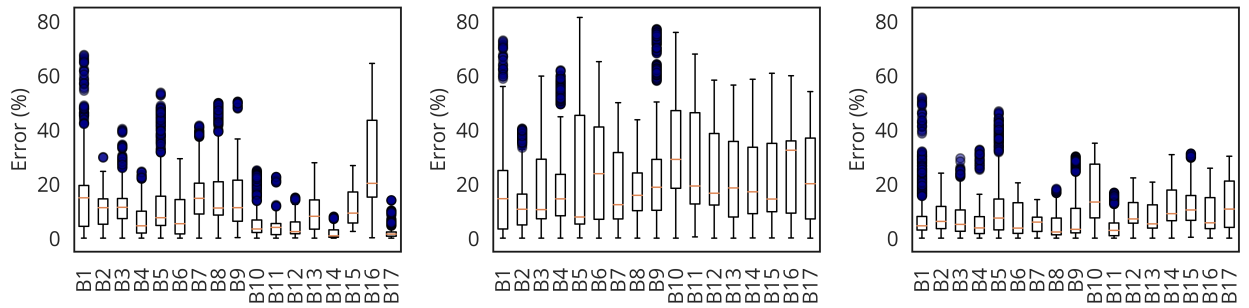
The boxplot of the error averaged over each frequency band is presented. Figure 7.8 shows this for all three models applied on the accelerometer V3Z.

The ensemble model shows the best performances and has its outliers closer to its distributions than with the other models. The distribution spread itself is much higher than for the classical evaluation protocol, as it is a much harder problem. Note that the frequency band B_1 is constantly a high error band. This is explainable by its higher variability and link to many variables affecting the aerodynamic field.

7.1.6.3 . Dynamic pressure projection

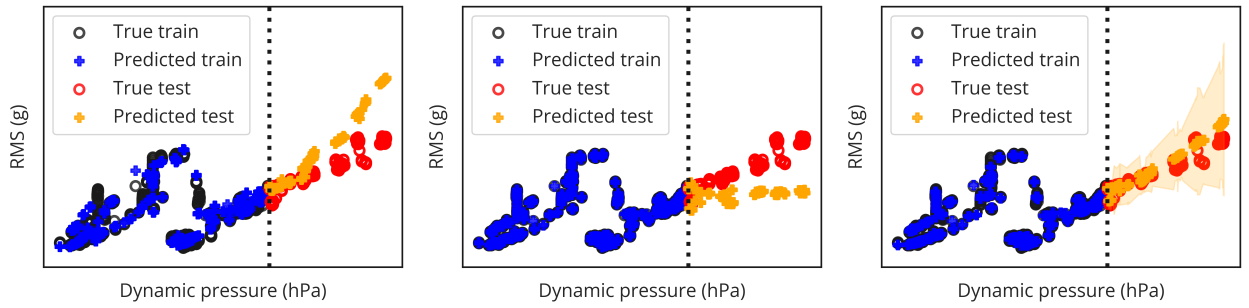
To better understand the behavior of the models, Figure 7.9 compares how predicted and real RMS levels in a specific frequency band vary with the dynamic pressure for both training and test sets. The frequency band B_8 is selected, which provides a stringent test of the models, as it is highly influenced by aircraft configuration for dynamic pressure values under 100 hPa and mainly by the dynamic pressure when it is over 100 hPa.

Linear regression is not able to fit accurately the training set, which is likely to originate from a lack of expressivity of the model. In the test set, a positive trend is predicted although its slope is over-



(a) Linear regression (b) XGBoost (c) ANN ensemble

Figure 7.8: Boxplot of the error per band for the accelerometer V3Z, extrapolation evaluation.



(a) Linear regression (b) XGBoost (c) ANN ensemble

Figure 7.9: RMS levels on the B_8 frequency band depending on the dynamic pressure for the accelerometer V3Z, using a log-log scale. The black dotted line separates the training set (left) and the test set (right). The shaded region corresponds to a value of $\pm 3\sigma$ for the ensemble.

estimated in this case. Since this is a one-dimensional projection from a much higher-dimensional space, a strictly linear dependence is not observed as other variables come into play in the linear representation.

XGBoost almost perfectly reproduces the training set but is unable to extrapolate to the test set, as it predicts an almost constant RMS with the dynamic pressure. This is consistent with the fact that it is based on decision trees.

In contrast, the ensemble model fits well the training set while extrapolating correctly the increasing trend of the test set. One can see that the uncertainty associated with the model also increases with the dynamic pressure, which is consistent with the increasing distance between the test data points and the training set.

7.1.6.4 . Global error

Finally, this evaluation is concluded by giving the overall predictive performance of each model in Table 7.2. For each accelerometer, the average of the ALE error is computed across every test data point and frequency band, while the last column gives the average across all 5 accelerometers.

The worst results are obtained with XGBoost, with a global mean error of 21.9%. As stated before,

	Accelerometer					
	V1Z	V2Z	V3Z	V4Z	V5Z	Mean
Linear regression (%)	15.1	16.2	10.2	16.9	15.9	14.9
XGBoost (%)	21.5	22.3	22.8	22.9	20.0	21.9
ANN ensemble (%)	**9.1**	**12.8**	**8.6**	**10.2**	**13.3**	**10.8**

Table 7.2: Mean error for each accelerometer depending on the model, extrapolation evaluation. Bold denotes the best results for a given accelerometer; the lower the better.

it is based on an ensemble of trees that do not apply any transformation to the data. As a result, its extrapolation abilities are limited to a constant function, which explains its poor performance. The error obtained with linear regression is smaller (about 15%), but still inferior to the ensemble model, which provides an error of only 10.8%. The difference can be attributed to the limited expressivity of the linear regression. In addition to the lower error, the ensemble model provides a confidence interval that captures 98.3% of the predictions into a $\pm 3\sigma$ interval, while 98.7% of the predictions are below $\mu + 3\sigma$.

The error of each neural network composing the ANN ensemble is given in Figure 7.10. The ensemble error is significantly lower than the average error of ANNs: for V2Z, the mean of ANNs error is two times higher than the ensemble error.

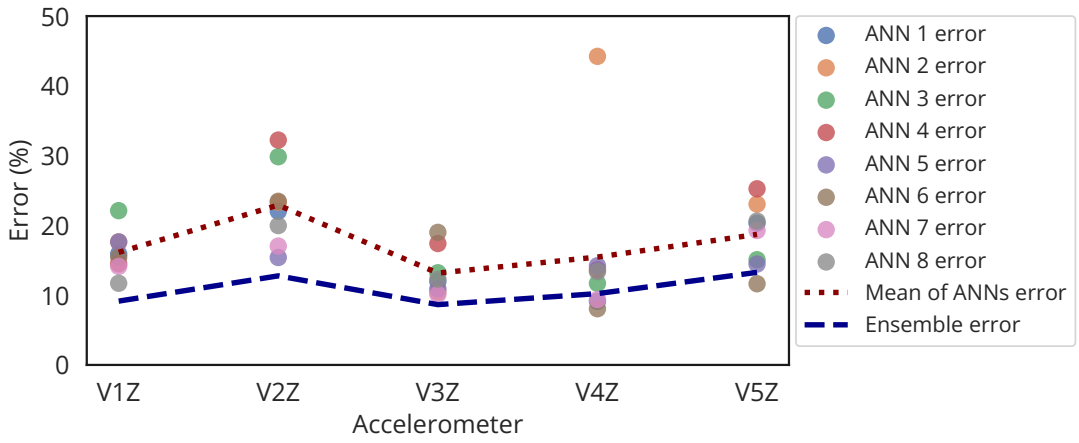


Figure 7.10: Mean error for each accelerometer depending on the neural network, extrapolation evaluation.

These results show that the ANN-based ensemble model is best suited to extrapolate the energy along the dynamic pressure variable. This is consistent with results indicating that ANNs using a ReLU activation function are able to extrapolate linearly [XZL+21]. Extrapolation properties for the model may be desirable in order to enforce good prediction outside the range of the training set, which is typically non-convex.

7.1.7 . Model sensitivity to frequency bands definition

Reducing PSDs into RMS levels per frequency band provides results closer to the actual engineering needs, increases robustness to noise, decreases training time, and allows for more straightforward interpretability of the results. However, the choice of frequency bands can induce a bias in the model: under the assumption of uncorrelated noise, the noise is expected to increase as the bandwidth decreases, since it includes fewer points.

Here, a sensitivity analysis of models to the width of RMS bands is conducted. The 3 models are compared using the classical evaluation protocol, for 4 different discretization levels: 8, 16, 32 bands, and 3897 points –corresponding to maximal resolution and no reduction–, represented in Figure 7.11.

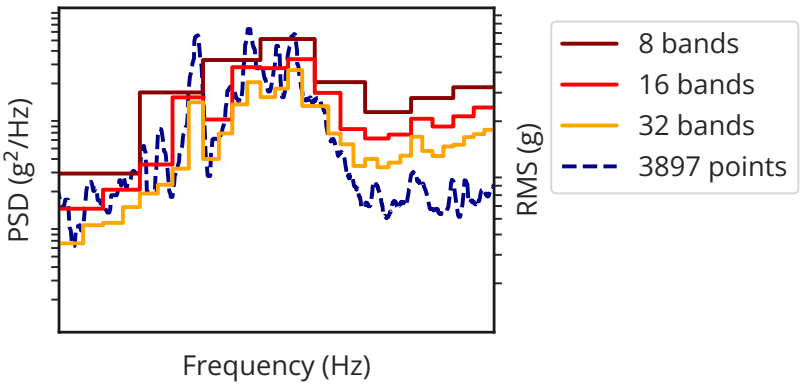


Figure 7.11: Illustration on the same spectrum of the different representations compared in this study, using a log-log scale.

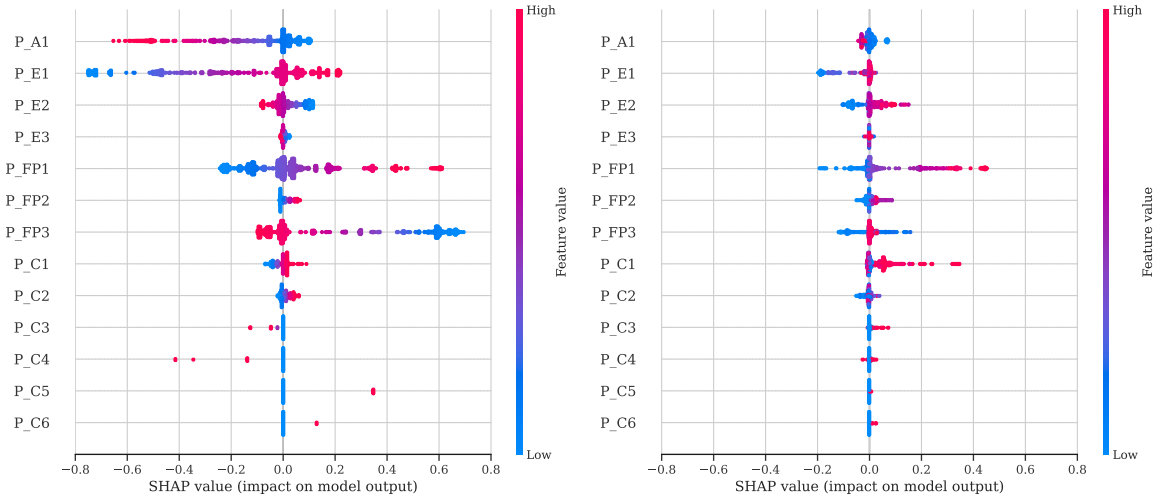
Results show that the error tends to grow with the number of outputs, see Table 7.3, which is not surprising since the complexity of the problem increases. However, the differences can be deemed non-significant when comparing different RMS band level discretization of 8, 16, and 32 bands.

	Discretization level (# points)			
	8	16	32	3897
Linear regression (%)	2.5	2.6	2.7	5.7
XGBoost (%)	0.2	0.3	0.3	1.0
ANN ensemble (%)	0.8	0.9	1.1	2.8

Table 7.3: Error on the classical evaluation protocol depending on spectra discretization level.

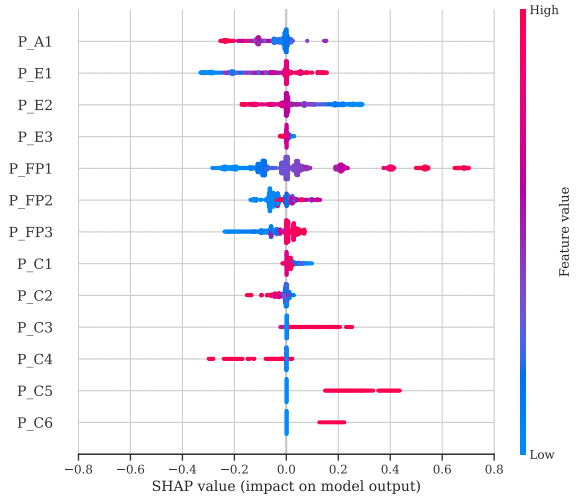
7.1.8 . SHAP values

While comparing two linear regression models can be a simple task, comparing widely different models – including black box ones such as ANNs – is much harder and necessitates a dedicated method. Interpretability, introduced in Section 5.5, is especially important to understand what a model bases its predictions on. This allows to trust a model within a given scope. For this problem, the method used should be model-agnostic and allow to efficiently quantify the importance of each feature for any given prediction. To this end, the SHAP method previously introduced in Section 5.5.4 is used.



(a) Linear regression

(b) XGBoost



(c) ANN 1

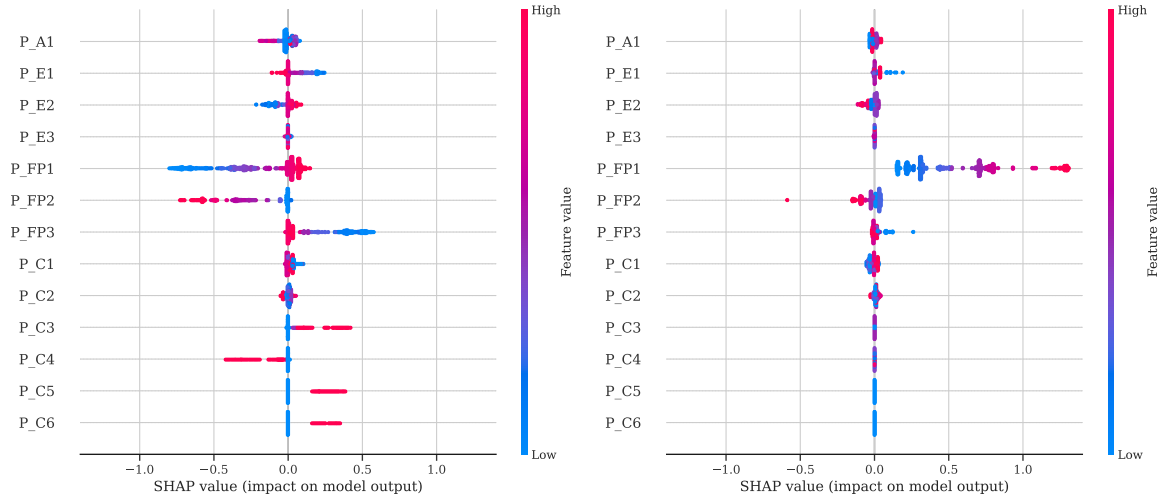
Figure 7.12: SHAP values on the training set of the classic protocol, for different models. Accelerometer V3Z, frequency band B_8.

In our case, the KernelSHAP implementation is used, which has the advantage of being model-agnostic. For consistency, the frequency band B_8 and the sensor V3Z are studied. Figure 7.12 shows the SHAP values of the entire training set of the classic evaluation protocol, for each model. For the ensemble, the SHAP values of only one ANN are considered. This is acceptable as SHAP values of different ANNs have been studied and proved to be similar.

SHAP values are computed relatively to a background data point, quantifying how much of an impact on the output a feature has. For this figure, the background data point consists of the median of the training set. It corresponds to an often encountered flight point with a clean configuration, when all high-lift devices and air brakes are retracted to minimize drag. It can be seen that the dynamic pressure P_{FP_1} has a similar importance for all models. Engine-related parameters P_{E_1} and P_{E_2} are also

important for each model, but at different levels. Surprisingly, configuration-related variables – with the exception of P_{C_1} – have low importance for the XGBoost model.

It is interesting to compare the results of Figure 7.12 with the input-output correlation matrix given in Figure 4.11b. First, it should be noted that SHAP values are relative to the background data point. Considering only the B_8 column of the correlation matrix for comparison purposes, several differences can be observed. Multiple low-correlated variables can have significant SHAP values, such as P_{C_1} and other configuration-related variables. Conversely, strongly correlated variables can have small SHAP values, for instance, if they are linearly related with a small proportionality coefficient. For the linear regression model, the features with the highest SHAP values are P_{E_1}, P_{FP_3}, P_{FP_1} and P_{A_1}, which are also the most weighted features. This shows consistency.



(a) Training set

(b) Test set

Figure 7.13: SHAP values on the training and test sets of the extrapolation protocol, for one ANN of the ensemble. Accelerometer V3Z, frequency band B_8.

Finally, Figure 7.13 shows the SHAP values of both the training and test set, but for the extrapolation protocol. Only one ANN is presented, and the frequency band and accelerometer are the same as in Figure 7.12. Again, the background data is the median data point of the training set, for both the training and test set evaluation, and is thus different from the classical evaluation protocol. It corresponds to a common flight point with a clean configuration. The differences in feature importance between the training and test set are significant. While the dynamic pressure P_{FP_1} is almost the only important variable for the model in the test set, the SHAP values of the training set are much more consistent with what is computed with the classical protocol, which is an expected result. For brevity, the values for the linear regression and XGBoost models are not given for the extrapolation protocol. In the test set, linear regression predictions are highly impacted by P_{FP_1}, P_{FP_3} and P_{E_1}, while XGBoost has extremely low SHAP values for all variables.

7.1.9 . Discussion on stabilized flight phases prediction

The general methodology proposed here allows to predict the vibration environment generated by a variety of unknown sources at different locations of an aircraft, from general flight parameters. Time series acquired at a high sample rate during flight tests are fed through a preprocessing pipeline which generates a reduced representation of broadband energy spectra. This representation is based on discrete frequency bands, allowing for easy manipulation and interpretation of the data. As shown in Section 7.1.7, it introduces robustness in the representation.

This methodology is however limited to stabilized flight phases only, and is not sufficient to produce specifications. As presented in Section 1.3, equipment need to be specified for *current cases*, well represented by stabilized flight phases, but also *severe cases*, which include maneuvers, not modeled here. As a consequence, an extension of this methodology is presented in the next section.

7.2 . Extension to unsteady flight conditions

This section proposes a solution to extend the prediction of broadband levels to all flight phases, including unsteady conditions. The objective is the same as in the previous section, but tackled differently in order to take into account the temporal dependence of the new dataset. This work has been presented in [FNM+23b].

7.2.1 . Extended dataset and features selection

The dataset is generated from the same flight tests, but all flight phases are now considered. This means that the representation of inputs and outputs needs to integrate a temporal dimension. PSDs are computed on a 1-second window using FFT of 2048 points[2], and the dataset is sampled at 10 Hz to have a finer time resolution. Each 0.1 second, a set of general parameters at time t is aligned to a spectrum calculated from $t - 1$ s to t.

As the dataset sample rate is set at $f_s = 10$ Hz and all flight phases are taken into account[3], its size is much larger than the previous one. It contains $m = 787199$ points, almost 22 hours of data.

Compared to the stabilized flight phases dataset, the correlations between features and RMS levels of different frequency bands are lower here. Indeed, the much more complex relationships occurring during unsteady phases are harder to capture using simple pairwise correlations.

Separation between broadband components and kinematic lines is not performed to work with less process data, but a comparison with and without this step could be interesting. The RMS reduction is applied using the same 17 bands as before, see Section 4.2.2.

7.2.2 . Modeling temporal relationships

To account for non-stationary phenomena, the set of features needs to be larger than the one previously used to model stabilized flight phases. In total, $n = 32$ features are kept, including all aircraft angles and angle variations, and the position of various control surfaces. Minimal feature engineering is performed, and could be a path to explore in order to enhance prediction performances.

The input consists of n features sampled with a step Δt over an interval $[t - \tau, t]$. Two parameters are to be set: the value of τ, and the step Δt. This greatly increases the number of features, as the

[2]This is the same as for the dataset of kinematic lines of ground tests.

[3]The exception being flight phases where emergency equipment are used, and non-subsonic regimes.

input vector can be written as:

$$\mathbf{x}_{\text{with history}} \triangleq \begin{bmatrix} \mathbf{x}^{(t-\tau)} \\ \vdots \\ \mathbf{x}^{(t-\Delta t)} \\ \mathbf{x}^{(t)} \end{bmatrix}, \tag{7.4}$$

where the superscript denotes the time at which the feature vector $\mathbf{x}$ is given. Values of $\tau = 1$ s and $\Delta t = 0.1$ s are selected through hyperparameters optimization, which is consistent with the dataset sampled at 0.1 s with PSDs computed on 1-second windows. Hyperparameters optimization shows that these values provide the best results.

The temporal dependence can also be taken into account in the architecture of the model. While using an ANN ensemble is an option, the model would not make any difference between two features at the same time, and the same feature at two different times. A solution is to use an RNN ensemble, as RNNs are able to encode temporal relationships, see Section 5.2.6. The selected architecture of each RNN constituting the ensemble is presented in Figure 7.14. They use one layer of LSTM cells with 30 units, followed by a dense layer of 17 units. LSTM units are used as they improve on the RNN architecture while being relatively easy to train.

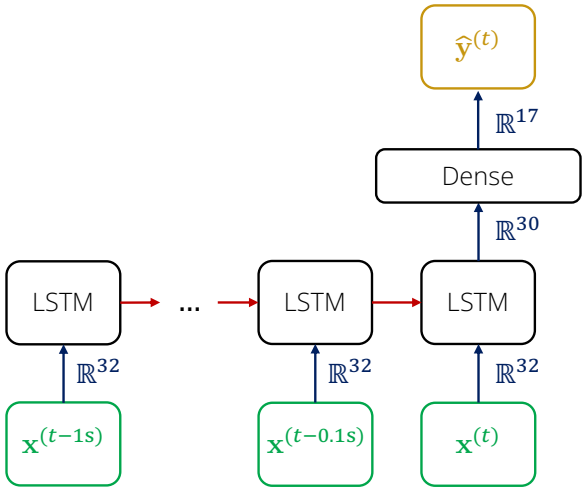


Figure 7.14: Model architecture, comprised of one layer of LSTM cells followed by a dense layer.

Multiple models are compared: linear regression, XGBoost with and without history, ANN ensemble with and without history, and RNN ensemble with and without history. Linear regression with history[4] is not presented here as it does not converge, as well as second-order polynomial regression.

7.2.3 . Comprehensive evaluation methodology

Time dependence requires a more comprehensive evaluation methodology. The one used here is based on the previous section, but intends to evaluate different aspects of generalization. A random

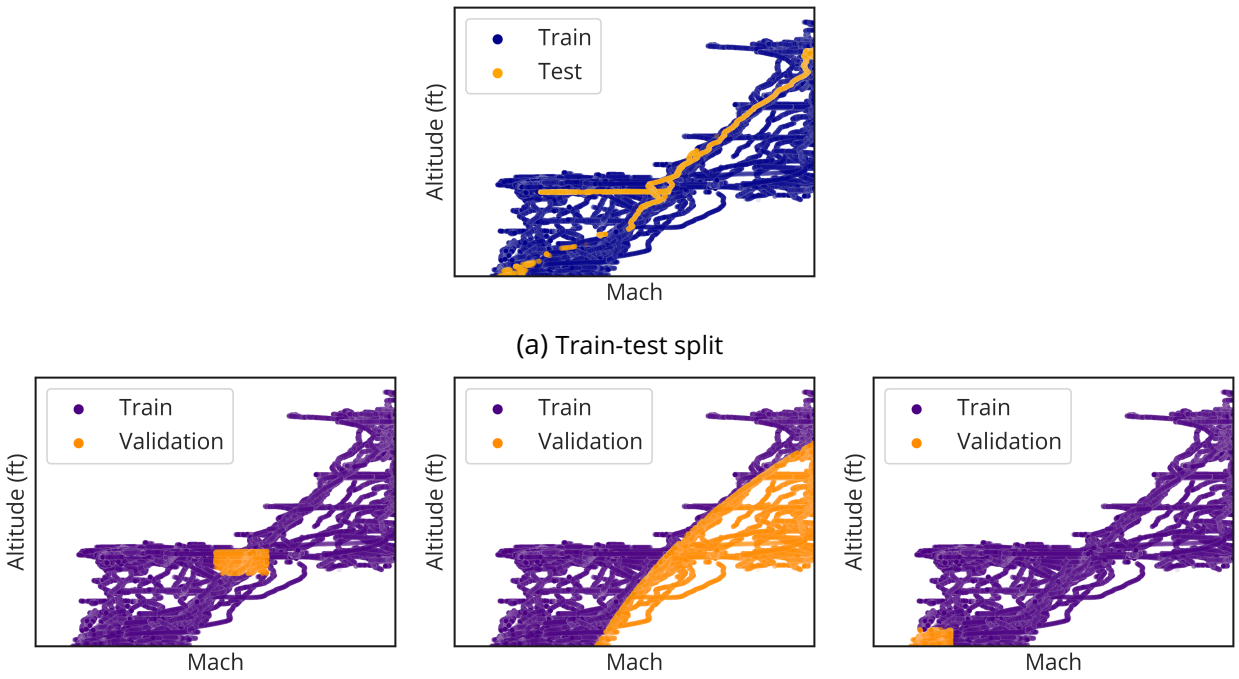
[4]Meaning that $\mathbf{x}_{\text{with history}}$ is used as an input instead of only $\mathbf{x}^{(t)}$.

split is even less relevant in this context, compared to Section 7.1, where time-dependencies are even greater. Thus, no random split is used as it was shown to be non-indicative of generalization ability.

The cross-validation is made out of 3 folds, each evaluating the ability to reconstruct different parts of the flight domain:

- Fold 1: reconstruction of a mid-MACH and mid-altitude flight domain region, learning on all other regions.
- Fold 2: reconstruction of a low-MACH and low-altitude flight domain region which exhibits approach configurations, learning on all other regions.
- Fold 3: reconstruction of high dynamic pressures, learning from low dynamic pressures. As previously, the split is performed at 100 hPa.

In addition to the cross-validation, a flight is kept aside to serve as a final test set.



(a) Train-test split

(b) Fold 1

(c) Fold 2

(d) Fold 3

Figure 7.15: Dataset split: a train-test split is applied (top), then 3 validation folds are taken from the train set (bottom).

The splits are illustrated in Figure 7.15. The train-test proportion is around 95-5%, with the exception of fold 2 at 70-30%. One thing to mention here is that the difficulty of predicting the folds is much harder than the one on the test set. This is especially the case for the dynamic pressure extrapolation.

7.2.4 . Results

Results are summarized for each model and validation/test set in Tables 7.4 and 7.5, using the MALE, see Eq. (5.34), and the maxALE, see Eq. (5.35). As previously, only the test error is considered, which is based on data unseen during training. Accelerometer V3Z is considered, see Figure 7.1, which is the focus of the previous section.

	Fold 1	Fold 2	Fold 3	Test set
Linear regression (%)	6.2	14.9	9.3	5.5
XGBoost (%)	4.2	15.8	6.3	4.1
XGBoost w/history (%)	4.0	16.2	5.6	**4.0**
ANN ensemble (%)	4.1	10.4	4.7	4.3
ANN ensemble w/history (%)	4.8	11.9	6.6	5.0
RNN ensemble w/history (%)	**3.8**	**8.6**	**4.7**	**4.0**

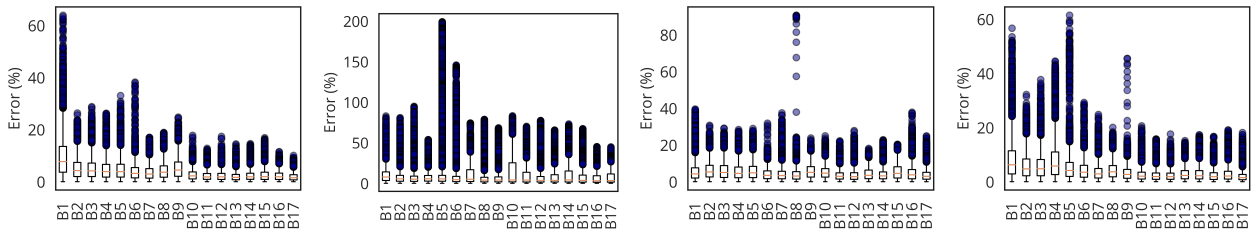
Table 7.4: Mean error, see Eq. (5.34), for accelerometer V3Z depending on the model and the validation/test set. Bold denotes the best results for a given set; the lower the better.

	Fold 1	Fold 2	Fold 3	Test set
Linear regression (%)	120.9	210.2	88.7	73.8
XGBoost (%)	123.9	205.0	93.5	70.3
XGBoost w/history (%)	66.2	219.6	89.1	65.8
ANN ensemble (%)	128.4	**181.8**	90.3	77.3
ANN ensemble w/history (%)	77.0	197.4	**83.8**	81.9
RNN ensemble w/history (%)	**63.9**	199.3	90.7	**61.5**

Table 7.5: Max error, see Eq. (5.35), for accelerometer V3Z depending on the model and the validation/test set. Bold denotes the best results for a given set; the lower the better.

Providing history to the ANN ensemble decreases its performance, while the RNN ensemble is generally better than the ANN ensemble with or without history. The RNN ensemble has the best average performances across all sets. Regarding the max errors, it is the best model on the test set, and the best or close to be the best for the validation folds. The high values of max errors can be seen in Figure 7.16, showing the boxplot of the RNN ensemble model for each set. Many outliers are present, meaning that some phenomena are not well captured. However, it also means that most of the dataset can be well predicted, as seen by the relatively low errors of the end of distributions. A quick analysis of outliers has shown that they are not explainable by the features: an in-depth study should be performed. Finally, it should be noted that the metric used is a relative error, but low-energy errors are less important. A complementary metric could be used to weight errors relative to the RMS value.

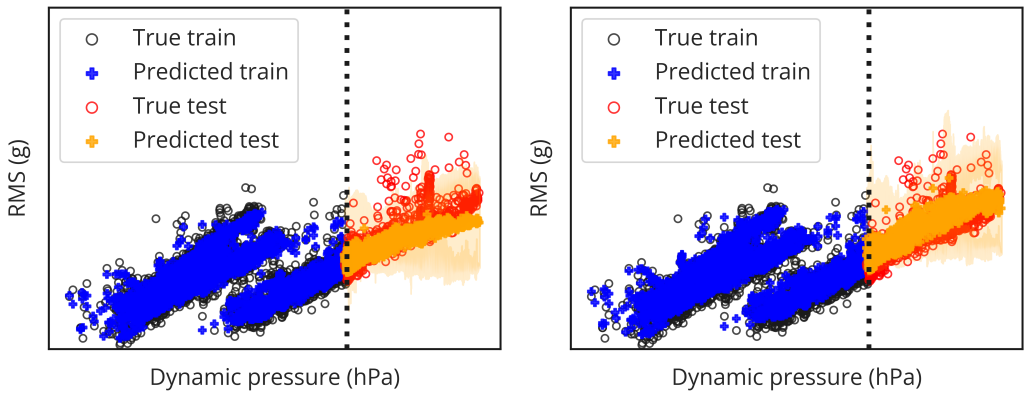
Fold 2, evaluating extrapolation along the dynamic pressure variable, is the hardest to predict and has the highest errors. A projection on frequency band B_8 is given in Figure 7.17 for the RNN ensemble, considering the mean and max prediction. Many points are present in the dataset, and



(a) Fold 1 (b) Fold 2 (c) Fold 3 (d) Test set

Figure 7.16: Boxplot of the error per band for the accelerometer V3Z using the RNN ensemble, for each validation/test set. An interesting result is that maximum errors are not on the same frequency band across folds.

this figure actually shows only a sample of 1% of them. It is interesting to see that the model mean prediction tends to be lower than reality, while the max prediction is a better fit.



(a) Mean prediction (b) Max prediction

Figure 7.17: Projection of the RMS levels on frequency band B_8 depending on the dynamic pressure, using a log-log scale. Superposed to measurements are the mean (left) and max (right) predictions of the RNN ensemble for accelerometer V3Z on fold 2.

Finally, a time-frequency representation of the test set is given in Figure 7.18. This is close to a spectrogram, see Section 1.4.3, but it is instead a succession of consecutive PSDs through time. Predictions (although smoothed-out by the ensemble averaging) are very close to the measurement, with an average error of 4% and a coefficient of determination across all frequency bands of $R^2 = 0.97$. Some phenomena are missed by the models, and correspond to the outliers in Figure 7.16. After further analysis, they are not explainable by the set of features used.

7.3 . Conclusion on the prediction of broadband levels

In this chapter, the problem of predicting broadband spectra for all flight points is tackled using supervised learning models introduced in Chapter 5. The methodology's core lies in data transforma-

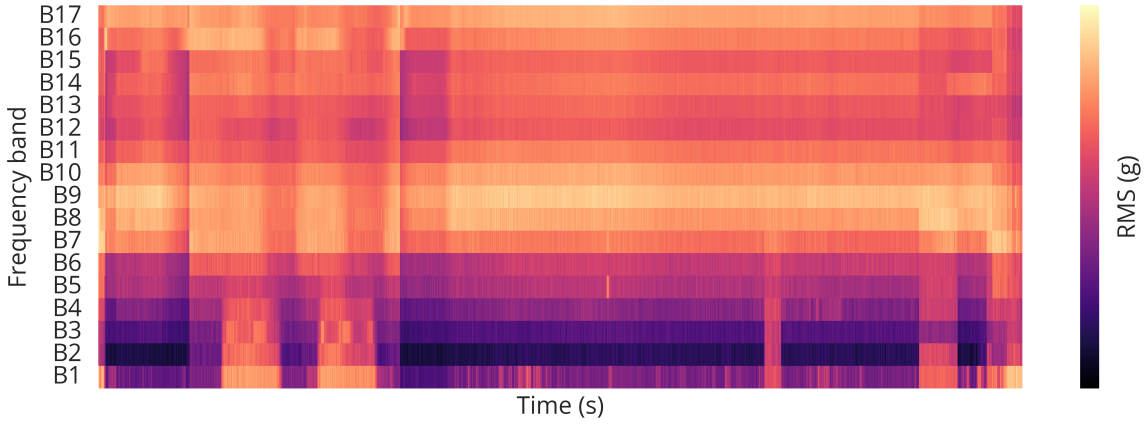
tions presented in Part I, allowing for a robust and interpretable representation. A focus is made on model generalization, where dataset splits based on domain knowledge are performed.

First, only **stabilized flight phases** are considered, as they allow to estimate statistically significant PSDs with assumptions of second-order stationarity and ergodicity being respected. Spectra separation is performed, along with RMS levels per band reduction. The prediction is then made based on a reduced set of general parameters describing the aircraft's attitude and state, using an ANN ensemble. Performances are good, with an average error of 10.8% for the proposed model in the case of extrapolation, along with a confidence interval provided by the use of an ensemble. In comparison, a linear regression gives a 14.9% error, and XGBoost an error of 21.9% on the same dataset. Regarding model interpretation, the SHAP method is used to better understand the predictions of the stabilized flight phases model.

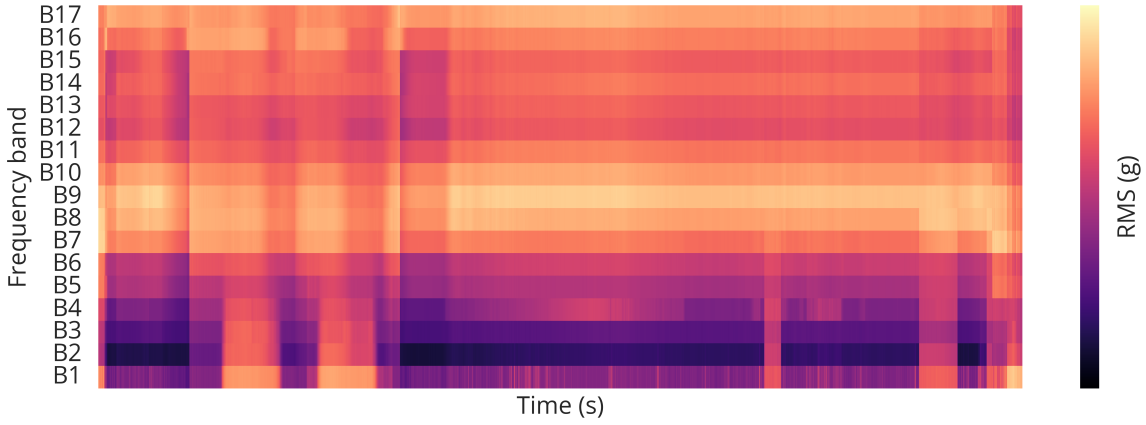
Second, the prediction is extended to **unsteady flight phases**. A larger set of general parameters is used, and temporal relationships are modeled using an RNN ensemble. While spectra are reduced into RMS levels per band, no separation is performed. Kinematic lines energy can thus jump between bands: the effect is assumed to be non-significant, but this should be confirmed. This task is much harder, but good results are still attained with an average extrapolation error of 8.6%. The test flight presented shows that most phenomena encountered during it are captured by the model. This extension to unsteady flight phases should be performed on multiple accelerometers, as for the prediction during stabilized flight phases. Other approaches to encode temporal relationships could be assessed. For example, each maneuver can be reduced to a pattern (set beforehand or learned), and the prediction can be a superposition of effects, summing a broadband level resulting from the flight point and low-frequency levels resulting from a maneuver. This kind of approach can be related to dictionary learning. While the learned dictionary is expected to be more interpretable, the right constraints must be set.

To add robustness to the general methodology and better identify model limitations, outliers need to be understood on a more general level. By implementing more flight phase identification algorithms (see Section 2.3.2), model performances on specific maneuvers could be analyzed. Finally, feature engineering could be performed on a more advanced level to incorporate even more domain knowledge.

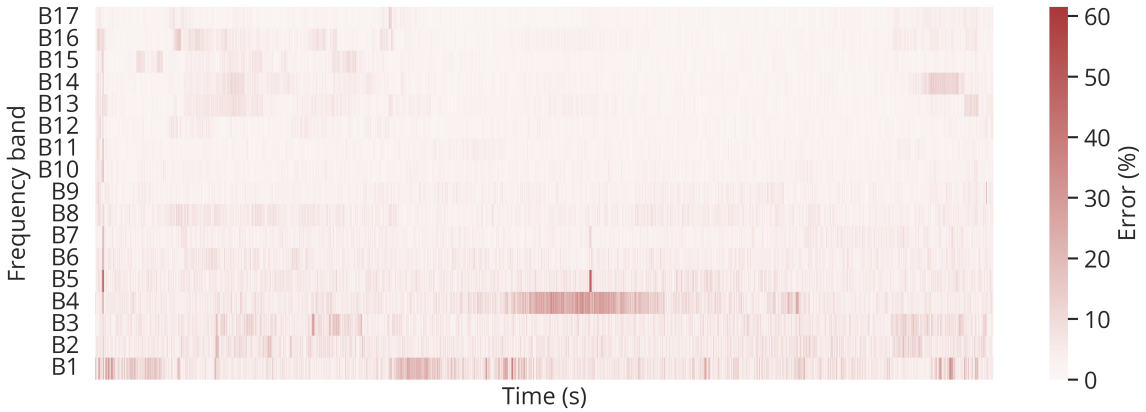
This last chapter of Part II presented the last building block to predict the vibration environment at any point of the flight domain, or more generally the feature space. The ability to predict a spectrum without having to characterize vibration sources and excitation propagation allows for many applications such as anomaly detection – by generating synthetic spectra on the fly and comparing them to measurement –, and vibration flight envelope opening – by using extrapolation abilities. Prediction of a complete spectrum can be attained by combining kinematic lines and broadband levels prediction, allowing for a reduced representation of broadband levels on top of which kinematic lines can be precisely superposed.



(a) True



(b) Prediction



(c) Error

Figure 7.18: Time-frequency representation of the test set: time is on the x-axis, frequency bands on the y-axis, and RMS levels on a log scale (or error on a linear scale) on the color-axis. Measurement (top), prediction (center) of the RNN ensemble for V3Z, and error (bottom) using the MALE without averaging.

Summary of Part II

Based on the tools presented in Part I as it uses the transformation techniques introduced and insights generated, this Part aims to provide a methodology to predict any point in the flight domain, or more generally for the **prediction of any new point in the feature space**. It precedes the work presented in Part III, which aims to extend the prediction to the complete structure of an aircraft.

First, a **literature review of supervised learning** is presented in Chapter 5 detailing different models with the learning and evaluation process. After addressing practical implementation challenges, the question of model interpretability is addressed. Then, based on the hypothesis of broadband separation presented in Section 4.1.1, the problems of predicting kinematic lines and broadband spectra are addressed separately in the subsequent chapters.

Kinematic lines prediction is presented in Chapter 6 with a focus on the left engine kinematic lines, a strong and isolated source. First, a Gaussian process is used to reconstruct RMS values centered on the frequency of kinematic lines during ground tests, with robustness to environmental test conditions. Then, a methodology for transposition from ground to flight conditions is proposed, with the perspective of predicting levels of non-instrumented points in the structure during flight tests.

Broadband prediction is presented in Chapter 7, starting with the prediction of stabilized flight phases. Multiple models are compared, and an ensemble of Artificial Neural Networks is selected as the best model. A focus is made on correctly evaluating models' generalization ability, which is assessed through the use of domain-oriented dataset splits. The problem of interpretability is tackled using the SHAP method. This work has been published in [FNM+23a]. An extension to unsteady flight phases is then proposed and is based on an ensemble of Recurrent Neural Networks, allowing the modeling of temporal relationships. This work has been presented in [FNM+23b].

Part III

Prediction through the aircraft structure

8 - Numerical models for structural dynamics

Contents

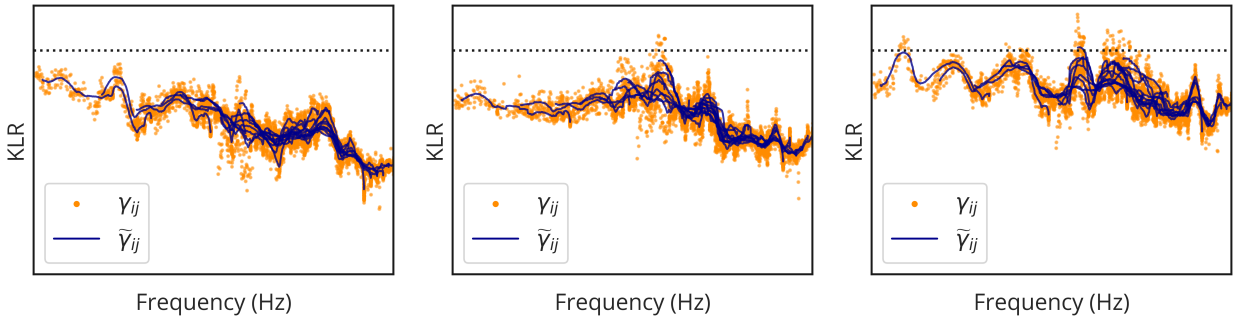
The models implemented in Part II can be seen as being "local", meaning that they are intrinsically linked to a specific point in the structure, i.e., an accelerometer. A general methodology able to predict the vibration environment in a complete aircraft structure requires to model the aircraft structure, allowing to be predictive under structural modifications or even new designs. This is the main objective of Part III, aiming ultimately to find an efficient representation of the aircraft structure that could be used during the design phase. This part begins with a literature review of numerical models. It is followed by their application for the study of vibration propagation paths in Chapter 9, and the assessment of an algorithm from the field of Graph Signal Processing in order to learn a graph structure from measurements in Chapter 10.

This chapter begins with opening remarks on vibration propagation in Section 8.1: while they can be modeled in the first order using the distance to the source, the need for a finer representation is manifest. In Section 8.2, numerical models adapted to different frequency ranges are introduced. In Section 8.3, Dynamic SuperElements are presented as a way to reduce finite elements models. Finally, Advanced Transfer Path Analysis is introduced in Section 8.4, allowing to analyze reduced models to extract propagation paths.

8.1 . A word on vibration propagation

Vibrations in an aircraft structure are propagated through solid and aerial means. In this section, KLRs (Kinematic Lines RMS Ratios) introduced in Sections 2.3.4 and 6.2.1 are used to analyze the

propagation and attenuation in the structure of engine kinematic lines.



(a) Front of the aircraft (b) Middle of the aircraft (c) Rear of the aircraft

Figure 8.1: Superposition of KLRs for several harmonics between the left engine mount j and an accelerometer i in the front (left), middle (center), or the rear of the aircraft (right). The horizontal dotted line denotes the unit ratio; the scale is log-log.

The same representation as in Figure 6.9 is presented in Figure 8.1, for multiple accelerometers placed at the front, middle and rear of the aircraft, see Figure 6.8. The smoothed-out KLRs $\widetilde{\gamma}_{ij}$ show a good superposition between harmonics for each accelerometer i. The higher the distance between the source and the accelerometer, the more attenuated the KLR. The attenuation also increases with the frequency, as seen with lower KLRs.

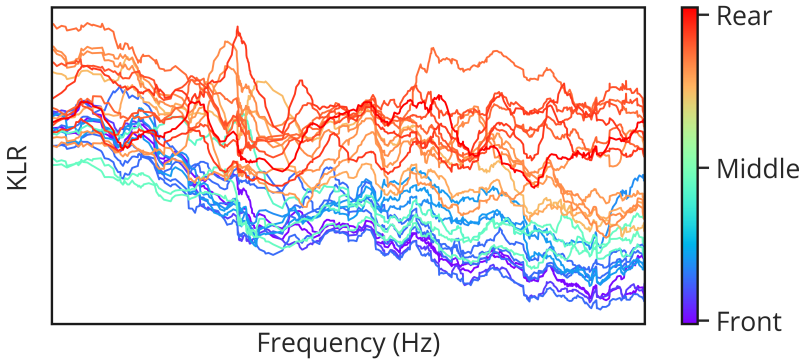


Figure 8.2: Smoothed-out KLRs of a specific harmonic between the left engine mount j and different accelerometers i denoted by the color, on a log-log scale.

Another representation of this effect is given for a broader set of accelerometers in Figure 8.2. Only the smoothed-out KLRs $\widetilde{\gamma}_{ij}$ for a specific harmonic are presented, and the color denotes the position in the aircraft along the longitudinal axis. The same results can be seen, with a greater attenuation with distance to the source and an increased frequency.

The trend of the distance effect is presented in Figure 8.2, showing the mean μ and standard deviation σ of KLRs γ_{ij} across the longitudinal axis. While this first-order effect is already helpful for rough modeling, as in Section 6.2, finer representations containing structural paths add precious information. This is the subject of the following sections.

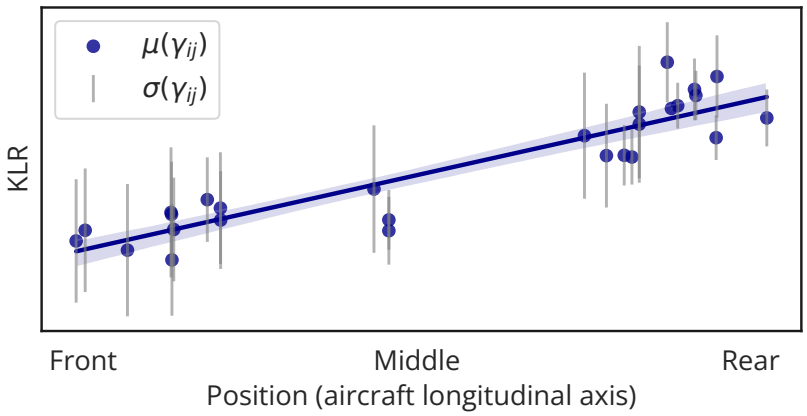


Figure 8.3: Mean μ and standard deviation σ of KLRs γ_{ij} across the aircraft longitudinal axis.

8.2 . Numerical methods

Numerical models allow the simulation of aircraft response to excitations based on a representation of the aircraft structure and materials, along with a characterization of the excitations. They are a tool that can be used during the aircraft design phase when no experimental data is available. Methods are usually divided according to their frequency range. Low-frequency methods are first presented with Finite Element Method (FEM) and Boundary Element Method (BEM). Then, high-frequency methods are considered through the example of Statistical Energy Analysis (SEA). Finally, the mid-frequency range is presented. Here, hypotheses used for the low- or high-frequency range cannot hold, and new methods have to be used. They can be a hybrid between previously shown methods such as FEM and SEA, or can be relying on other assumptions.

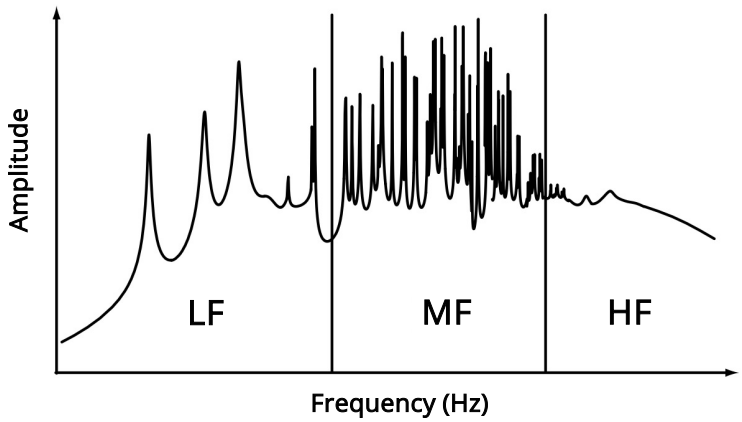


Figure 8.4: Spectrum divided intro low-, mid- and high-frequency ranges, adapted from [OS98].

The frequency range is defined based on the modal density: in the low-frequency domain, modes can be easily separated while in the high-frequency domain their high density brings the need for a statistical approach. This is illustrated in Figure 8.4.

8.2.1 . Low-frequency: Finite and Boundary Elements

In the low-frequency domain, the structure exhibits a modal behavior. The modal overlap is low, and so is the sensitivity to uncertainties. Deterministic techniques[1] such as FEM (Finite Element Method) [ZTZ10] or BEM (Boundary Element Method) [BB94] are used. In practice, these methods are limited to low frequencies, as they require a very fine discretization and a high degree of interpolation to remove numerical pollution and dispersion effects, when the numerical solution deteriorates with increasing non-dimensional wave number [DBB99].

8.2.1.1 . Finite Element Method

The general idea of finite element is to establish, based on weak formulations, a discrete algorithm to seek an approximate solution for a PDE (Partial Differential Equation) problem within a compact domain, considering boundary and/or interior conditions [Man18]. The discretization step allows to represent a domain Ω wish a *mesh* made of *elements*. These elements cover the whole domain without overlapping.

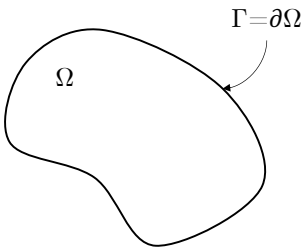


Figure 8.5: *Patatoid* representation of the domain Ω and its boundary $\Gamma = \partial\Omega$.

Let Ω be a domain in $\mathbb{R}^n$ (in practice, $n = 1$, 2 or 3) with a boundary $\Gamma = \partial\Omega$, see Figure 8.5, on which the objective is to solve a PDE with boundary conditions. In weak form, this problem can be written as:

$$\text{Find } u \in \mathcal{H} \quad \text{s.t. } a(u, v) = f(v), \quad \forall v \in \mathcal{H}, \tag{8.1}$$

where $\mathcal{H}$ is a Hilbert space, and assuming the uniqueness of u. $a_h(\cdot, \cdot)$ is a bilinear operator and $f(\cdot)$ a linear operator.

This problem is discretized at several levels: the domain Ω is discretized into elements, the unknown field is interpolated using a finite number of simple functions defined on each element, and $\mathcal{H}$ is replaced by $\mathcal{H}_h$ of finite dimension N_h. The approximation u_h of u in this basis is:

$$u_h = \sum_{i}^{N_h} q_i e_i, \tag{8.2}$$

with $(e_1, \ldots, e_{N_h})$ a basis of $\mathcal{H}_h$. Thus, the problem becomes (if a is linear):

[1] Although stochastic formulations exist.

$$\text{Find } q_1, \dots, q_{N_h} \quad \text{s.t.} \sum_i^{N_h} q_i a_h(e_i, v_h) = f_h(v_h), \quad \forall v_h \in \mathcal{H}_h, \tag{8.3}$$

which can be rewritten, using bilinearity of $a_h(\cdot, \cdot)$ and linearity of $f(\cdot)$, and decomposing v_h, as the following system:

$$\begin{bmatrix} a_h(e_1, e_1) & \dots & a_h(e_{N_h}, e_1) \\ \vdots & \ddots & \vdots \\ a_h(e_1, e_{N_h}) & \cdots & a_h(e_{N_h}, e_{N_h}) \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ q_{N_h} \end{bmatrix} = \begin{bmatrix} f_h(e_1) \\ \vdots \\ f_h(e_{N_h}) \end{bmatrix}. \tag{8.4}$$

In matrix form, it can be written as:

$$\mathbf{K}\mathbf{q} = \mathbf{f}, \tag{8.5}$$

with $\mathbf{K}$ the *stiffness matrix*. By choosing e_i with small support, $\mathbf{K}$ can be sparse. Moreover, the numbering of e_i is done so that the bandwidth of $\mathbf{K}$ is as thin as possible.

If the continuous bilinear form a is coercive, i.e., $\exists\ \alpha > 0$ s.t. $a(u,u) \geq \alpha \|u\|_2\ \forall\ u \in \mathcal{H}$, then from Lions–Lax–Milgram theorem there is a unique solution. Multiple inversion methods exist, such as Gauss or matrix decomposition (LU, Cholesky, etc.).

In the case of structural dynamics, the problem becomes:

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{C}\dot{\mathbf{q}}(t) + \mathbf{K}\mathbf{q}(t) = \mathbf{f}(t), \tag{8.6}$$

where $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are respectively the first and second order time derivatives of $\mathbf{q}$. It is Eq. (8.5) with the addition of inertial forces $-\mathbf{M}\ddot{\mathbf{q}}$ and viscous forces $-\mathbf{C}\dot{\mathbf{q}}$. For the determination of a free response (excitation $\mathbf{f} = \mathbf{0}$) or a periodic response ($\mathbf{f}$ is periodic), modal decomposition can be used. The time dependence allows to model a non-stationary problem to determine a transient response. In this case, a temporal scheme, explicit or implicit, can be used to find the discretization $\mathbf{q}$ of the field u and its temporal derivatives $\ddot{\mathbf{q}}$ and $\dot{\mathbf{q}}$ at each time t, satisfying the equation of dynamics.

When the frequency content of the excitation is broadband, errors on each wavelength accumulate quickly. In this case, it can be interesting to solve into the frequency domain. The problem is thus rewritten using angular frequency ω:

$$(-\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K})\mathbf{q} = \mathbf{f}, \tag{8.7}$$

and can be solved assuming a proportional damping[2]: $\mathbf{C}$ is a linear combination of $\mathbf{M}$ and $\mathbf{K}$, which holds if only structural damping is at hand. By computing the first modes representing the first frequencies of the system, the approximated solution can be projected into the associated subspaces to solve a smaller problem.

Multiple FEM improvements were proposed: adaptive FEM [SJA04, SH97], stabilized FEM [BIPS95, BB07, DHTY12, FDDC89, HH92], Domain Decomposition Methods [ACMP08, FR91, FFML97, Hug95], Component Mode Synthesis [Mac71, SAJEH11], Multiscale FEM [AV12, KT11], and enrichment methods [MB96].

[2] Also known as Basile hypothesis.

8.2.1.2 . Boundary Element Method

The FEM is based on Ritz method [Rit09], where the unknown function is a sum of basis functions, defined in the whole domain Ω. The BEM (Boundary Element Method) [BB94, Bon99] is based on Trefftz method [Tre26], and translates the domain problem into a boundary value problem by applying Green's theorem[3] to the weak formulation, while using linearly independent interpolation functions that satisfy *a priori* the PDE inside the domain. See [KK95] for more details on Trefftz method.

This method can only be used for differential equations that are linear and homogenous. The problem is still written as $\mathbf{Kq} = \mathbf{f}$ but the integrals are only defined on $\Gamma = \partial\Omega$, the boundary of the domain. BEM is preferred over FEM when the propagation domain becomes infinite, as only its boundary surface has to be discretized.

8.2.1.3 . Additional low-frequency methods

FEM and BEM are both robust methods used in industrial applications. However, they are not adapted to large deformations, in which case the mesh is problematic. For this meshless methods [DA16, IOCDP03, ZLSL01] can be used, which rely on a set of scattered nodes or points to represent the domain instead of a mesh. This method was first introduced in smoothed particle hydrodynamics [GM77], then applied in solid mechanics [LPC+93]. While they allow to model large deformations, meshless methods can be computationally expensive and require careful treatment of boundary conditions.

Finally, many variations have been developed. X-FEM (eXtended-FEM) [MDB99] allows for the modeling of discontinuities such as cracks or interfaces without the need for explicit mesh refinement. Finite Element Exterior Calculus [AFW06] combines FEM with concepts from differential geometry and exterior calculus.

8.2.2 . High-frequency: Statistical Energy Analysis

In the high-frequency domain, the interest is in describing a global behavior in terms of energy. The modal overlap is high, and so is the sensitivity to uncertainties.

SEA (Statistical Energy Analysis) [LM62, LDD95, Smi62, LB04, CLK05] models the vibration energy transfers across partitions – subsystems – of a structure. An illustration is given in Figure 8.6. The balance between substructures Ω_i in terms of power is written:

$$P_i^{\text{Input}} = P_i^{\text{Dissipation}} + \sum_j P_{ij}^{\text{Coupling}}, \tag{8.8}$$

where P_i^{Input} is the input power of Ω_i, $P_i^{\text{Dissipation}}$ is the dissipated power of Ω_i, and P_{ij}^{Coupling} the exchanged power between Ω_i and Ω_j, supposed to be proportional to their modal energy difference:

$$P_{ij}^{\text{Coupling}} = \omega\, \eta_{ij}\, n_i \left(\frac{E_i}{n_i} - \frac{E_j}{n_j} \right), \tag{8.9}$$

where n_i and n_j are the modal densities of Ω_i and Ω_j, η_{ij} is the coupling loss factor, and E_i and E_j the energies of Ω_i and Ω_j.

[3] Green's theorem is the 2D special case of Stokes' theorem, and relates a line integral around a simple closed curve C to a double integral over the place region D bounded by C.

SEA relies on strong assumptions, see [Mac03]. Mode groupings are constructed such that statistically, the level of each mode group is similar. Therefore, the method relies on the assumption of a high modal density within the studied frequency range. Another strong assumption of SEA pertains to the transmission energy, assumed to be only between adjacent subsystems, using an assumption of weak coupling.

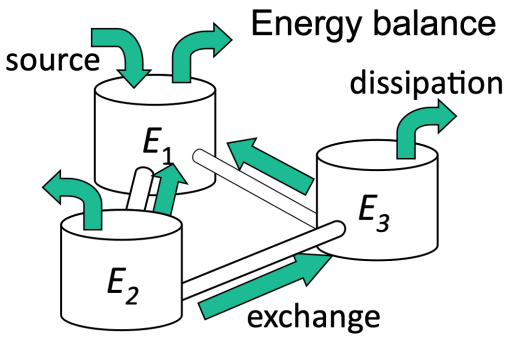


Figure 8.6: Illustration of SEA from [LBCM10]: a complex structure is divided into sub-systems exchanging vibrational energy.

The energy flow between subsystems is described using a set of linear equations for the input, storage, transmission (proportionally with the difference of modal energies), and dissipation of energy.

SEA provides an average vibrational energy level per sub-structure and is too global for mid-frequency problems. This method does not yield a predictive solution as it requires a priori knowledge of measured coupling factors. Finally, substructuring is a challenging problem [TG06], and there is no trivial way to discretize a domain Ω into subdomains Ω_i.

8.2.3 . Mid-frequency: hybrid and Trefftz methods

At medium frequencies, while a modal behavior is still observable, the modes are superposed and cannot be separated: the Finite Element Method is ill-suited due to the required mesh refinement, and the calculation of the modal basis is also unfeasible. Statistical Energy Analysis, on the other hand, is too global and does not allow a satisfactory description of the solution.

The mid-frequency methods can be divided into **hybrid approaches**, combining low and high frequency methods, and **Trefftz approaches** in which solutions of the homogenous differential equation (more generally, adjoint differential equation) are used as weighting functions to solve the problem on the boundary [KK95]. It must be mentionned that additional methods exists, notably based on an energy operator [Soi98] or on the theory of structural fuzzy [Soi93].

8.2.3.1 . Hybrid approaches

The **Hybrid Finite Element/Statistical Energy Analysis** (FE/SEA) [SL05] approach partitions the system into low-frequency and high-frequency regions. The low-frequency regions are analyzed using FEM, while the high-frequency regions are analyzed using SEA. The results from both analyses are then combined to provide a full system response. Multiple developments of this method have been

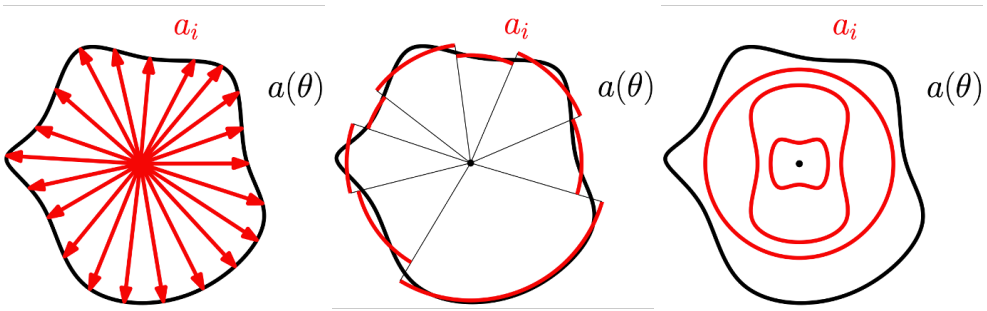
proposed [DOP12, LC09, RLDV14], with attempts to leverage experimental data [CML+19] or to model uncertainties [CL13].

The **wave-based method** [DvSV02] provides a detailed representation of the energy flow in a system, considering the wave nature of the energy propagation. It can be coupled with FEM [vDV05] or with BEM [ABH+14]. An extension of BEM to plane waves has been introduced in the **Wave Boundary Element Method** [PTB03, PTB04].

8.2.3.2 . Variational Theory of Complex Rays

The Variational Theory of Complex Rays (TVRC) [Lad96] is a Trefftz method specifically made for the mid-frequency range. It is particularly adapted to problems where the structure can be divided into homogeneous sub-structures.

The idea is to divide a domain into subdomains, then get a weak form from the boundary and coupling functions using a test function solution of the reference equation. Shape functions are then injected into the weak formulation, their amplitude being the unknown. Multiple shape functions can be used: ray-VTCR [Lad96] uses plane waves, sector-VTCR [LRRB03] uses angular sectors, and Fourier-VTCR [KLR12, RLK13] uses a Fourier series. An illustration is given in Figure 8.7.



(a) Rays (b) Sectors (c) Fourier

Figure 8.7: Different types of shape functions used in VTCR with their amplitude a, from [Cat16].

As per [Cet19], this approach has been applied to several problems: plates [LARB01], general shells [Cat16, RLK13, RLR04], transient dynamics [Che05, CLR07], 2D unbounded domains [Kov11, KRL13], 3D acoustics [Kov11, KLRB12], othotropic plates [KRL14], orthotropic shells [CBRL15], pyrotechnic shocks [Béz12], broad frequency band [BLRK14].

8.3 . Dynamic Superelements

Analyzing vibration propagation paths can be a complex task when a model contains millions of degrees of freedom. As such, using a reduced representation that is still able to produce exact results can be useful. To this end, the concept of DSE (Dynamic SuperElement) is introduced. It allows to reduce a numerical model by condensing its information on a selected subset of degrees of freedom, which offers multiple advantages [NXN14].

Information can be exchanged between industrial actors without having to provide a full model. For example, discussions between aircraft, mount, and engine manufacturers involve exchanges of

DSEs to perform proper dimensioning. As the reduced model can be considered as a black box (only the model response on a chosen subset of degrees of freedom can be computed), only a minimal amount of information is shared.

Another advantage of DSEs is that they allow to compute multiple load cases quickly. After the initial model reduction, requiring computation on the full system (no precision is lost), computation is fast as only a handful of degrees of freedom needs to be solved compared to the full model that could contain millions of degrees of freedom.

This section introduces the DSEs used in this manuscript, see Chapter 9 for the application.

8.3.1 . Formalism

The behavior of a dynamical system reduced on n degrees of freedom can be summarized by linking a force $\mathbf{f}$ with a resulting displacement $\mathbf{x}$ using an equivalent impedance $\mathbf{Z}$, the DSE, such that:

$$\mathbf{Z}\mathbf{x} = \mathbf{f}, \tag{8.10}$$

with $\mathbf{Z} \in \mathbb{C}^{n\times n}$ the **impedance** matrix in $\mathrm{N/m}$, $\mathbf{x}$ the response of the system, and $\mathbf{f}$ the imposed efforts. All equations are given for a specific frequency. As presented in [GB02], the term *impedance* was first introduced in [Web19] to describe the ratio between the cause of motion, a force, and its effect, a displacement: this is the definition used here[4]. Equivalently, Eq. (8.10) can be inverted and written as:

$$\mathbf{x} = \mathbf{H}\mathbf{f}, \tag{8.11}$$

with $\mathbf{H} \in \mathbb{C}^{n\times n}$ the **admittance** matrix in $\mathrm{m/N}$.

In the following, the admittance at a given frequency is supposed to be equivalent to the value of a frequency response function at a given frequency, as the system is assumed to be linear and time-invariant. Moreover, the terms *transfer function* and *frequency response function* (FRF) are used interchangeably

Two types of **transfer functions** can be defined, whether a signal[5] is measured in response to an applied force or to an applied signal.

For a signal-force transfer function, a signal is measured at i relative to an imposed effort at j:

$$\mathrm{FRF}_{ij}^{\text{Force}} = H_{ij}. \tag{8.12}$$

For a signal-signal transfer function, a signal is measured at i relative to a signal at j. It can be computed using the signal-force transfer function:

$$\mathrm{FRF}_{ij}^{\text{Displacement}} = \frac{\mathrm{FRF}_{ij}^{\text{Force}}}{\mathrm{FRF}_{jj}^{\text{Force}}} = \frac{H_{ij}}{H_{jj}} = \frac{x_i}{x_j}. \tag{8.13}$$

There are multiple ways to compute a DSE. One of them is to compute or measure transfer functions between all relevant degrees of freedom, then invert the resulting system. Thus for each frequency, Eq. (8.10) is solved for $\mathbf{Z}$ for all unitary load cases from $\mathbf{f} = [1, 0, ..., 0]^{\mathsf{T}}$, to $\mathbf{f} = [0, 0, ..., 1]^{\mathsf{T}}$. This lead to $\mathbf{Z}\mathbf{X} = \mathbf{F}$ where $\mathbf{F} = \mathbf{I}$, which gives:

[4] Conventionally, the mechanical impedance is the ratio between a force and a velocity.

[5] A signal can be a response of the system in displacement, velocity, or acceleration.

$$\mathbf{X} = \mathbf{Z}^{-1} \equiv \mathbf{H}. \tag{8.14}$$

This means that the displacement matrix, equivalent to the admittance matrix as the forces are unitary, can be obtained by inverting the impedance matrix. As it is ill-conditioned, the inversion needs to take numerical stability into account.

8.3.2 . Numerical stability

Let us consider the system $\mathbf{A}\mathbf{x} = \mathbf{b}$. When a matrix $\mathbf{A}$ is ill-conditioned, the computation of its inverse is prone to large numerical errors, as a small change in the input (the coefficients of the matrix) can lead to a large change in the output (the solution to a system of equations). This can be quantified using the **condition number**, which estimates how much the output value will change due to small changes in the input value. The condition number of a matrix $\mathbf{A}$ is defined as:

$$\operatorname{cond}(\mathbf{A}) \triangleq \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|, \tag{8.15}$$

which, using an ℓ^2-norm, can be rewritten as:

$$\operatorname{cond}(\mathbf{A}) = \frac{\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{A})}, \tag{8.16}$$

with $\sigma_{\max}$ and $\sigma_{\min}$ the maximum and minimum singular values of $\mathbf{A}$.

This number is always superior to 1, and is small for well-conditioned matrices. As $\mathbf{A}$ gets closer to a singular matrix, $\sigma_{\min} \to 0$ which makes the condition number increase.

Preconditioning a matrix is a strategy used to transform a given problem into a form that is more suitable for numerical solving methods. Its main goal is to reduce the condition number of the problem, which in turn improves the convergence speed of iterative solvers and the numerical stability of the solution. The idea is thus to use a preconditioner P to replace the original problem by:

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{x} = \mathbf{P}^{-1}\mathbf{b}, \tag{8.17}$$

such that $\operatorname{cond}(\mathbf{P}^{-1}\mathbf{A}) < \operatorname{cond}(\mathbf{A})$. This is a left preconditioned system, which could also be written as a right preconditioned system where the following has to be solved:

$$\mathbf{A}\mathbf{P}^{-1}\mathbf{y} = \mathbf{b} \quad \text{and} \quad \mathbf{x} = \mathbf{P}^{-1}\mathbf{y}. \tag{8.18}$$

Both Eq. (8.17) and (8.18) give the same result as long as $\mathbf{P}$ is non-singular. Finally, a two-sided preconditioned system can be used:

$$\mathbf{Q}\mathbf{A}\mathbf{P}^{-1}(\mathbf{P}\mathbf{x}) = \mathbf{Q}\mathbf{b}, \tag{8.19}$$

where $\mathbf{Q}$ and $\mathbf{P}$ are real preconditioners such that $\mathbf{Q}^{\mathsf{T}} = \mathbf{P}^{-1}$.

The choice of $\mathbf{P}^{-1}$ is a trade-off between preconditioning computing time and solving the resulting system. Choosing $\mathbf{P} = \mathbf{I}$ is cheap, but does not precondition the system, while choosing $\mathbf{P} = \mathbf{A}$ gives an optimal condition number of 1, but preconditioning is as difficult as solving the original system. One of the simplest forms of preconditioning is by using a diagonal preconditioner, where $\mathbf{P} = \operatorname{diag}(\mathbf{A})$. This method is particularly efficient for diagonally dominant matrices.

8.3.3 . Admittance space and impedance space

Suppose a 2-degree-of-freedom system. The relationship between admittance $\mathbf{H}$ and impedance $\mathbf{Z}$ can be written as:

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}^{-1} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}, \tag{8.20}$$

and if a third degree of freedom is introduced in the same system, it can be rewritten as:

$$\begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}^{-1} = \begin{bmatrix} Z'_{11} & Z'_{12} & Z'_{13} \\ Z'_{21} & Z'_{22} & Z'_{23} \\ Z'_{31} & Z'_{32} & Z'_{33} \end{bmatrix}, \tag{8.21}$$

where admittances remain unchanged on the degrees of freedom in common, but not the admittance.

More generally, let us consider a system of n degrees of freedom and a system of m degrees of freedom, such that one is a subset of the other. The set of degrees of freedom common to both systems is written S. The equivalence in the admittance space can be written as:

$$H_{ij} = H'_{ij}, \quad \forall\, i,j \in S,\, \mathbf{H} \in \mathbb{C}^{n\times n},\, \mathbf{H'} \in \mathbb{C}^{m\times m}, \tag{8.22}$$

while the non-equivalence in the impedance space can be written as:

$$Z_{ij} \neq Z'_{ij}, \quad \forall\, i,j \in S,\, \mathbf{Z} \in \mathbb{C}^{n\times n},\, \mathbf{Z'} \in \mathbb{C}^{m\times m}. \tag{8.23}$$

Physically, this is coherent with the fact that a transfer function is invariant to the number of degrees of freedom considered, while the impedance is actually an **equivalent impedance** on the set of degrees of freedom, allowing to find the same admittance values as with the original model. When reducing the number of degrees of freedom, information is not lost but condensed into the remaining points.

8.3.4 . Assembling DSEs

Degrees of freedom in a DSE are usually split into **internal degrees of freedom** and **frontier degrees of freedom**. The former are used to compute a value at points of interest, while the latter are used to assemble multiple DSEs.

For example, let us consider the case of assembling the DSE of an aircraft $\mathbf{Z}^{\mathsf{A}} \in \mathbb{C}^{n\times n}$ with the DSEs of the left and right engines, respectively $\mathbf{Z}^{\mathsf{L}} \in \mathbb{C}^{2\times 2}$ and $\mathbf{Z}^{\mathsf{R}} \in \mathbb{C}^{2\times 2}$. For the sake of simplicity, engine mounts are ignored. Assembling them is done through frontier degrees of freedom, by ordering them such that they are aligned. Then, a simple summation is performed. The result, $\mathbf{Z}^{\mathsf{Assembled}}$, can be written as:

$$\mathbf{Z}^{\text{Assembled}} = \begin{bmatrix} Z^{\text{R}}_{11} & Z^{\text{R}}_{12} & 0 & \cdots & \cdots & 0 \\ Z^{\text{R}}_{21} & Z^{\text{R}}_{22} + Z^{\text{A}}_{11} & Z^{\text{A}}_{12} & \cdots & Z^{\text{A}}_{1n} & \vdots \\ 0 & Z^{\text{A}}_{21} & Z^{\text{A}}_{22} & \cdots & Z^{\text{A}}_{2n} & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ \vdots & Z^{\text{A}}_{n1} & Z^{\text{A}}_{n2} & \cdots & Z^{\text{A}}_{nn} + Z^{\text{L}}_{11} & Z^{\text{L}}_{12} \\ 0 & \cdots & \cdots & 0 & Z^{\text{L}}_{21} & Z^{\text{L}}_{22} \end{bmatrix}.$$

The resulting assembled matrix can then be used to solve $\mathbf{Z}^{\text{Assembled}}\mathbf{x} = \mathbf{f}$, with $\mathbf{x}$ and $\mathbf{f}$ of size $n+2+2$. Vibrations generated from engines can then be modeled by injecting efforts at Z^{R}_{11} and Z^{L}_{22}, and the response at internal degrees of freedom such as Z^{A}_{22} can be studied.

The condensation of a numerical model into a DSE is thus a way to quickly exploit it on a selected set of degrees of freedom. As the characterization of the structure is a central interest, understanding how vibration transfers between two points is critical. In the next section, a method that can be applied to DSEs to understand transfer paths is introduced.

8.4 . Advanced Transfer Path Analysis

Transfer Path Analysis (TPA) is a family of methodologies used to analyze the transfer of noise and vibration from sources to receivers. While first being used to ensure stealth of military ships and submarines, it quickly began to be applied in the aeronautic industry to ensure safety, and then more broadly in the automotive industry to ensure acoustic comfort [vdR16]. TPA can be split into three main types:

- **Classical TPA** involves performing operational tests on the assembled product to obtain interface forces between an active and a passive side. The operational forces are determined for the complete system, and modifications to any subsystem require retesting of the entire system. While this method provides high accuracy, it can be time-consuming and complex, as it requires dismounting the active side from the passive side.

- **Component-based TPA** aims to characterize the source excitation by using a set of equivalent forces or velocities that are an inherent property of the active component itself. This is in opposition to classical TPA, where the interface forces obtained are not a characteristic of the source alone but of the assembled dynamics.

- **Transmissibility-based TPA** avoids to explicitly model the vibration transmission , and instead focuses solely on sensor responses. While the insights gained from it are limited to ranking sources and their dominant paths, this class of method is faster and simpler than the two previous ones.

Multiple methods belong to the transmissibility-based TPA: **ATPA (Advanced Transfer Path Analysis)** [MSC05], OTPA (Operational Transfer Path Analysis) [NY06], and Operational mount identification with eXogenous Inputs (OPAX) [JMG+08, JGG+11]. ATPA is the focus of the manuscript, and is treated as a way to better exploit the content of a DSE. Below is a short summary that does not intend to be a comprehensive literature review of this method.

ATPA originated from GTDT (Global Transmissibility Direct Transmissibility), first introduced in [Mag81], and later developed in [GM04, Gua09, GGJA13] through analytical, numerical and experimental results. In [GM04], links to causality are highlighted, see [KDTB01, Eic09]. In [MG05], links to graphs are made, where the direct transfer matrix of a system is shown to be similar to a connectivity matrix. In [APA+19], a comparison between numerical and experimental results is performed, while [MARG21] shows that the solution of a linear system can be decomposed into a finite sum of propagation paths.

To describe vibration propagation paths, ATPA proposes to deal with global and direct transfers, which are detailed below after introducing the methodology.

8.4.1 . Methodology

After selecting the degrees of freedom to study, the ATPA methodology consists of two steps:

1. Measuring the global transfers between all points, experimentally or numerically.
2. Computing the direct transfers from the global transfers.

The selection of the set of degrees of freedom is non-trivial, and it is an active research problem for many applications, including SEA [DPR15, MPR18, TG06].

8.4.2 . Global and direct transfers

A formal definition of global and direct transfers is given below, as they are at the core of the ATPA methodology. Then, a link between them is highlighted to introduce the notion of paths.

8.4.2.1 . Global transfer

A global transfer T_{ij}^{G} is defined as the ratio between the signal at i and the signal at j, with an external force applied on j. It is thus a signal-signal transfer function between those two points, as defined in Eq. (8.13).

$$T_{ij}^{\mathsf{G}} \triangleq \frac{x_i}{x_j} \equiv \mathrm{FRF}_{ij}^{\text{Displacement}}, \quad f_j = 1, f_k = 0, \quad \forall\, k \neq j. \tag{8.24}$$

8.4.2.2 . Direct transfer for ij

A direct transfer T_{ij}^{D} for $i \neq j$ is defined as the ratio between the signal at i and the signal at j, with an external force applied on j, with all degrees of freedom other than i and j being blocked:

$$T_{ij}^{\mathsf{D}} \triangleq \left.\frac{x_i}{x_j}\right|_{x_k=0\ \forall\, k\neq i,j}, \quad f_j = 1, f_k = 0, \quad \forall\, k \neq j, \tag{8.25}$$

which is equivalent to solving the following system:

$$\begin{bmatrix} Z_{ii} & Z_{ij} \\ Z_{ji} & Z_{jj} \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} f_i = 0 \\ f_j = 1 \end{bmatrix}, \tag{8.26}$$

which gives:

$$x_i = \frac{-Z_{ij}}{Z_{ii}Z_{jj} - Z_{ji}Z_{ij}}, \quad x_j = \frac{Z_{ii}}{Z_{ii}Z_{jj} - Z_{ji}Z_{ij}}, \tag{8.27}$$

and thus:

$$T_{ij}^{\mathsf{D}} = -\frac{Z_{ij}}{Z_{ii}}. \tag{8.28}$$

8.4.2.3 . Direct transfer for ii

A direct transfer T_{ii}^{D} at a point i is defined as the ratio between the signal at i with an external force applied at i and all other degrees of freedom that i at blocked, and the signal at i with an external effort applied at i:

$$T_{ii}^{\mathsf{D}} \triangleq \frac{x_i|_{x_k=0 \; \forall \, k \neq i}}{x_i}, \quad f_i = 1, f_k = 0, \quad \forall \, k \neq i. \tag{8.29}$$

Finding $x_i|_{x_k=0 \; \forall \, k \neq i}$ involves solving the one degree of freedom system $Z_{ii}x_i = f_i = 1$ which gives $x_i = \frac{1}{Z_{ii}}$. Numerator x_i is directly equal to $x_i = H_{ii} \equiv \mathrm{FRF}_{ii}^{\mathsf{Force}}$.

Thus:

$$T_{ii}^{\mathsf{D}} = \frac{1}{Z_{ii}H_{ii}}. \tag{8.30}$$

8.4.3 . Link between direct and global transfers

The global transfer between two points i and j, thus its signal-signal transfer function, can be described as a sum of global and direct transfers such that:

$$T_{ij}^{\mathsf{G}} = \sum_{k \neq i} T_{ik}^{\mathsf{D}} T_{kj}^{\mathsf{G}}, \quad \forall i \neq j. \tag{8.31}$$

This is illustrated in Figure 8.8, and can be seen as a decomposition into a sum of paths. In Chapter 9 where this method is applied, the highest contributing paths are identified.

ATPA is thus a simple method to analyze a system under various signals such as acceleration, velocity, displacement, and energy. It does not require dismounting of active and passive parts, nor force measurement. Finally, it is not affected by path cross-coupling issues, allowing to quantify the various sources and paths including those that cancel each other out.

However, it is based on multiple assumptions. The system is supposed to be linear, and causal closure is assumed, meaning that all direct influences between variables should be included in the path diagram. If degrees of freedom are missed, it may lead to incomplete or inaccurate representations of the transfer paths.

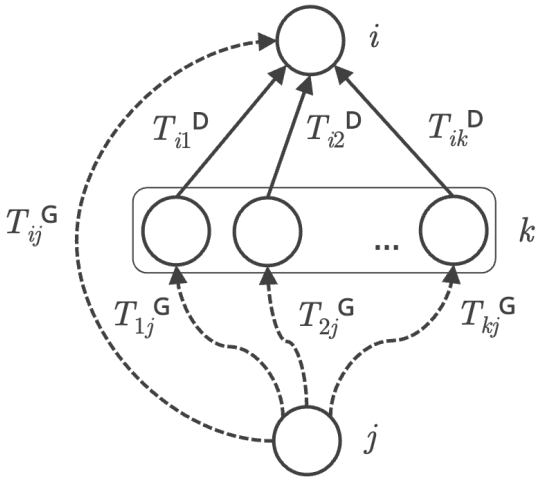


Figure 8.8: Illustration of the decomposition of a global transfer into a sum of paths, see Eq. (8.31).

This chapter has introduced various numerical models, followed by a model reduction method, the DSE. To analyze it, ATPA is introduced. The elements presented here are applied in the following chapter.

9 - Vibration propagation path identification

Contents

When modeling the vibration environment of an aircraft, understanding vibration propagation paths is critical to identify where to adjust the structure in order to reduce vibration levels. Equipment that undergoes lower vibration levels can be less sturdy and thus have a reduced weight, which is critical in the aeronautical industry. In addition to the benefits for equipment dimensioning, identifying vibration propagation paths can be beneficial for other challenges in aircraft design, such as the acoustic comfort of passengers.

To this end, this chapter puts into practice the methods presented in Chapter 8, more specifically the DSEs (Dynamic SuperElements) of Section 8.3 and ATPA (Advanced Transfer Path Analysis) of Section 8.4. In order to assess them extensively, they are applied on a finite element model, with the objective to see what objects are generated by the methods, and how well they can help identify vibration propagation paths. The end goal would be to make the methodology applicable to experimental data, including at higher frequencies. To evaluate this methodology, the generated DSE is first presented, and it is used to compute direct transfer matrices using the ATPA methodology. Then, direct transfer matrices are visualized using 3D graphs. Finally, an iterative propagation path identification algorithm is proposed, and its results are presented.

9.1 . From numerical impedances to direct transfers

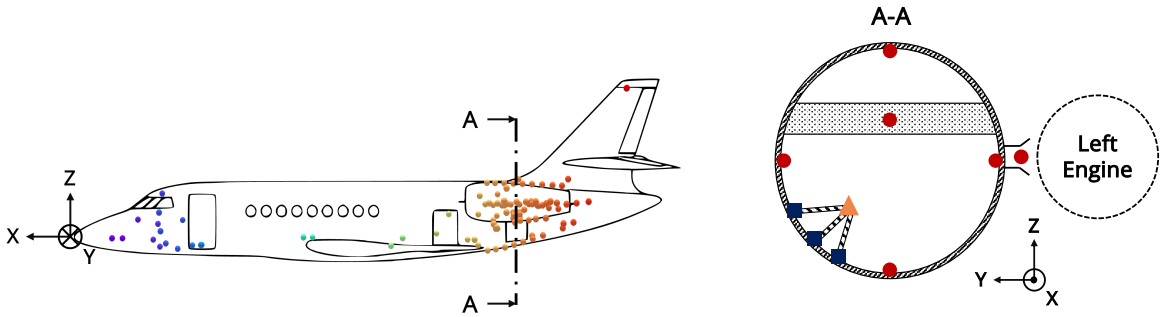
As presented in Section 8.3, a DSE can be used to quickly compute results after performing a model reduction. The DSE is based on a fine FEM model made specifically for vibration and acoustic analysis. This section presents the DSE generated and the computation of direct transfers using ATPA.

9.1.1 . Dynamic Superelement generation

The DSE generated consists of an impedance matrix $\mathbf{Z}$ at each frequency from 10 to 300 Hz. Degrees of freedom are disposed on:

- Various vibration environment accelerometers across the complete aircraft structure.
- The left, right, top, and bottom of each frame at the rear of the aircraft, near the engines.
- At the center of beams linking these rear frames along the lateral axis.
- On left and right engine mounts.

All degrees of freedom are on the 3 axes of translation, X, Y, Z, with the exception of engine mounts, also measuring the 3 axes of rotation. An illustration of the selected degrees of freedom is given in Figure 9.1. In Figure 9.1a, the approximate position of degrees of freedom across the aircraft is presented, and a detail showing a section on the YZ plane through a rear frame is given in Figure 9.1b.



(a) Approximate position across the aircraft

(b) Section A-A on a rear frame

Figure 9.1: Left: Approximate position of the accelerometer superposed to a generic Falcon aircraft for illustration purposes. Node colors denote the longitudinal position. Right: section on the YZ plane showing the position of fuselage degrees of freedom (red circles), an accelerometer degree of freedom (orange triangle), and 3 supplementary degrees of freedom for its link to the structure (blue squares). Neither the left engine nor the left mount DSEs are assembled; only the aircraft DSE is used. NB: a left-handed coordinate system is used.

9.1.2 . Reduction: recalculating an equivalent impedance

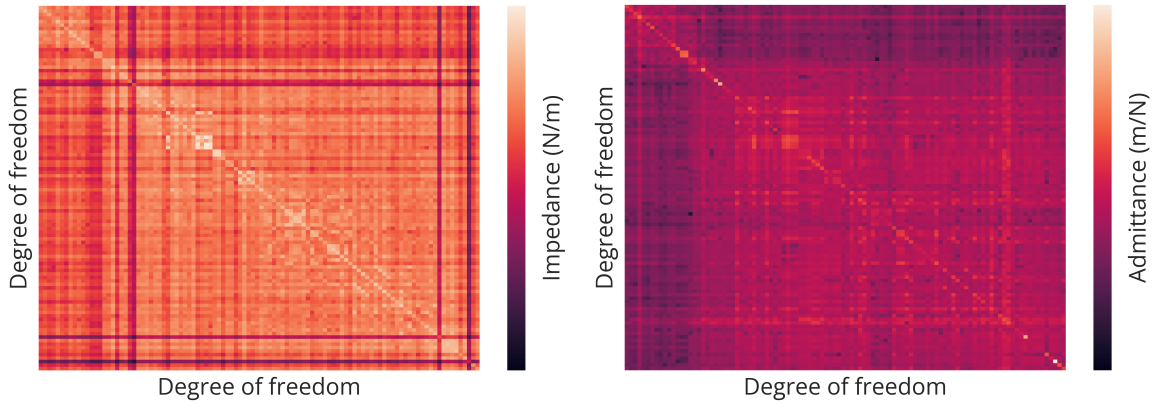
For this study, a focus is made on the Z axis to ease the visualization process. Moreover, a single excitation originating from the left engine is considered; thus, degrees of freedom on the right engine mounts are removed. In further studies, it could be interesting to analyze vibration paths from the left engine to the structure, through the mounts. Here, as only the aircraft DSE is used, only one degree of freedom is kept on the left engine mounts, as presented in Figure 9.1b. For each frequency, an impedance matrix $\mathbf{Z} \in \mathbb{C}^{n \times n}$ is used, with $n = 104$ the number of freedom.

Reducing the size of the impedance matrix $\mathbf{Z}$ is done through inverting and selecting a subset in the FRF space, before inverting again. This allows to recompute the equivalent impedance, and preconditioning must be performed to ensure numerical stability. This methodology stems from Section 8.3.3. Algorithm 4 showcases this process, where $\mathbf{A}_n$ denotes a matrix of size $(n \times n)$.

Algorithm 4 Recalculating an equivalent impedance matrix from size $(m \times m)$ to $(n \times n)$

Require: $\mathbf{Z}_m \in \mathbb{C}^{m \times m}$
  for each frequency **do**
    Precondition $\mathbf{Z}_m$
    $\mathbf{H}_m = \mathbf{Z}_m^{-1}$
    Select subset $\mathbf{H}_n \subset \mathbf{H}_m$
    Precondition $\mathbf{H}_n$
    $\mathbf{Z}_n = \mathbf{H}_n^{-1}$
  end for

To ensure the reduction is performed correctly, a comparison is performed between $\mathbf{H}_n$ obtained by inverting $\mathbf{Z}_n$, and $\mathbf{H}_n$ obtained by inverting $\mathbf{Z}_m$ then selecting a subset of degrees of freedom $n < m$. No error larger than numerical precision is reported. The resulting impedance matrix and its associated admittance matrix are presented in Figure 9.2.



(a) Impedance matrix $\mathbf{Z}$ (b) Admittance matrix $\mathbf{H}$

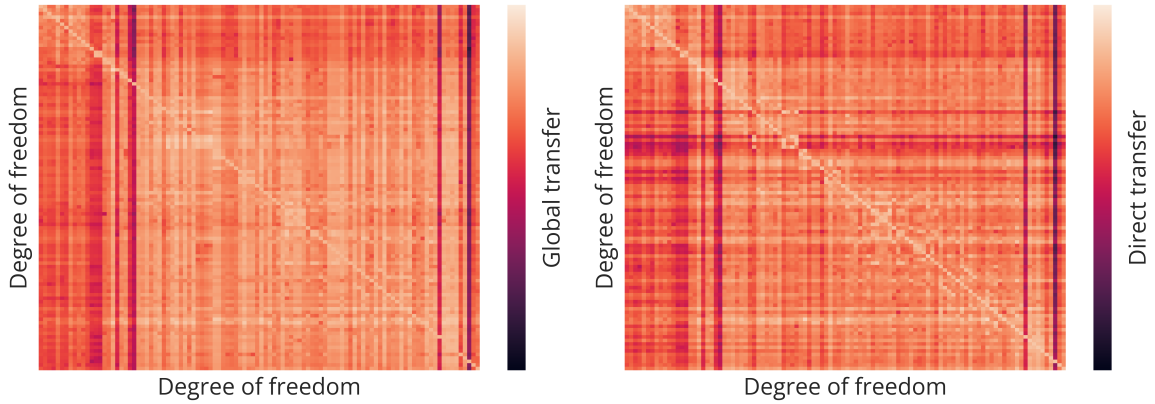
Figure 9.2: Modulus of the impedance and admittance matrices for a given frequency. The colorbar uses a logarithmic scale.

9.1.3 . Calculation of direct transfer matrices

Using the impedance and admittance matrices previously defined, the ATPA methodology can be applied, see Section 8.4.

First, **global transfers** are obtained from the admittance matrix $\mathbf{H}$. This is done through normalizing columns by their diagonal values H_{jj}, see Eqs. (8.24) and (8.13). The result is that FRFs are expressed in signal-signal instead of force-signal, and that the resulting matrix is non-hermitian. A matrix of global transfers at a given frequency is denoted $\mathbf{T}^{\mathsf{G}}$.

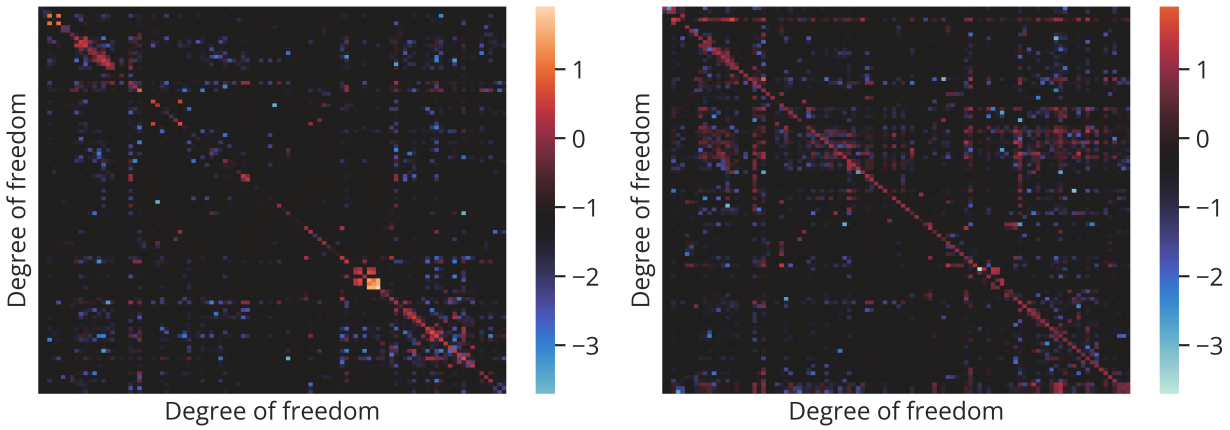
Then, **direct transfers** are computed from the global transfers. Diagonal terms are the inverse of the product of diagonal impedance and admittance, see Eq. (8.30). Non-diagonal terms are negative impedances normalized by their diagonal values, see Eq. (8.28). A matrix of direct transfers at a given frequency is denoted $\mathbf{T}^{\mathsf{D}}$. The resulting matrices for a given frequency are presented in Figure 9.3.



(a) Global transfers $\mathbf{T}^{\mathrm{G}}$ (b) Direct transfers $\mathbf{T}^{\mathrm{D}}$

Figure 9.3: Modulus of global and direct transfer matrices for a given frequency. The colorbar uses a logarithmic scale.

Comparing $\mathbf{T}^{\mathrm{D}}$ and $\mathbf{Z}$ can be done by looking at Figures 9.2a and 9.3b, but another way is presented here. In Figure 9.4, the modulus of each matrix is standardized, see Eq. (3.1), and plotted on a colorbar with the same range, but centered around the mean value of the matrix. This figure allows to compare how the values are distributed on the degrees of freedom of the matrix. While $\mathbf{Z}$ is symmetric and contains strong values in specific regions, $\mathbf{T}^{\mathrm{D}}$ is non-symmetric with lower values distributed on more degrees of freedom.



(a) Normalized $\mathbf{Z}$ (b) Normalized $\mathbf{T}^{\mathrm{D}}$

Figure 9.4: Normalized modulus of impedance and direct transfers matrices. Colorbars are set to the same range (with a logarithmic scale), but their center is set to the mean value of each matrix.

Finally, no comparison between experimental FRFs and numerical ones is presented as this would only inform on how good the model is, which is not the interest here. Moreover, this would require to add the DSEs of mounts and engines to have the correct equivalent impedances, as presented in Section 8.3.4.

9.2 . Visualizing a direct transfer matrix

While looking at a direct transfer matrix is informative, one can miss the underlying structure. In the context of degrees of freedom relating to points in an aircraft structure, visualizing it through a 3D graph can enhance the understanding. This section aims to implement 3D visualizations of the direct transfer matrix and see if they fit domain knowledge.

9.2.1 . Graph representation of the direct transfer matrix

As presented in [MG05], a direct transfer matrix can be seen as the connectivity or adjacency matrix of a graph. A graph is a set of nodes connected by a set of edges. How the set of edges connects the set of nodes is described by the adjacency matrix. As it is a *weighted* adjacency matrix, it is denoted $\mathbf{W}$. For a more formal presentation of graphs, see Chapter 10.

The objective of this section is to visualize the direct transfer matrix for a given frequency as a 3D graph. In this graph, the nodes are the degrees of freedom of the system, disposed at their XYZ coordinates, and the edges are the connections between the nodes given by the direct transfer value. Setting $\mathbf{W} = \mathbf{T}^{\mathsf{D}}$ poses multiple problems. As the diagonal is non-zero, nodes are connected to themselves which clutters the visualization. As there is no zero value in $\mathbf{T}^{\mathsf{D}}$, all nodes are connected. To overcome these problems, the diagonal is set to 0, as well as low values through a hard threshold. In addition, values that are considered to be too high are also set to 0. It must be emphasized that this is performed only for visualization, as identifying some strong local connections is not the point. Only the modulus is considered as $\mathbf{T}^{\mathsf{D}}$ is a complex matrix. Thus, the elements of $\mathbf{W}$ are set as follows:

$$W_{ii} = 0, \quad W_{ij} = \begin{cases} \left|T_{ij}^{\mathsf{D}}\right| & \text{if } Q_2 < \left|T_{ij}^{\mathsf{D}}\right| < P_{99}, \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \neq j, \tag{9.1}$$

where Q_2 and P_{99} are the median and 99th percentile[1] of $\left|T_{ij}^{\mathsf{D}}\right|$.

Figure 9.5 shows a representation of $\mathbf{W}$ in the form of a 3D graph, for a frequency around 250 Hz. Nodes are colored by their position along the longitudinal axis, as in Figure 9.1a, and edge weights are represented by their width and color. At this frequency, links are very local. While the aircraft's rear shows some strong connections, the most distant ones are very weak. Again, this is only a snapshot of a full 3D view allowing to move the camera into the structure and focus on specific zones. This representation could be further improved by carefully adjusting the thresholds, set of colors, and ranges of edge widths.

9.2.2 . Attenuation through the structure

An interesting thing to evaluate is how the nodes react to an excitation originating from the engine, and how much their response is coherent with direct transfer values.

This is presented in Figure 9.6, where the graph is superposed to a rough aircraft frame drawn to ease the visualization. Contrary to Figure 9.5, nodes' diameter and color change with their normalized response to the excitation (using a min-max normalization, see Eq. (3.2)), here imposed at the left engine mount. Edges' width and color are constant, but their opacity changes with the weight. Three frequencies are used, representing three different modal behaviors.

[1] As 50% of values are above the median, 99% of values are below the 99th percentile.

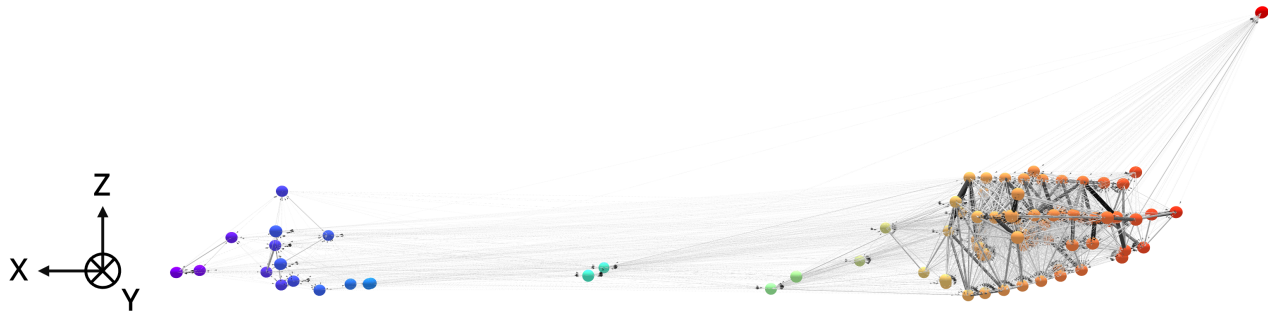
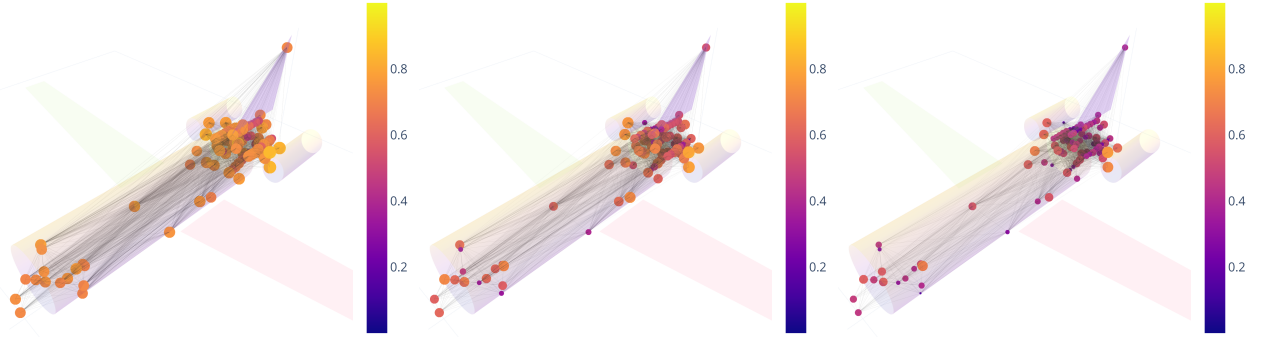


Figure 9.5: Graph representation of the direct transfer matrix $\mathbf{T}^{\mathrm{D}}$ at a given frequency transformed into a weighted adjacency matrix $\mathbf{W}$. Edges' width and color denote their weight: wider and darker means more weight. Nodes' color denotes their position in the structure along the longitudinal axis, as in Figure 9.1a (NB: a left-handed coordinate system is used).



(a) ∼10 Hz (b) ∼150 Hz (c) ∼250 Hz

Figure 9.6: Response of degrees of freedom to an excitation at the left engine mount at 3 frequencies. Node color and diameter denote the normalized response. For edges, only the opacity varies with their weight; their color and width are fixed.

For the low frequency, links are global, and most of the nodes have a high response. For the mid frequency, the excitation is much more attenuated, and mostly seen around the rear of the aircraft, with weaker links. For the high frequency, the effect is even stronger, and several nodes at the rear show an attenuated response. Some nodes in the front have higher values than nodes at the rear, which may be informative of a direct transmission path or another phenomenon. Results are coherent with the ones obtained previously in Section 8.1, showing that the attenuation increases with the frequency and distance to the source.

To further assess the connection between attenuation and direct transfers, one could try to reconstruct the values using direct transfers and the values of neighboring nodes, as performed later in Section 10.3.4.2 to evaluate a learned graph structure.

9.3 . Extracting propagation paths

This section proposes an answer to the problem of identifying the vibration propagation paths between any two given points, with a method based on the notion of direct transfer. First, the relative contributions of several paths are analyzed with regard to their frequency. Then, an iterative algorithm to trace back to the excitation source is proposed, and applied on 3 target points at the rear, center, and front of the aircraft.

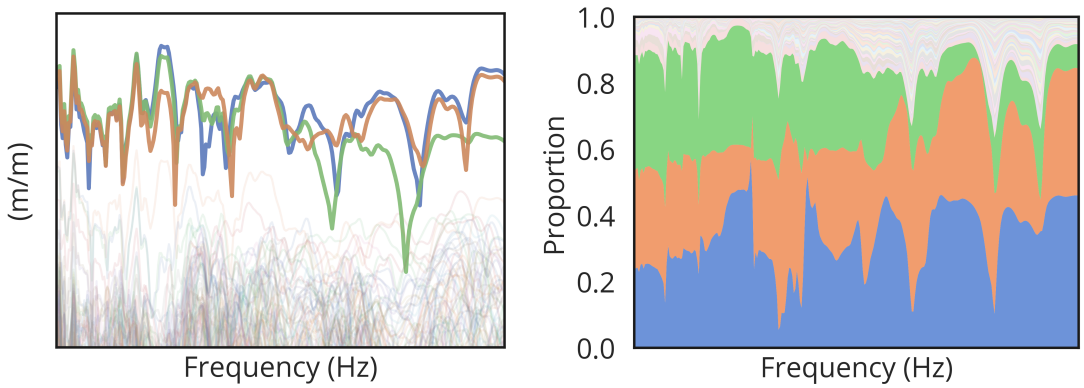
9.3.1 . Relative frequency contributions of paths

Let us consider a point at the rear of the aircraft, attached to the frame through three beams, close to the engine. This point is illustrated in Figure 9.1b as an orange triangle, where the three attachment points are blue squares.

The global transfer between the left engine j and this point i can be written as a sum of paths, see Eq. (8.31). The path C passing through a node k is defined as follows:

$$C_{ij}^{(k)} \triangleq T_{ik}^{\mathsf{D}} T_{kj}^{\mathsf{G}}, \quad \forall i \neq j. \tag{9.2}$$

Ranking the path by their average broadband contribution[2] allows to identify the main propagation paths. This is presented in Figure 9.7, highlighting the 3 main paths. It shows that more than 80% of vibrations go through these 3 points for almost all frequencies, corresponding to each attachment point.



(a) Absolute value of each path

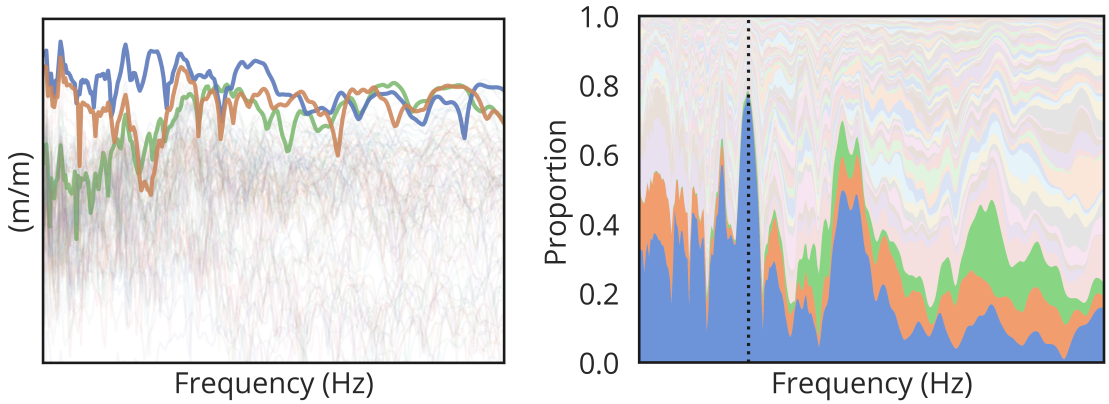
(b) Proportion of each path

Figure 9.7: Highlight of the 3 main paths from the source j to the point i.

From this, it is possible to iterate and select the highest contributor $\arg\max_k C_{ij}^{(k)}$, corresponding to the blue path, and set this as the target point i. The other degrees of freedom are removed here, as they are not of interest.

The result is given in Figure 9.8, showing that the 3 main paths stand out less, which is expected as the new starting point is closer to the aircraft skin. A very interesting result here is that it is possible to say that 80% of the vibrations go through the blue path at the frequency identified by a dotted

[2] Using a broadband value is a choice that is later confronted with alternatives.



(a) Absolute value of each path

(b) Proportion of each path

Figure 9.8: Highlight of the 3 main paths from the source j to the new target i, being the main path at the previous step. The dotted line shows a frequency at which 80% of vibrations pass by the blue path.

line. This has important applications in vibration characterization, as certain noises and vibrations that must be suppressed are very localized in frequency. By repeating these steps multiple times, a complete path corresponding to the main vibration propagation path can be identified. This is detailed in the following.

9.3.2 . Main path identification

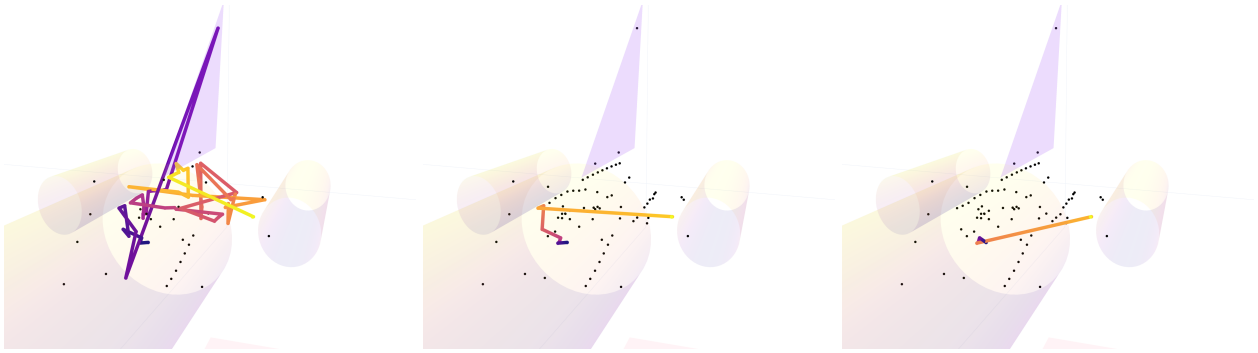
This methodology presented previously can be turned into an iterative algorithm to identify the main vibration path to the source. Algorithm 5 proposes an implementation allowing to iteratively go back to the source j from a target i. The set of all degrees of freedom is denoted S, and the path is described as a list denoted P. At each step, the next degree of freedom is identified using a ranking algorithm. The previous target is added to the list P, and removed from the set S as it does not have to be explored again. The algorithm stops when the source j is reached.

Algorithm 5 Iterative main propagation path identification

Require: $\mathbf{T}^{\mathsf{G}}, \mathbf{T}^{\mathsf{D}} \in \mathbb{C}^{n \times n}$ with $n \in S$, source j, target i, ranking algorithm
  Let $P = [\emptyset]$ be an ordered empty list
  while $i \neq j$ **do**
    $P \leftarrow P + [i]$
    Compute $C_{ij}^{(k)}$ for all k in S
    Identify the main path $p = \arg\max_k C_{ij}^{(k)}$ using the ranking algorithm
    $S \leftarrow S \setminus \{i\}$
    $i \leftarrow p$
  end while

One thing to define is the ranking algorithm. Previously in Section 9.3.1, a broadband criterion

is used, meaning that the whole frequency range is considered for the ranking. An alternative is to define a frequency range, which is helpful if a phenomenon characterized by a specific frequency needs to be suppressed. In the case of Figure 9.8b, it could be interesting to consider a frequency range centered around the dotted line.



(a) ∼10 Hz (b) ∼150 Hz (c) ∼250 Hz

Figure 9.9: Propagation paths from the left engine mount to an accelerometer at the rear, for different frequencies. The path color denotes the step number.

In Figure 9.9, the paths identified for 3 frequencies are compared, from the left engine to the same accelerometer used previously. The path color denotes the step in the algorithm. Performing the ranking by considering the value at a specific frequency can make the result sensitive to noise. Thus, a frequency range of 5 Hz centered around the frequency is considered to increase robustness. A path length is defined here as the number of nodes it contains. At the low frequency, the path is long and many connections are important all over the structure, corresponding to a global behavior. At the high frequency the path is short, which is consistent with the expected local behavior. More connections are observed in the mid-frequency range, but the number of connections remains limited.

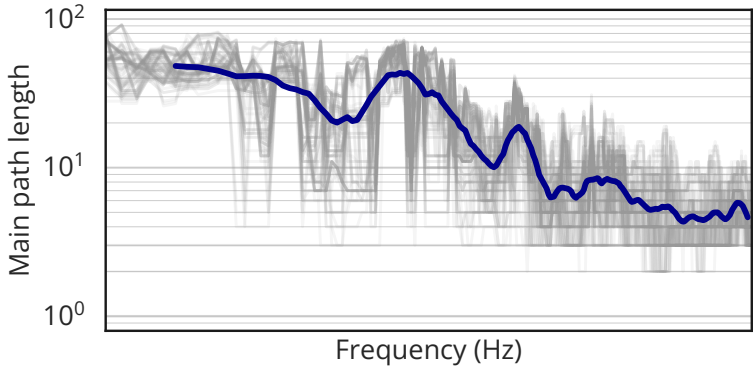
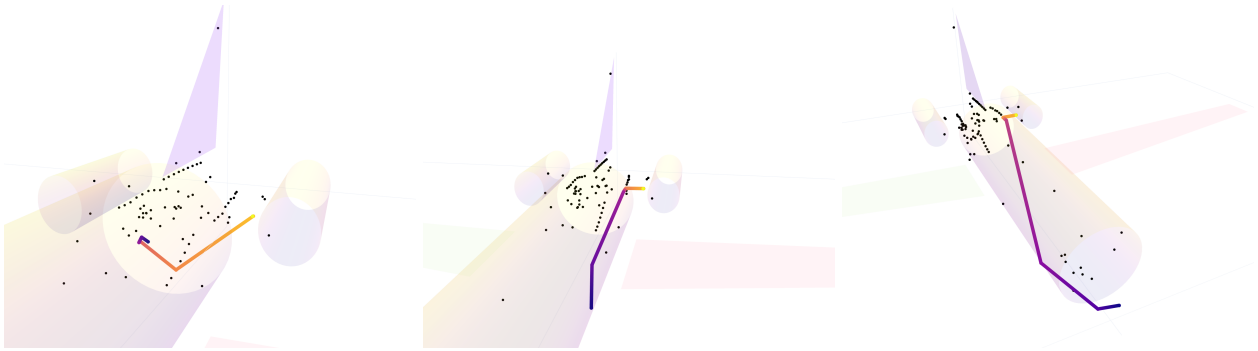


Figure 9.10: Path length measured in number of degrees of freedom, depending on the frequency criterion used, on a log-log scale.

The effect of the frequency on path length is presented in Figure 9.10, where all degrees of freedom are considered. The length only considers the number of nodes and is not weighted by the

distance or impedance value. The black line is the average across accelerometers. The trend confirms the previous observations and is consistent with the physics of propagation: the higher the frequency, the shorter the path. This is due to the fact that at high frequencies, behaviors become very local, and thus the algorithm is dealing with noise. Finally, an interesting result is that the average curve presents localized trends in frequency that seem to be shared with all accelerometers: the initial decrease in path length is followed by two peaks before finally reaching a plateau. This effect should be studied more in-depth and may result from the iterative algorithm implemented or the distance metric used. It is unclear whether it is a numerical artifact of the algorithm or if it reflects the physics. One way to explore this would be to weigh the distance by the direct transfer or impedance values.



(a) Rear (b) Center (c) Front

Figure 9.11: Propagation paths from left engine mount to various accelerometers, considering a broadband path. The path color denotes the step number.

Using a **broadband criterion** is thus a more robust way to proceed as it consists in considering the complete frequency range to perform the ranking. In Figure 9.11, Algorithm 5 is applied using such a criterion, for 3 accelerometers at the rear, center, and front of the aircraft. Their path length is, respectively, 8, 7, and 8 degrees of freedom, corresponding to the plateau of Figure 9.10. Qualitatively, the paths are consistent with domain knowledge. For example, the main path for the accelerometer at the rear follows the frames from below before arriving at the left engine mount, which looks like a shortest path. However, it looks like the left degree of freedom on the frame is skipped, whereas it is included in the main path for the other accelerometers.

9.4 . Conclusion and perspectives

In this Chapter, the challenge of vibration propagation paths identification is tackled. To do so, the tools introduced in Chapter 8 are used: a Finite Element model is reduced into a Dynamic SuperElement, from which direct transfers are computed using Advanced Transfer Path Analysis.

After presenting the computation of direct transfers, two domain-oriented 3D visualizations are provided. Results are consistent with those presented in Section 8.1, which are based on engineering knowledge. Then, propagation paths are studied in terms of their direct transfer, and an iterative algorithm allowing to extract the main propagation path between two points is implemented. This

algorithm can be used with any given path ranking criterion. Frequency-specific, as well as broadband criteria are considered. The paths identified with the method are in good agreement with the expected physics. With the 3D visualizations proposed, results are easily interpretable. Moreover, it is possible to quantify at each frequency the contribution of neighboring degrees of freedom.

The method is still at an exploratory stage. Only the modulus has been considered so far, but it could be useful to take into account phase information as well. The identified paths are based on the concept of direct transfer, which needs to be tied more closely to physics. This method could be applied to experimental data through the use of model developed in Section 6.1, or on a learned graph, as presented in Chapter 10. The methodology provided here is thus a first step to exploiting a structural representation of an aircraft and better understanding how vibrations propagate.

Finally, the ranking algorithm could be improved, as well as the iterative algorithm, and allow to extract more than the first path. For example, one could find the shortest on a graph built from a direct transfer matrix using Dijkstra's algorithm [Dij59]. Finding the minimum spanning tree on such a graph could highlight the most connected nodes in the aircraft, see Section 10.1.2.

10 - Graph signal processing for structure learning

Contents

A natural way of representing an array of sensors is through graphs, which can represent data on a possibly irregular domain. In the previous chapter, direct transfers computed from a numerical model can be treated as a graph. Here, the question is how can this be linked with experimental data.

Graph theory is introduced in this chapter. Then, the field of Graph Signal Processing is presented, allowing to extend signal processing tools to graph domains. A specific tool is of interest here: learning a graph structure from experimental data collected at various points in the aircraft. They are applied in Section 10.3, where the results of a Graph Signal Processing algorithm are compared to an *ad-hoc* correlation-based graph. This work has been presented in [FNM+22].

10.1 . Graph theory

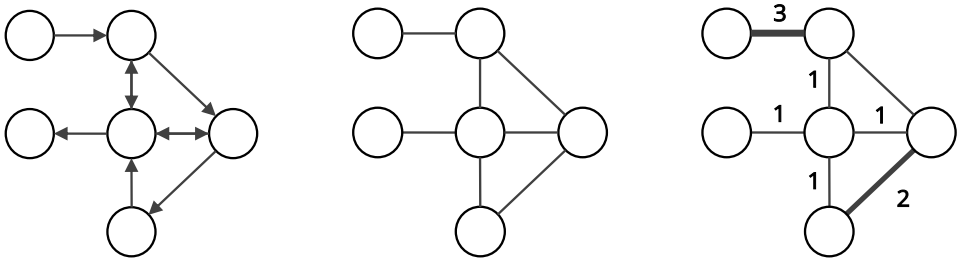
Graph theory is a branch of mathematics that deals with the study of graphs, which are structures used to represent relationships between objects. Graphs are used to model various real-world scenarios, such as social networks [HYL18, LG21] or transportation networks [Bar11]. This section provides a brief introduction to graphs.

10.1.1 . Formal definition

A graph can be defined as a set of nodes (or vertices), linked by a set of edges. Let $\mathcal{G}$ be a **directed graph** of size N. It consists of:

- A set of nodes/vertices such that $\mathcal{V} = \{1, ..., N\}$. The size of $\mathcal{G}$ is the number of nodes.
- A set of edges/links that connects pairs of nodes such that $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$.
- A set of edge weights such that $\mathcal{W} : \mathcal{E} \to \mathbb{R}$.

An **undirected graph** is a directed graph whose edge set is symmetric: $\forall (u, v) \in \mathcal{E}, (v, u) \in \mathcal{E}$. A **weighted graph** has weights $w_{ij} \in \mathbb{R}$ assigned to each edge, quantifying the importance of the connection between nodes. If $w_{ij} \in \{0, 1\}$, it is a **binary graph**. An illustration of different types of graphs is given in Figure 10.1. In this section, only undirected graphs with no self-loops are considered, i.e., $\forall u \in \mathcal{V}, (u, u) \notin \mathcal{E}$, with a maximum of one edge for each pair of nodes.



(a) Directed graph (b) Undirected graph (c) Weighted graph

Figure 10.1: Representation of different types of graphs.

A graph can be described by its weighted adjacency matrix $\mathbf{W}$, its degree matrix $\mathbf{D}$, and its Laplacian matrix $\mathbf{L}$. The **weighted adjacency matrix** $\mathbf{W} \in \mathbb{R}^{N \times N}$ contains edges weights such that:

$$W_{ij} \triangleq \begin{cases} \mathcal{W}(i, j) & \text{if} (i, j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{10.1}$$

The degree of a node corresponds to the sum of connected edges' weights. For a binary graph, it is the number of nodes to which it is connected. The **degree matrix** $\mathbf{D} \in \mathbb{R}^{N \times N}$ contains all the degrees arranged diagonally such that:

$$D_{ii} \triangleq \sum_{j=1}^{N} W_{ij}. \tag{10.2}$$

Finally, the **Laplacian matrix** $\mathbf{L} \in \mathbb{R}^{N \times N}$, more specifically the combinatorial graph Laplacian, is defined as the difference between the degree matrix and the weighted adjacency matrix:

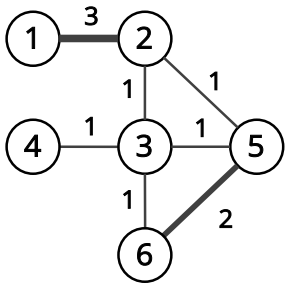
$$\mathbf{L} \triangleq \mathbf{D} - \mathbf{W}. \tag{10.3}$$

A graph is entirely defined by its Laplacian matrix. Figure 10.2 provides an example of a weighted undirected graph with its matrices $\mathbf{D}$, $\mathbf{W}$, and $\mathbf{L}$.

As $\mathcal{G}$ is defined here as being undirected with no self-loops, $\mathbf{L}$ is a real symmetric positive semi-definite matrix such that $\mathbf{x}^{\mathsf{T}} \mathbf{L} \mathbf{x} > 0 \, \forall \, \mathbf{x} \in \mathbb{R}^N$. Thus, it can be written as:

$$\mathbf{L} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^{\mathsf{T}}, \tag{10.4}$$

with $\mathbf{U}$ the eigenvectors matrix of $\mathbf{L}$ and $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_N)$ the diagonal eigenvalues matrix of $\mathbf{L}$.



(a) Graph $\mathcal{G}$

$$\mathbf{D} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} 0 & 3 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 2 & 0 \end{bmatrix} \quad \mathbf{L} = \begin{bmatrix} 3 & -3 & 0 & 0 & 0 & 0 \\ -3 & 5 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & -1 & -1 & -1 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 4 & -2 \\ 0 & 0 & -1 & 0 & -2 & 3 \end{bmatrix}$$

(b) Degree matrix (c) Weighted adjacency matrix (d) Laplacian matrix

Figure 10.2: Illustration of a binary undirected graph $\mathcal{G}$ and its matrices $\mathbf{D}$, $\mathbf{W}$ and $\mathbf{L}$.

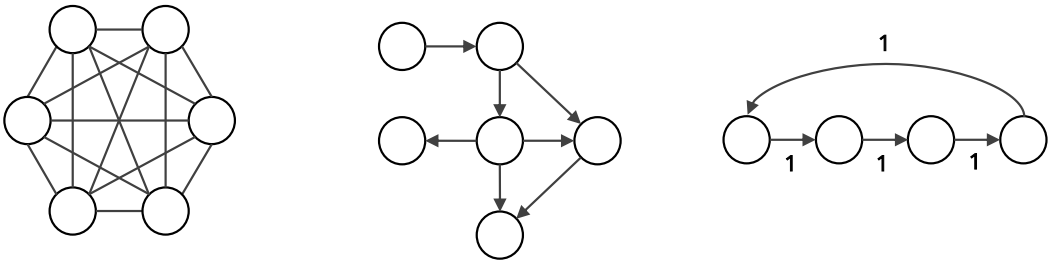
Multiple particular graphs can be defined. If all nodes are connected, it is a **complete graph**. In reality, most real networks are sparse, or at least a sparse representation is preferable. If it is a directed graph with no cycle, meaning that the same node is never encountered twice by following the edges, it is a **directed acyclic graph**. Such graphs have applications in many fields, such as biology [SPP+05] and genetics [ZGB+13], and can represent causal relationships [SGS93]. The problem of learning a directed acyclic graph from data is NP-hard [Chi96, CHM04]. However, a continuous optimization approach has been proposed in [ZARX18], which complexity has been reduced from cubic to quadratic in [DS22]. A **unitary directed cyclic graph** can be used as the support of a time-varying signal, where each node represents a time step. This type of graph is mentioned again in Section 10.2. Figure 10.3 illustrate those 3 graphs.

10.1.2 . Paths and distances on a graph

To navigate on a graph, the notions of paths and distances have to be defined. A **path** is an ordered list of nodes in $\mathcal{V}$. The length of a path can be defined as the number of elements in this ordered list. This gives a first notion of **distance**. It can be developed by taking into account edge weights.

In Chapter 9, graphs are generated and paths are extracted. The **shortest path** problem is a classic and widely studied problem in graph theory. It involves finding the shortest path between two nodes in a graph, where the length of the path is determined by the sum of the weights of its constituent edges.

Dijkstra's algorithm [Dij59], mentioned in Chapter 9, is a popular solution for the shortest path



(a) Undirected complete graph (b) Directed acyclic Graph (c) Unitary directed cyclic Graph

Figure 10.3: Particular graphs.

problem in graphs with non-negative edge weight. The algorithm works by maintaining a set of unvisited nodes and continuously picking the node with the smallest tentative distance from the start node, then updating the tentative distances to its neighboring nodes. Common alternatives to this algorithm are the Bellman-Ford algorithm [Shi55, Bel58, For56, Sho59], more versatile but slower, and the A* algorithm (pronounced A-star) [HNR68], which can be seen as an extension of Dijkstra's algorithm, adding heuristics to achieve better performances.

A **minimum spanning tree** of a weighted graph is a subset of the graph's edges that connects all the vertices without any cycles and with the minimum possible total edge weight, see Figure 10.4. In other words, it is a tree-shaped subset of the graph that includes every vertex and has the smallest possible total edge weight.

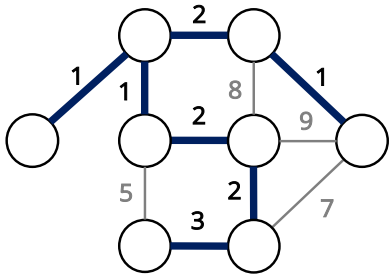


Figure 10.4: Highlight of the minimum spanning tree of a graph.

There are several algorithms to find the minimum spanning tree of a graph, such as Prim's algorithm and Kruskal's algorithm. Prim's algorithm [Jar30] is similar to Dijkstra's algorithm, but instead of choosing the edge with the smallest weight from the start node, it chooses the edge with the smallest weight connected to the tree constructed so far. Kruskal's algorithm [Kru56], on the other hand, sorts all the edges from the lowest weight to the highest and keeps adding the lowest edges that do not form a cycle until all nodes are included.

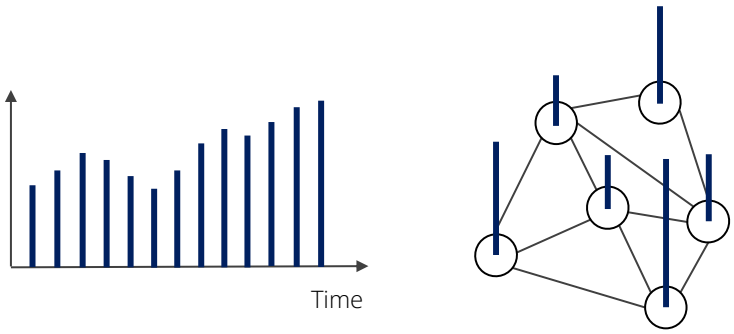
All of these tools can be used to better understand a graph, provided that its structure contains the correct information. Learning a graph structure is a possibility given by Graph Signal Processing.

10.2 . Graph Signal Processing

GSP (Graph Signal Processing) [SNF+13][OFK+18][SDS19] is a growing field that generalizes signal processing to graph domains. It extends concepts such as Fourier transform and filters to graphs and allows, under some hypothesis, to learn a graph structure given a set of signals. This section introduces the tools of GSP before focusing on the graph learning methodologies.

10.2.1 . Tools

A graph signal is defined as a function $x : \mathcal{V} \to \mathbb{R}^N$ that assigns a scalar value to each node. It can be represented as a vector $\mathbf{x} \in \mathbb{R}^N$ with x_i the function value at node i. An illustration of a graph signal is given in Figure 10.5.



(a) Temporal signal (b) Graph signal

Figure 10.5: Two signals representation, through time, and on a graph. The height of vertical blue lines denotes the signal value.

By using the eigendecomposition of the Laplacian matrix, see Eq. (10.4), the **Graph Fourier Transform** of a graph signal $\mathbf{x}$ can be defined:

$$\hat{\mathbf{x}} \triangleq \mathbf{U}^\mathsf{T}\mathbf{x}. \tag{10.5}$$

The components of $\hat{\mathbf{x}}$ are interpreted as Fourier coefficients, the eigenvalues λ of $\mathbf{\Lambda}$ as distinct frequencies, and the eigenvectors $\mathbf{u}$ of $\mathbf{U}$ as a decomposition basis.

A signal lying on a unitary directed cyclic graph such as the one presented in Figure 10.3c is equivalent to a classical temporal signal. Applying a graph Fourier transform on such a signal produces the same result as applying a Fourier transform on the temporal signal. Indeed, the eigenvector decomposition of a Laplacian matrix gives a Fourier matrix [SMLR16, HGW+16]. However, as it can be applied to any graph describing an irregular domain, its interpretation can be of a spatial decomposition of the signal. The frequency, described by eigenvalues, gives information about variations in the graph space instead of time. From the definition of the graph Fourier transform can be defined spectral sparsity and smoothness.

A graph signal x admits a $k \in \mathbb{N}^+$ **sparse spectral representation** with respect to a graph $\mathcal{G}$ if for $\hat{\mathbf{x}} = \mathbf{U}^\mathsf{T}\mathbf{x}$, the number of non-zero elements in its Fourier coefficient vector is equal to k:

$$\|\hat{\mathbf{x}}\|_0 = k, \tag{10.6}$$

where $\|\cdot\|_0$ is the l^0-pseudonorm denoting the number of non-zero elements[1]. This definition can also be referred to as the **bandlimitedness** of the graph signal. Spectral sparsity can be associated with the number of clusters on a graph [von07, SBL19], and to the sampling problem on a graph [CVSK15, TEOC20], where k is related to number of nodes required to reconstruct a signal properly.

The **smoothness** of a signal $\mathbf{x}$ on a given graph describes the notion that signal values taken on adjacent nodes should take similar values, and can be written as:

$$\text{smoothness}(\mathbf{x}) \triangleq \left\|\mathbf{L}^{1/2}\mathbf{x}\right\|_2^2 = \mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} W_{ij}(x_i - x_j)^2, \tag{10.7}$$

where a lower value denotes a smoother signal. The smoothness of an eigenvector $\mathbf{u}$ of $\mathbf{L}$ is its eigenvalue λ:

$$\text{smoothness}(\mathbf{u}_i) = \mathbf{u}_i^\mathsf{T}\mathbf{L}\mathbf{u}_i = \lambda_i. \tag{10.8}$$

As in signal processing, **filters** can be defined on a graph. The transfer function between a graph signal $\mathbf{x}$ and its filtered counterpart $\mathbf{y}$ is the weighted sum of the Laplacian matrix at various powers:

$$\mathbf{y} = \left(\sum_{i=0}^{p} h_i \mathbf{L}^i\right)\mathbf{x}, \tag{10.9}$$

which is equivalent to applying the Laplacian operator multiple times, and can be seen as a diffusion process on a graph. Equivalently in the graph frequency domain :

$$\hat{\mathbf{y}} = \left(\sum_{i=0}^{p} h_i \lambda^i\right)\hat{\mathbf{x}}. \tag{10.10}$$

Since $\mathbf{\Lambda}$ is diagonal in Eq. (10.4), filtering in the graph frequency domain is equivalent to computing:

$$\hat{y}_j = \widetilde{\lambda}_j \hat{x}_j, \tag{10.11}$$

where $\widetilde{\lambda}_j = \sum_{i=0}^{p} h_i \lambda_j^i$. Thus, filtering is done by building a function h that acts on the eigenvalues such that:

$$\widetilde{\lambda}_j = h(\lambda_j). \tag{10.12}$$

10.2.2 . Structure learning

The problem of interest here is to learn the structure of a graph based on node values. To this end, multiple methods based on various assumptions are presented. In addition, the common assumption is graph-stationarity, meaning that the graph does not change over time.

In this chapter, the algorithm **GL-SigRep** (Graph Learning for Smooth Signal Representation) proposed in [DTFV16] is applied. Based on a smoothness assumption, a Laplacian matrix is learned such

[1] In practice l^0 is not even a pseudonorm as it does not satisfy the scalability property.

that the graph signal is smooth on the learned graph. In practice, it consists of learning, from a measured signal $\mathbf{X}$, the Laplacian matrix $\mathbf{L}$ and a smooth signal reconstruction $\mathbf{Y}$ using a least square optimization under regularization constraints:

$$\min_{\mathbf{L},\mathbf{Y}} \|\mathbf{X}-\mathbf{Y}\|_{\mathrm{F}}^2 + \alpha \mathrm{tr}(\mathbf{Y}^\mathsf{T}\mathbf{L}\mathbf{Y}) + \beta \|\mathbf{L}\|_{\mathrm{F}}^2, \quad \text{s.t.} \begin{cases} \mathrm{tr}(\mathbf{L}) = N, \\ L_{ij} = L_{ji} \leq 0, \quad i \neq j, \\ \mathbf{L}\,\mathbf{1} = \mathbf{0} \end{cases} \tag{10.13}$$

where α and β are two positive regularization hyperparameters. The first one is related to the smoothness, and the second one is a Tikhonov regularization, see Section 5.3.1. Moreover, $\mathbf{1}$ and $\mathbf{0}$ are constant unitary and null vectors, while $\mathrm{tr}(\cdot)$ and $\|\cdot\|_{\mathrm{F}}$ are respectively the trace and the Frobenius norm of a matrix. Regarding the constraints, the first one acts as a normalization factor to avoid trivial solution, while the second and third constraints guarantee that $\mathbf{L}$ is a valid Laplacian matrix.

As this optimization problem is not jointly convex in $\mathbf{L}$ and $\mathbf{Y}$, an alternating minimization scheme is used: $\mathbf{Y}$ is first initialized to be equal to the observed signal $\mathbf{X}$, and $\mathbf{L}$ is determined by Eq. (10.14). Then, $\mathbf{L}$ is fixed and $\mathbf{Y}$ is determined by Eq. (10.15). Finally, graph weights with a value under 10^{-4} are set to zero using a hard threshold.

$$\min_{\mathbf{L}} \alpha \mathrm{tr}(\mathbf{Y}^\mathsf{T}\mathbf{L}\mathbf{Y}) + \beta \|\mathbf{L}\|_{\mathrm{F}}^2, \quad \text{s.t.} \begin{cases} \mathrm{tr}(\mathbf{L}) = N, \\ L_{ij} = L_{ji} \leq 0, \quad i \neq j, \\ \mathbf{L}\,\mathbf{1} = \mathbf{0} \end{cases} \tag{10.14}$$

$$\min_{\mathbf{Y}} \|\mathbf{X}-\mathbf{Y}\|_{\mathrm{F}}^2 + \alpha \mathrm{tr}(\mathbf{Y}^\mathsf{T}\mathbf{L}\mathbf{Y}). \tag{10.15}$$

Alternatives exist, such as **ESA-GL** (Estimated-Signal-Aided Graph Learning) proposed in [SBL19]. Instead of using a smoothness assumption, it is based on a sparsity assumption. A Fourier basis is learned instead of a Laplacian matrix, and the algorithm has a complexity of at least $\mathcal{O}(N^8)$.

Finally, **IGL-3SR** (Iterative Graph Learning for Smooth and Sparse Spectral Representation) is an algorithm proposed in [LBHOK19, HBO+21], based on both a smoothness and a sparsity assumption. It aims to learn the Laplacian matrix and a smooth bandlimited approximation of the observed signal. **FGL-3SR** (Fast Graph Learning for Smooth and Sparse Spectral Representation), proposed also in [LBHOK19, HBO+21], uses relaxation to accelerate the resolution, and offers a complexity of $\mathcal{O}(N^5)$.

10.3 . Applying graph structure learning on vibration data

The tools previously introduced are now applied to experimental data. Given a set of experimental measurements, the goal is to learn the underlying structure linking accelerometers.

10.3.1 . Dataset generation

A dataset similar to the one used in Section 7.1 is extracted. The aircraft is flying, and only stabilized flight phases are considered. Broadband separation is applied, see Section 4.1, and the signals are reduced into RMS levels per frequency band, see Section 4.2.2. Each frequency band is considered independently: the matrix $\mathbf{X} \in \mathbb{R}^{m \times N}$ contains the values of $N = 36$ accelerometer channels on

$m = 170$ flight points for a given frequency band B. The position of the 12 tri-axial accelerometers is given in Figure 10.6.

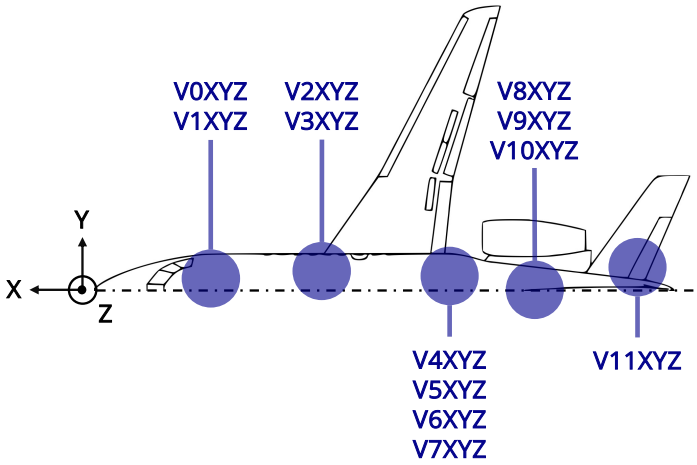


Figure 10.6: Position of the accelerometer channels considered in this study (NB: a left-handed coordinate system is used).

A graph signal is written $\mathbf{x} = X_{t:}$ and denotes the t-th row of $\mathbf{X}$, thus nodes values at time t. A temporal signal is written $X_{:p}$ and denotes the p-th column of $\mathbf{X}$ thus the values of accelerometer p through time. Before being used for the graph learning, $\mathbf{X}$ is log-transformed and normalized using a min-max normalization, see Eq. (3.2).

10.3.2 . Learning algorithm

The learning problem is the following: given a multivariate time series $\mathbf{X} \in \mathbb{R}^{m \times N}$, the goal is to learn a graph structure, thus the link between each node representing an accelerometer. A graph stationarity hypothesis is made: the graph is assumed to be invariant for all m signals on the graph. For each frequency band, the goal is thus to learn an undirected graph with positive weights.

The GL-SigRep algorithm is used, allowing to learn the Laplacian matrix $\mathbf{L}$ of a graph $\mathcal{G}$ for each frequency band B based on the assumption that signal $\mathbf{x}$ is smooth on the graph $\mathcal{G}$. Two regularization parameters α and β must be set before the optimization process. When $\alpha = \beta = 0$, a completely connected graph is learned with equal weights on each node. Outside of this case, only the β/α ratio has a significative influence [DTFV16]. Here, it is fixed at 1, with $\alpha = \beta = 10^{-6}$.

As a comparison point, graphs built from accelerometer correlations are also used. This method assumes no model of the data and is purely based on observation, and may thus be sensitive to the noise it contains [DTRF19]. If Pearson's correlation coefficient ρ between two accelerometers, see Eq. (4.6), is higher than a threshold λ, then the nodes representing those accelerometers are connected with a weight equal to the correlation value:

$$W_{ii} = 0, \quad W_{ij} = \begin{cases} \rho\left(X_{:i}, X_{:j}\right) & \text{if } \rho\left(X_{:i}, X_{:j}\right) > \lambda, \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \neq j. \tag{10.16}$$

The lower the threshold λ, the more connected the graph, and each node depends on a higher number of neighbors. A value of $\lambda = 0.8$ is chosen as a compromise between a low number of connections and a low reconstruction error.

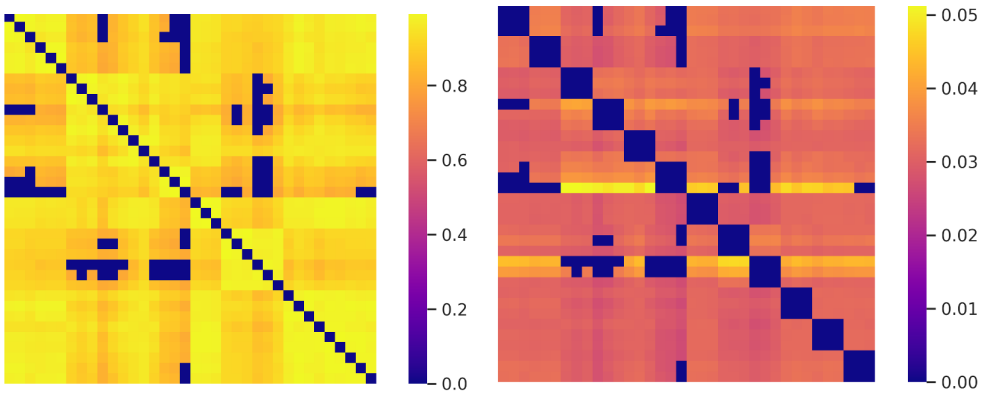
10.3.3 . Evaluation through signal reconstruction

For the learning process, 80% of the dataset is used as a training set and 20% as a test set, using a random split.

In the absence of a specific protocol to evaluate the learned structure, a reconstruction task is defined. The goal is to reconstruct node values from neighbor node values. By assuming a linear system satisfying the Markov property[2], the reconstruction in a given frequency band can be defined as:

$$\hat{\mathbf{X}} = \mathbf{X}\,\mathbf{W}_{\text{adjusted}}^{\mathsf{T}}, \tag{10.17}$$

with $\hat{\mathbf{X}}$ the reconstructed signal, $\mathbf{X}$ the measured signal, and $\mathbf{W}_{\text{adjusted}}$ the adjusted weighted adjacency matrix, obtained from $\mathbf{W}$ by setting connections between the different axes of each accelerometer to zero (so that the reconstruction of each accelerometer is not based on any information related to that accelerometer). It is then normalized by rows. Figure 10.7 shows the transformation from $\mathbf{W}$ to $\mathbf{W}_{\text{adjusted}}^{\mathsf{T}}$.



(a) $\mathbf{W}$ (b) $\mathbf{W}_{\text{adjusted}}^{\mathsf{T}}$

Figure 10.7: Transformation of the weighted adjacency matrix built from correlations in the band B_{17}.

The reconstruction quality is then assessed using the MALE metric, see Eq. (5.34), applied on unnormalized data.

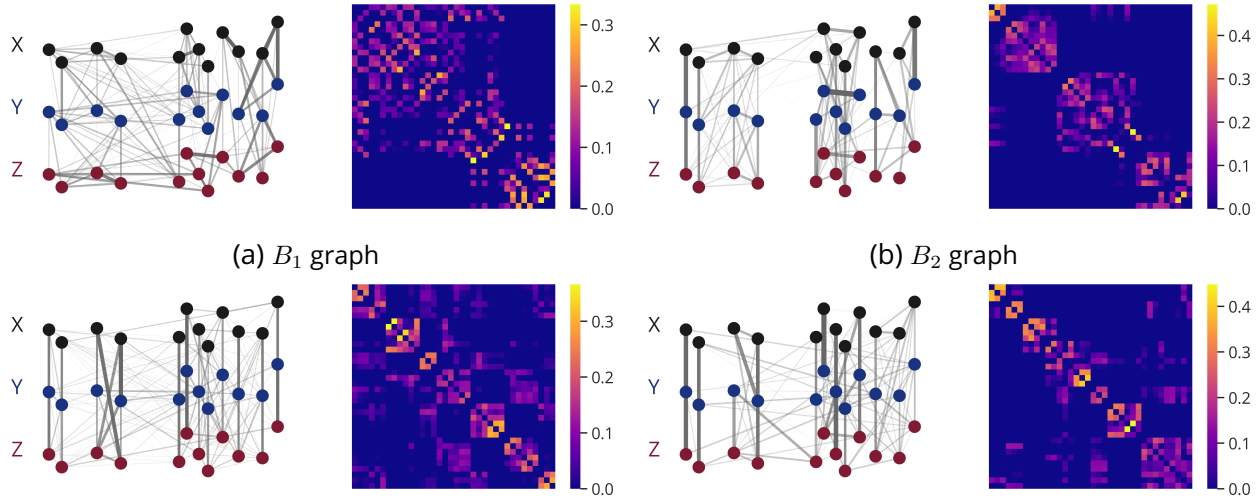
10.3.4 . Results

Two types of results are presented: first, a qualitative analysis of the learned graphs, and second, a quantitative analysis of signal reconstruction.

[2] Past information is not useful to predict the present

10.3.4.1 . Graphs evaluation

Figure 10.8 shows the graphs learned for 4 different frequency bands, exhibiting the 4 main regimes observed. Nodes are disposed horizontally according to their position along the structure, and the vertical separation corresponds to different axes of a same accelerometer. The width and opacity of edges are proportional to their weight. The weighted adjacency matrix is represented to the right of each graph. Node numbering is done according to their position: if i is close to j, then W_{ij} is the weight between two spatially close nodes. Edges are sparse, and the structure of the graph is strongly dependent on the frequency band. Below, an analysis of graphs presented in Figure 10.8 is proposed.



(a) B_1 graph (b) B_2 graph

(c) B_{12} graph (d) B_{17} graph

Figure 10.8: For 4 frequency bands, graph $\mathcal{G}$ learned with the smoothness assumption (left), with its weighted adjacency matrix $\mathbf{W}$ (right). The horizontal axis represents the position of the accelerometer in the aircraft. The three axes of each accelerometer are stacked vertically and represented with different colors. Edges' width and opacity denote their weight. In the weighted adjacency matrix, nodes are ordered along their position in the structure from front to rear.

- The first frequency B_1 contains the global structural modes. The learned graph shows connections that are independent of the distance, which is consistent with the global nature of modes. This is also visible in $\mathbf{W}$ where weights are not grouped around the diagonal. The different axes of a single sensor are rarely connected.
- Conversely, the second frequency band B_2 contains more local modes. Thus, connections are more localized in the structure, and 3 clusters are identified which roughly correspond to the front, middle, and rear of the aircraft. These 3 groups of sensors in the same zones having a similar response can be identified in $\mathbf{W}$.
- The frequency band B_{12} is more difficult to interpret in terms of relevant phenomena. Connections are mainly between the different axes of a single sensor, or between close sensors, with

some connections between distant sensors. The connections appear to be less related to the horizontal position of the accelerometer in the aircraft.

- Finally, the frequency band B_{17} corresponds to random excitations originating from the aerodynamic field. Even more than for the previous band, connections are very localized and are mostly between the different axes of a single sensor. This can be seen on the graph through the vertical links or on the matrix by the weights being close to the diagonal of $\mathbf{W}$.

Even though no physical constraint has been included explicitly in the optimization process, the learned graphs are consistent with the physics of the different frequency bands.

10.3.4.2 . Reconstruction performances

Using the previously defined metric MALE, see Eq. (5.34), it is possible to assess the reconstruction quality of an accelerometer axis given the values of other accelerometers. In Table 10.1, the average error over the different accelerometers and frequency bands is given for each graph learning method and dataset.

	Correlations	Smoothness
Training set (%)	7.67	4.55
Test set (%)	7.78	4.60

Table 10.1: MALE, see Eq. (5.34), averaged over accelerometers and frequency bands for each graph learning method and dataset. Lower values are better.

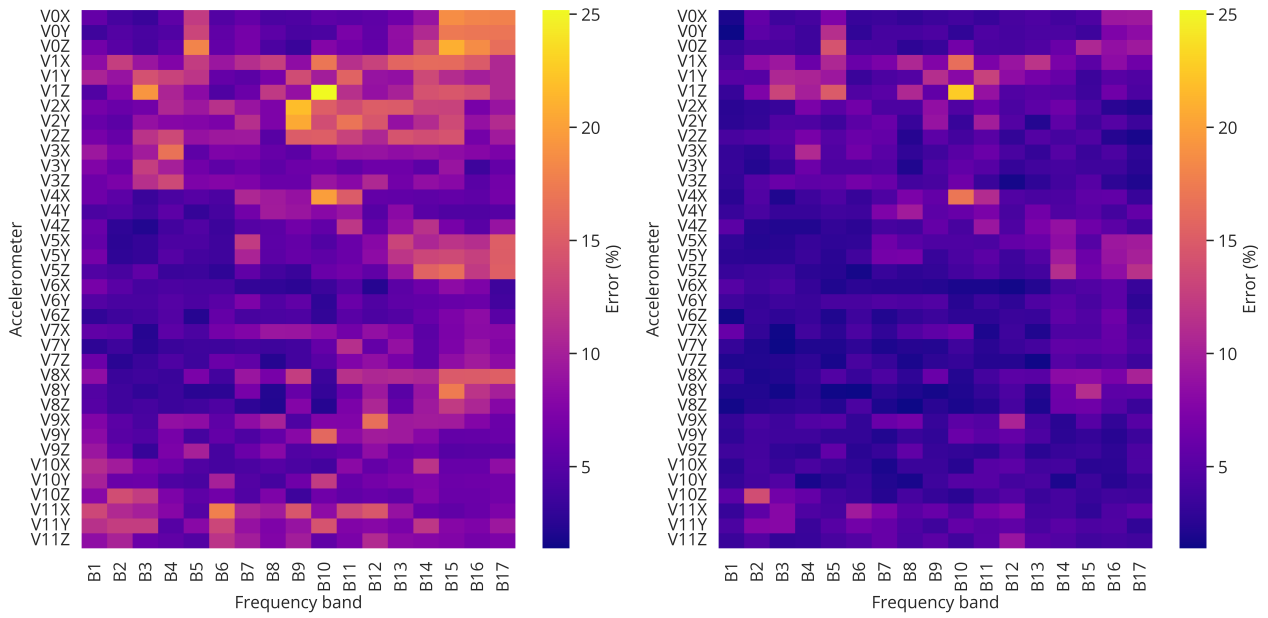
The differences in performance between the training and test sets are negligible. Graphs learned using the smoothness assumption allow significantly better reconstruction and reduce the error of roughly 40% compared to correlation-based graphs.

Figure 10.9 details the results for each accelerometer and frequency band. Correlations-based graphs are prone to higher errors in high frequencies, a phenomenon not observed on smoothness-based graphs. Accelerometers V1XYZ and V4X seem particularly hard to reconstruct, along with the frequency band 430 to 536 Hz.

Although some reconstruction problems are shared between the two methods, the smoothness-based one seems to be less dependent on the accelerometer and frequency band than the correlations-based one. The good performances are notable given the order of the measurement uncertainties present in the data.

10.4 . Conclusion on graph learning

A graph learning-based approach is implemented to reconstruct the vibration environment within the structure of an aircraft. The method directly extracts relevant connections in the data without taking into account explicit physical constraints and performs significantly better than the *ad-hoc* correlations-based method.



(a) Correlations-based

(b) Smoothness-based

Figure 10.9: Reconstruction performances on the test set depending on the graph learning algorithm: correlations-based (left) or smoothness-based (right). Lower values are better.

Two avenues for improvement are possible. The first one is related to graph learning, which could be based on different assumptions. Using ESA-GL [SBL19] would replace the smoothness assumption with a sparsity assumption, while using IGL-3SR or FGL-3SR [LBHOK19, HBO+21] would add a sparsity assumption to the smoothness assumption. Frequency bands are treated separately, but could be incorporated into a single graph to model their relationships and potentially enhance reconstruction performances. Additionally, physical information could be included within the learning assumptions to add new connections or even nodes within a graph without measurements.

The second avenue for improvement concerns the reconstruction method. The linearity assumption could be compared with other regression methods, particularly those tailored for graph-based data [NGO13], [RMG17].

Perspectives include extending this to other types of data, such as ground test experimental data or even numerical data generated from a DSE, see Section 8.3. Instead of learning graphs on broadband signals, kinematic lines from a single source could be used as in Chapters 6 and 9. The model developed in Section 6.1 can be used to generate data to inject into the DSE. The link between a learned graph weighted adjacency matrix and a system equivalent impedance or direct transfer matrix could be investigated.

Hence, the proposed method opens the door to various developments and could lead to predictive models of the vibration environment throughout the structure of an aircraft. From an industrial point of view, such tools could also be used to reconstruct missing or faulty values of accelerometers during flight tests.

Summary of Part III

In this part, the framework presented in Parts I and II is extended to the **prediction at any point of the aircraft structure**.

In Chapter 8, **numerical models** adapted to the low-, mid-, and high-frequency ranges are presented. While being the primary solution to model vibrations in the industry, they are not the solution chosen for the problem at hand because of their limitations. To study vibration propagation paths, **Dynamic SuperElements** are used for model reduction and are analyzed using the formalism of **Advanced Transfer Path Analysis**. Results are presented in Chapter 9. An algorithm enabling main path identification is proposed, and provides results coherent with domain knowledge.

In Chapter 10, experimental data generated by the Flight Test Center is used again. **Graphs** can be a natural way of representing an array of sensors and are introduced along with **Graph Signal Processing**, allowing to manipulate signals on graphs. These tools are applied to the case of **learning a graph structure from data**. They provide a way to reconstruct the data based on identified connections, which seem consistent with structural paths. The reconstruction quality is superior to that obtained with ad-hoc correlation-based methods. This work has been presented in [FNM+22].

Conclusion

This thesis proposes different ways to answer the challenge of predicting the vibration environment of an aircraft through a multidisciplinary data-driven approach. Various tools from machine learning and Graph Signal Processing are combined with expert knowledge in order to construct analysis and predictive tools. The main contributions of the manuscript are summarized below.

First, a methodology is presented for transforming test data from its voluminous and hard-to-handle original state. It allows us to define a robust framework for the data-driven study of vibration environment. Several efficient low-order representations are proposed to provide robust and domain-interpretable ways to transform and compress the data. The introduced spectral separation is an essential part of these transformations, and the hypothesis is confirmed by experimental results. This process, detailed as a set of algorithms, is implemented in a software prototype written in Python. Tools from the field of unsupervised learning, such as clustering, are shown to be an efficient way to generate physical insight and understand the cause of vibration. They can be used to identify the most important features for modeling.

Prediction of the vibration environment at any point in the flight domain/feature space is addressed by applying machine learning to experimental data generated at the Flight Test Center. Kinematic lines are predicted during ground tests, and the prediction is found to be robust to test conditions. A methodology is proposed to transpose this prediction during flight. This method is exploratory, but has important implications. It could allow the prediction of non-instrumented points during flight from ground test data. Broadband spectra are predicted in the flight domain using the spectra separation hypothesis. Prediction of stabilized flight phases is the first step that allows for results that meet industrial needs and a degree of robustness. The model is evaluated for a standard random split protocol as well as for an extrapolation protocol, which makes it possible to assess its ability to generalize. Furthermore, the interpretability of the models is studied, which is crucial in this aeronautical context. The prediction is then extended to the full spectrum for unsteady flight phases, essential for specification. A model adapted to temporal dependencies is used, and good results are obtained. Different parts of the spectrum (kinematic lines and broadband) and of the flight domain (steady/unsteady) can also be predicted independently, which could be useful to provide additional insight.

Finally, the problem of structural modeling is also addressed. Two approaches are used. First, numerical models are used to study vibration propagation. An algorithm to extract the main propagation paths is proposed and shows results consistent with the expected physics. Second, a method from Graph Signal Processing is used to learn a structure from experimental data. Several graphs are learned, one for each frequency band. Again, the results show consistency with the physics and provide excellent reconstruction results. Both approaches can be combined; they are closely related. The first approach uses Dynamic SuperElements to reduce a numerical model, but such a reduced model can be obtained from experimental data. The second approach learns a graph from experimental data, but it could be generated from a numerical model. Vibration propagation path identification can be applied on these two types of representation.

The different prediction methods presented here are building blocks that could be integrated into a comprehensive data-driven framework leveraging existing experimental measurements and numerical model results in order to improve the vibration environment prediction in early design phases. The model could then be iteratively enriched with new experimental data when the aircraft goes through successive design phases as ground and flight tests are conducted.

Because this work is primarily exploratory, several avenues for expansion are suggested. The use of machine learning in the aviation industry can certainly become a subject of standards. In this context, it is crucial to explore the interpretability of models, and predictions should be explainable and justified. SHAP is used here, but other methods should be tested. In the line of thought, robustness must be ensured, at least by showing when a model should not be relied upon, e.g., when an ensemble model shows diverging individual predictions. Using Occam's razor as a heuristic, relying on simpler models for specific contexts could be an option as long as feature engineering is performed thoroughly. Furthermore, the quantification of uncertainties is crucial to enable model confidence.

The study of vibration propagation paths needs to be further explored, using algorithms from graph theory and relying more on physics. It should be applied to data generated by other models, different systems, and also to experimental data. Since results depend on system substructuring (the set of nodes chosen), solutions implemented to tackle this issue in the context of SEA could be adapted to our problem. The field of Graph Signal Processing itself should be studied in more depth. Other methods of graph learning and interpolation on a graph need to be tested. Examining signals in the graph space could also provide a new way of interpreting data.

Other mid-frequency numerical approaches could be explored. While this thesis focused on machine learning, theories such as the variational theory of complex rays would benefit from being studied.

It is essential to study how experimental data and methods, e.g., from the field of Graph Signal Processing, can be integrated into numerical models used in the aircraft design phase. On the one hand, the numerical model contains a spatially resolved description of the structure, but its prediction is likely to contain inaccuracies, while on the other hand, the graph representation provides a correct prediction over a coarse discretization of the structure, corresponding to instrumented points. How to combine the information from both is a crucial question. Hybridization could also be used through frequency domain extrapolation: models that can predict only the low frequencies could be combined with data that gives results over the entire frequency range.

As mentioned in the first chapter, the vibration environment also takes into account shocks, which are not studied here. This would be an important thing to study and could be done using some of the methodological elements developed here.

By further exploring and developing these methods, a comprehensive data-driven approach combining theoretical, numerical, and experimental sources of knowledge could be implemented. This could be used to predict the vibration environment of a complete aircraft from the design stage, generating significant cost and time savings.

A - Résumé étendu en français

A.1 . Introduction

Les vibrations, que ce soit au sol ou en vol, sont un phénomène ressenti par tout passager empruntant l'avion. La prédiction et l'atténuation de ces vibrations sont l'un des aspects critiques du confort des passagers. Par ailleurs, les niveaux vibratoires peuvent avoir un impact significatif sur la sécurité et la durabilité de l'avion et de ses équipements [Vib12]. La prédiction de l'ambiance vibratoire d'un avion est par conséquent essentielle pour garantir des conditions opérationnelles sûres et efficaces.

L'ambiance vibratoire est un terme utilisé pour décrire les sollicitations dynamiques de 10 à 2000 Hz subies par les équipements embarqués durant l'utilisation de l'avion. Ces sollicitations sont générées par diverses sources d'excitations externes, telles que les interactions avec le champ aérodynamique, ou internes, provenant notamment des machines tournantes telles que les moteurs et les pompes. L'ambiance vibratoire doit être spécifiée relativement tôt dans le processus de conception d'un avion, qu'il soit civil ou militaire, et consiste à déterminer les niveaux de robustesse et d'endurance que l'équipement doit être capable de supporter. Alors que ces spécifications doivent être produites au moment de la conception d'un nouvel avion, l'ambiance vibratoire ne peut être déterminée qu'au moment des essais en vol, qui sont réalisés dans les dernières phases d'un programme.

L'ambiance vibratoire est mesurée à l'aide d'accéléromètres uniaxiaux ou triaxiaux avec une fréquence d'échantillonnage élevée, fixée ici à 5000 Hz. Situés au pied des équipements, ils sont reliés à une instrumentation d'essais qui permet également de mesurer les informations nécessaires à caractériser l'état de l'appareil à tout instant, appelées dans ce manuscrit paramètres généraux. Les signaux mesurés par les accéléromètres sont généralement analysés via leur contenu fréquentiel à l'aide de densités spectrales de puissance.

Il n'y a à l'heure actuelle pas de solution permettant de modéliser le comportement dynamique d'un avion complet dans la gamme de fréquence considérée, s'étendant jusqu'à 2000 Hz. Par conséquent, les spécifications sont basées sur les normes existantes et l'expérience des avions antérieurs. Cette approche peut avoir deux conséquences négatives : le surdimensionnement de l'équipement, qui a un impact négatif sur la masse totale de l'aéronef, ou un retard de qualification des équipements dans le but de démontrer leur tenue aux niveaux spécifiés.

Pour réaliser des spécifications plus précises, les limitations des **modèles numériques** doivent être contournées pour permettre de modéliser un avion complet. Différentes familles de modèles ont été développées pour modéliser le comportement dynamique dans différentes gammes de fréquences, mais aucune n'est adaptée à la problématique rencontrée ici.

Dans le domaine des basses fréquences, les techniques se concentrent sur le comportement modal pour extraire des réponses localisées en fréquence. Des méthodes déterministes telles que la méthode des éléments finis [ZTZ10] ou la méthode des éléments de frontière [BB94] sont utilisées.

En pratique, ces méthodes sont limitées aux basses fréquences car elles requièrent un niveau élevé de discrétisation en espace et en temps, et un degré élevé d'interpolation pour contrer les effets de pollution numérique et de dispersion [DBB99].

Dans le domaine des hautes fréquences, l'intérêt est cette fois de décrire un comportement global en termes d'énergie, la densité modale étant élevée. La SEA ou analyse statistique en énergie [LM62, LDD95, Smi62, LB04, CLK05] est adaptée à cette gamme de fréquences car elle modélise les transferts d'énergie vibratoire entre les sous-systèmes d'une structure. Les échanges entre les sous-systèmes sont décrits à partir d'un ensemble d'équations linéaires décrivant l'apport, le stockage, la transmission et la dissipation de l'énergie via un ensemble de coefficients à déterminer. La réponse étant moyennée en termes de fréquence et d'espace, elle n'est pas adaptée à une caractérisation locale.

Dans le domaine des moyennes fréquences, ni les approches basses fréquences ni les approches hautes fréquences ne sont appropriées : les méthodes basses fréquences deviennent trop coûteuses en raison du raffinement élevé du maillage nécessaire pour réduire l'effet de pollution numérique, et les méthodes hautes fréquences ne fournissent pas de réponses locales. Les méthodes moyennes fréquences peuvent être divisées entre les approches hybrides (combinant des méthodes basses et hautes fréquences) et les méthodes de Trefftz (approximant la solution comme une combinaison linéaire de fonctions de forme satisfaisant a priori les équations d'équilibre), cf. la revue de littérature [DOP12]. Toutefois, des travaux supplémentaires sont nécessaires pour les rendre applicables dans un contexte industriel.

Les outils de prédiction pourraient bénéficier de la richesse considérable des informations obtenues à partir des données d'essais en vol, qui ne sont généralement utilisées qu'au cours des dernières phases du programme. Le défi de cette thèse est de déterminer comment intégrer ces informations dans un outil prédictif. Parmi les questions abordées figurent la compression des données, l'extraction des variables d'intérêt et la modélisation robuste. **L'apprentissage automatique** offre une grande variété de techniques pour traiter ces différents problèmes.

Parmi les modèles d'apprentissage automatique, les réseaux de neurones artificiels se sont avérés efficaces pour prédire les vibrations dans une grande variété d'applications telles qu'en aérodynamique, exploitation minière ou génie civil. Dans le domaine temporel, l'architecture LSTM (*Long Short-Term Memory*) s'est révélée efficace pour prédire la réponse de systèmes mécaniques complexes à une excitation large bande [SCC+22, NB18]. Elle a également été en mesure de prédire les vibrations excessives de moteurs d'avion [EWHD16]. Les LSTM ont également été utilisés pour prédire l'environnement vibratoire des voitures électriques et se sont avérés plus performants que les modèles ARMAX (*Auto Regressive Moving Average with eXternal inputs*) classiques [DGD+20]. Dans le domaine fréquentiel, il a été constaté que les réseaux de neurones artificiels pouvaient prédire avec succès les contraintes structurelles [WOdS20]. Ils ont également été appliqués aux écoulements turbulents pour prédire des niveaux de bruit à partir de mesures limitées de pression dans les écoulements de jets [TGRB20], ainsi qu'au bord de fuite d'un profil aérodynamique [AF21]. Parmi les autres applications des réseaux de neurones artificiels figure la prédiction des vibrations du sol dues au dynamitage pour les applications minières [Sin04], avec des performances qui se sont avérées supérieures à celles de la régression conventionnelle [BP20]. Enfin, ils peuvent permettre de déterminer les niveaux de vibration du sol dus au passage de trains [PDCFN18].

L'objectif à terme du sujet est d'utiliser efficacement les données d'essais et les méthodes d'apprentissage automatique pour prédire l'ambiance vibratoire dans un avion complet, et ce dès la phase de conception. Pour remplir cet objectif, les étapes suivantes ont été identifiées :

- La première étape est de permettre l'extraction et l'analyse de l'important volume de données d'ambiance vibratoire généré par les essais en vol. Des représentations adaptées de ces données ainsi que des méthodes de réduction doivent être implémentées, en plus d'outils permettant la prise en compte des règles métier.
- La deuxième étape est de développer une méthodologie de prédiction de l'ambiance vibratoire en tout point de vol (ou tout point de l'espace des paramètres d'entrée), et ce en utilisant les données précédemment mises en forme ainsi que des modèles d'apprentissage automatique. Elle aurait ainsi des applications notamment en détection d'anomalies, où un modèle est comparé aux mesures, et où tout écart génèrerait une alerte nécessitant l'attention d'un ingénieur. Une autre application possible serait l'optimisation des essais en vol via l'identification des points de vol maximisant l'apport d'information.
- La troisième étape est d'étendre cette prédiction à tout point de la structure avion, y compris ceux non instrumentés par un accéléromètre. Une représentation adaptée de la structure doit alors être définie. Ceci permettrait notamment l'optimisation de l'instrumentation d'essais, où le placement des accéléromètres serait choisi de sorte à maximiser l'apport d'information.

Une perspective à terme serait d'étendre ces prédictions à un nouvel avion. L'idée générale serait de définir une représentation équivalente paramétrée à partir de connaissances métier. Des estimations préliminaires pourraient être obtenues sans données expérimentales dès la phase de conception ; ces estimations seraient ensuite raffinées dès les premiers essais. Une telle méthodologie pourrait aider à la conception d'un nouvel avion en alertant sur de possibles problèmes liés aux vibrations et en aidant au processus de spécification.

Pour répondre à ces problématiques, le manuscrit est organisé en trois parties, se terminant chacune par un résumé. La **Partie I** se concentre sur l'extraction et la représentation des données. Le Chapitre 1 introduit l'ambiance vibratoire, et le Chapitre 2 détaille les algorithmes implémentés pour la génération de jeux de données. Les différentes techniques de représentation des données sont introduites dans le Chapitre 3, puis appliquées dans le Chapitre 4 sur des données d'essais en vol. La **Partie II** s'attache au problème de la prédiction en tout point de vol. L'apprentissage supervisé est présenté dans le Chapitre 5, avec un accent sur les méthodes de régression. Ces méthodes sont alors appliquées dans le Chapitre 6 pour prédire les valeurs des raies cinématiques, générées par des sources isolées et localisées, et dans le Chapitre 7 pour prédire les niveaux large bande. La **Partie III** traite de l'extension de la prédiction à une structure avion complète. À cette fin, les modèles numériques (à bases physiques ou mécaniques) sont introduits dans le Chapitre 8, ainsi que la réduction de modèle basée sur des super-éléments dynamiques, et l'analyse des chemins de propagation. Ces outils sont alors appliqués dans le Chapitre 9, où les chemins de propagation des vibrations sont identifiés en combinant les outils précédemment introduits ainsi qu'un algorithme de classement. Enfin, le Chapitre 10 présente la théorie des graphes et le traitement du signal sur graphes, avant d'appliquer

une méthode permettant d'apprendre la structure sous-jacente de données provenant d'un réseau de capteurs uniquement à partir de mesures.

Cette thèse a pour but de rechercher des solutions innovantes répondant aux problématiques précédemment mentionnées, et ce par le biais d'une collaboration industrielle avec Dassault Aviation. Du fait de la nature des données traitées, ces travaux sont centrés sur la méthodologie et toutes les figures sont anonymisées, bien qu'un maximum d'information soit conservé.

A.2 . Résumé de la partie I

La première partie traite de l'ensemble des transformations appliquées aux données expérimentales générées aux essais en vol. En permettant la génération de différents jeux de données adaptés à l'analyse et la modélisation de l'ambiance vibratoire, il constitue la base de la **méthodologie orientée données** décrite dans les Parties II et III.

Une introduction à l'**ambiance vibratoire** est tout d'abord proposée dans le Chapitre 1. Il décrit ce qu'est l'ambiance vibratoire, ses sources, et la nécessité de son étude dans un contexte industriel aéronautique. Plusieurs représentations fréquentielles standards des signaux sont présentées, dont certaines utilisées dans la méthodologie développée dans ce manuscrit.

Puis, une vue d'ensemble du **processus de génération des jeux de données** est présentée dans le Chapitre 2. Il comprend des techniques d'identification basées sur les connaissances métier, qui sont décrites en détail. Il permet également d'appliquer plusieurs techniques de **représentation des données**, présentées dans le Chapitre 3 au côté de techniques de *clustering* pour l'extraction de variables.

Dans le Chapitre 4, les méthodes introduites dans le chapitre précédent sont appliquées pour produire une **représentation réduite des données d'ambiance vibratoire** et permettre une meilleure compréhension des phénomènes via une exploration des données. Les techniques de représentation utilisées dans le reste du manuscrit sont ici justifiées.

A.3 . Résumé de la partie II

Sur la base des outils présentés dans la Partie I, cette partie vise à fournir une méthodologie de prédiction applicable en n'importe quel point de vol, ou plus généralement une **prédiction en tout point de l'espace des paramètres d'entrée**. Elle précède ainsi les travaux présentés dans la Partie III, ayant pour but d'étendre ces prédictions à la structure complète de l'avion.

Tout d'abord, une bibliographie de l'**apprentissage supervisée** est présentée dans le Chapitre 5, détaillant différents modèles, leur entrainement, et leur évaluation. Après avoir abordé les défis liés à l'implémentation pratique de ces méthodes, la question de l'interprétabilité des modèles est traitée. Puis, sur la base de l'hypothèse de séparation du large bande présentée dans la Section 4.1.1, les problématiques de prédiction des raies cinématiques et des spectres large bande sont abordées séparément dans les chapitres suivants.

La **prédiction des raies cinématiques** est présentée dans le Chapitre 6 en mettant l'accent sur les raies cinématiques générées par le moteur gauche, une source puissante et isolée. Un processus gaussien est d'abord utilisé pour reconstruire les valeurs RMS centrées sur la fréquence des raies

cinématiques pendant des essais au sol, avec une robustesse aux conditions environnementales d'essai. Ensuite, une méthodologie de transposition des conditions au sol vers les conditions de vol est proposée, avec la perspective de prédire les niveaux de points non instrumentés dans la structure pendant les essais en vol.

La **prédiction du large bande** est présentée dans le Chapitre 7, en commençant par la prédiction des phases de vol stabilisé. Plusieurs modèles sont comparés, et un ensemble de réseaux de neurones artificiels est sélectionné comme étant le meilleur modèle. Un accent est mis sur la bonne évaluation de la capacité de généralisation des modèles, ce via un découpage du jeu de données basé sur les connaissances métier. La problématique de l'interprétabilité est abordée à l'aide de la méthode SHAP. Ce travail a été publié dans [FNM+23a]. Une extension à toute phase de vol est ensuite proposée, et est basée sur l'utilisation d'un ensemble de réseaux de neurones récurrents, permettant de modéliser les relations temporelles. Ce travail a été présenté dans [FNM+23b].

A.4 . Résumé de la partie III

Dans cette partie, la méthodologie présentée dans les Parties I et II est étendue à la **prédiction en tout point de la structure**.

Dans le Chapitre 8, les **modèles numériques** adaptés aux basses, moyennes et hautes fréquences sont présentés. Bien qu'étant des solutions industrielles standards, elles ne sont pas adaptées à la problématique traitée de par leurs limitations. Pour étudier les chemins de propagation des vibrations, des **super-éléments dynamiques** sont utilisés pour la réduction de modèle, puis analysés en utilisant le formalisme de l'***Advanced Transfer Path Analysis***. Les résultats sont présentés dans le Chapitre 9. Un algorithme permettant l'identification du chemin de propagation principal est proposé, et offre des résultats en adéquation avec les connaissances métier.

Dans le Chapitre 10, les données expérimentales générées par les essais en vol sont de nouveaux utilisées. Les **graphes** peuvent être considérés comme une manière naturelle de représenter un champ de capteurs, et sont présentés aux côtés du **traitement du signal sur graphes**, permettant de manipuler des signaux reposant sur des graphes. Ces outils sont appliqués à la problématique de **l'apprentissage de graphe à partir de données**. Ils permettent la reconstruction de signaux à partir des connexions apprises, qui sont cohérentes avec les chemins structuraux. La qualité de reconstruction est supérieure à celle produite via les corrélations. Ces travaux ont été présentés dans [FNM+22].

A.5 . Conclusion

Cette thèse propose un ensemble de solutions permettant de répondre à la problématique de la prédiction de l'ambiance vibratoire dans un avion, et ce via une approche multidisciplinaire centrée sur les données. Différents outils provenant des domaines de l'apprentissage automatique et du traitement du signal sur graphes sont combinés avec des connaissances métier dans le but de produire des outils d'analyse et de prédiction. Les principales contributions de ce manuscrit sont détaillées ci-dessous.

Tout d'abord, une méthodologie est proposée pour transformer les données à partir de leur état d'origine, volumineux et complexe à manipuler. Elle permet de définir un cadre robuste adapté à l'étude de l'ambiance vibratoire centrée sur les données. Plusieurs représentations d'ordre réduit sont proposées pour fournir des transformations de l'information robustes et interprétables par le métier. La séparation spectrale introduite est une étape essentielle de ces transformations, et l'hypothèse sur laquelle elle repose est corroborée expérimentalement. Ce processus, détaillé sous la forme d'un ensemble d'algorithmes, est implémenté dans un prototype logiciel écrit en Python. Des outils provenant du domaine de l'apprentissage non-supervisé tels que le *clustering* se révèlent efficaces pour fournir un éclairage physique et mieux comprendre l'origine de certaines vibrations. Ils peuvent également être utilisés pour identifier les variables les plus pertinentes pour la modélisation.

La prédiction de l'ambiance vibratoire en tout point du domaine de vol / de l'espace des paramètres d'entrée est abordée en appliquant des modèles d'apprentissage automatique aux données expérimentales générées par le centre d'essais en vol. Les raies cinématiques sont prédites au cours d'essais sol, et ces prédictions s'avèrent robustes aux conditions environnementales d'essai. Une méthodologie permettant de transposer ces prédictions aux conditions de vol est proposée. Cette méthode reste exploratoire mais a des implications importantes : elle pourrait permettre la prédiction en vol de points non-instrumentés à partir d'essais sol. Les spectres large bande sont prédits dans le domaine de vol à partir de l'hypothèse de séparation spectrale. La prédiction au cours des phases de vol stabilisé est la première étape qui offre des résultats répondants au besoin métier et un certain degré de robustesse. Le modèle est évalué via un découpage aléatoire classique des données ainsi que via une extrapolation, ce qui permet d'évaluer sa capacité à généraliser. De plus, l'interprétabilité du modèle est étudiée, élément crucial dans un contexte aéronautique. La prédiction est ensuite étendue à un spectre complet en phases de vol instationnaires, qui sont également à prendre en compte pour les spécifications. Un modèle adapté à la modélisation des relations temporelles est utilisé, fournissant de bons résultats. La prédiction des différentes parties d'un spectre (raies cinématiques et large bande) et du domaine de vol (palier stabilisé / manœuvres) peut être réalisée indépendamment, ce qui permet une meilleure compréhension des phénomènes en jeu.

Enfin, le problème de la prédiction à travers la structure est également abordé. Deux approches sont étudiées. Tout d'abord, des modèles numériques sont utilisés pour étudier la propagation des vibrations. Un algorithme d'extraction du chemin de propagation principal est proposé, fournissant des résultats cohérents avec la physique. Puis, une méthode provenant du traitement du signal sur graphes est utilisée pour apprendre une structure à partir de données expérimentales. Plusieurs graphes sont appris, un par bande de fréquence. Encore une fois, les résultats sont cohérents avec la physique et fournissent de très bons résultats en reconstruction. Ces deux approches peuvent être combinées. La première se base sur un super-élément dynamique pour réduire un modèle numérique, mais cette réduction peut se faire à partir de données expérimentales. La seconde, qui consiste à apprendre un graphe à partir de données expérimentales, peut se baser sur des données générées à partir d'un modèle numérique pour apprendre les graphes. Sur ces deux types de représentations, la méthodologie d'identification des chemins de propagation peut être appliquée.

Les différentes méthodologies de prédiction présentées peuvent être considérées comme des briques élémentaires pouvant être assemblées dans un cadre plus général, exploitant les données expérimentales existantes et les modèles numériques à disposition afin de permettre la prédiction de

l'ambiance vibratoire d'un avion encore phase de conception. Un modèle pourrait alors être enrichi de manière itérative par de nouvelles données expérimentales au fur et à mesure que l'avion passe par les phases de conception successives et que les essais au sol et en vol sont réalisés.

Ce travail étant avant tout exploratoire, plusieurs pistes de développement sont suggérées. L'utilisation de l'apprentissage automatique dans l'industrie aéronautique va certainement être soumise à un certain nombre de normes. Dans ce contexte, il est important d'explorer la problématique de l'interprétabilité des modèles, leurs prédictions devant être explicables et justifiées. La méthode SHAP est utilisée ici, mais d'autres méthodes existent et doivent être étudiées. De la même façon, la robustesse doit être garantie et ce a minima en indiquant quand un modèle ne peut pas être considéré comme fiable, par exemple quand les prédictions d'un modèle ensembliste divergent. En utilisant le rasoir d'Occam comme heuristique, s'appuyer sur des modèles plus simples dans certains contextes pourrait être envisagé, à condition que l'ingénierie des caractéristiques (*feature engineering*) soit réalisée de manière approfondie. En outre, la quantification des incertitudes est cruciale pour assurer la confiance au modèle.

L'étude des chemins de propagation des vibrations doit être approfondie, en utilisant notamment des algorithmes provenant de la théorie des graphes et en se basant d'avantage sur la physique. Elle devrait être appliquée également sur des données générées par d'autres modèles, pour différents systèmes, ainsi qu'à des données expérimentales.

Étant donné que les résultats dépendent de la sous-structuration du système (l'ensemble des nœuds choisis), les solutions mises en œuvre pour résoudre ce problème dans le contexte de l'analyse statistique énergétique pourraient être adaptées à ce problème. Le domaine du traitement du signal sur graphe devrait être étudié plus en détail : d'autres méthodes d'apprentissage et d'interpolation sur un graphe doivent être testées. L'examen des signaux dans l'espace des graphes pourrait également apporter une nouvelle façon d'interpréter les données.

D'autres approches numériques adaptées aux moyennes fréquences pourraient être explorées. Bien que cette thèse se soit concentrée sur l'apprentissage automatique, des approches telles que la théorie variationnelle des rayons complexes mériteraient d'être étudiées.

Il est essentiel d'étudier comment les données expérimentales et les méthodes, provenant par exemple du domaine du traitement du signal sur graphe, peuvent être intégrées dans les modèles numériques utilisés dans la phase de conception des avions. D'une part, le modèle numérique contient une description spatialement fine de la structure, mais sa prédiction est susceptible de contenir des inexactitudes, tandis que d'autre part, la représentation par graphe fournit une prédiction correcte sur une discrétisation plus grossière de la structure, correspondant aux points instrumentés.

La question de comment combiner les informations provenant de ces deux sources est cruciale. L'hybridation pourrait également être utilisée par le biais de l'extrapolation dans le domaine des fréquences : les modèles qui ne peuvent prédire que les basses fréquences pourraient être combinés avec des données expérimentales qui fournissent des résultats sur toute la gamme de fréquences.

Comme mentionné dans le premier chapitre, l'environnement vibratoire tient compte des chocs (au contraire de l'ambiance vibratoire), qui ne sont pas étudiés ici. Il serait utile de les étudier, notamment en utilisant certains des éléments méthodologiques développés ici.

En poursuivant les études et le développement de ces méthodes, il serait possible de mettre en

œuvre une approche globale axée sur les données, combinant des sources de connaissances théoriques, numériques et expérimentales. Cette approche pourrait être utilisée pour prédire l'ambiance vibratoire d'un avion complet dès sa conception et ainsi réaliser d'importantes économies en termes de coûts et de délais.

List of Figures

List of Tables

List of Algorithms

Bibliography

[AB14] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, January 2014.

[ABH+14] Onur Atak, Bart Bergen, Daan Huybrechs, Bert Pluymers, and Wim Desmet. Coupling of Boundary Element and Wave Based Methods for the efficient solution of complex multiple scattering problems. *Journal of Computational Physics*, 258:165–184, February 2014.

[ACMP08] Kazuo Aoki, Guido Cavallaro, Carlo Marchioro, and Mario Pulvirenti. On the motion of a body in thermal equilibrium immersed in a perfect gas. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(2):263–275, March 2008.

[AF21] Renzo Arina and Andrea Ferrero. Data-Driven Aeroacoustic Modelling: Trailing-Edge Noise. *AIAA AVIATION 2021 FORUM*, July 2021.

[AFW06] Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica*, 15:1–155, May 2006.

[AIR79] AIR 2004/E Résistance des avions, March 1979.

[AJ18] David Alvarez-Melis and Tommi S. Jaakkola. On the Robustness of Interpretability Methods, June 2018.

[APA+19] Àngels Aragonès, Jordi Poblet-Puig, Kevin Arcas, Pere Vicens Rodríguez, Francesc Xavier Magrans, and Antonio Rodríguez-Ferran. Experimental and numerical study of Advanced Transfer Path Analysis applied to a box prototype. *Mechanical Systems and Signal Processing*, 114:448–466, January 2019.

[ASY+19] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th {ACM} {SIGKDD} International Conference on Knowledge Discovery and Data Mining*, 2019.

[AV07] David Arthur and Sergei Vassilvitskii. K-means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms.*, pages 1027–1035, Philadelphia, PA, USA, 2007.

[AV12] Assyr Abdulle and Gilles Vilmart. A priori error estimates for finite element methods with numerical quadrature for nonmonotone nonlinear elliptic problems. *Numerische Mathematik*, 121(3):397–431, July 2012.

[Bar48] M. S. Bartlett. Smoothing Periodograms from Time-Series with Continuous Spectra. *Nature*, 161(4096):686–687, May 1948.

[Bar50] M. S. Bartlett. Periodogram Analysis and Continuous Spectra. *Biometrika*, 37(1-2):1–16, 1950.

[Bar11] Marc Barthélemy. Spatial networks. *Physics Reports*, 499(1-3):1–101, February 2011.

[BB94] P. K. Banerjee and P. K. Banerjee. *The Boundary Element Methods in Engineering*. McGraw-Hill, London ; New York, 2nd ed edition, 1994.

[BB07] Stefania Bellavia and Stefano Berrone. Globalization strategies for Newton–Krylov methods for stabilized FEM discretization of Navier–Stokes equations. *Journal of Computational Physics*, 226(2):2317–2340, October 2007.

[Bel58] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.

[Béz12] Guillaume Bézier. *Sur l'étude fréquentielle de la propagation des chocs pyrotechniques dans les structures complexes*. PhD thesis, École normale supérieure de Cachan - ENS Cachan, May 2012.

[Bio32] Maurice A. Biot. *Transient Oscillations in Elastic Systems*. PhD thesis, California Institute of Technology, 1932.

[BIPS95] Ivo Babuška, Frank Ihlenburg, Ellen T. Paik, and Stefan A. Sauter. A Generalized Finite Element Method for solving the Helmholtz equation in two dimensions with minimal pollution. *Computer Methods in Applied Mechanics and Engineering*, 128(3):325–359, December 1995.

[BLRK14] Andrea Barbarulo, Pierre Ladevèze, Hervé Riou, and Louis Kovalevsky. Proper Generalized Decomposition applied to linear acoustic: A new tool for broad band calculation. *Journal of Sound and Vibration*, 333(11):2422–2431, May 2014.

[Bon99] Marc Bonnet. Boundary Integral Equation Methods for Solids and Fluids. *Meccanica*, 34(4):301–302, October 1999.

[BP20] S. K. Bisoyi and B. K. Pal. Prediction of Ground Vibration Using Various Regression Analysis. *Journal of Mining Science*, 56(3):378–387, May 2020.

[BR07] Michael R. Blanton and Sam Roweis. K-Corrections and Filter Transformations in the Ultraviolet, Optical, and Near-Infrared. *The Astronomical Journal*, 133(2):734, January 2007.

[Bre01] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001.

[BSF94] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994.

[BYAV13] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized Denoising Auto-Encoders as Generative Models. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[Cat16] Alessandro Cattabiani. *Simulation of Low- and Mid-Frequency Response of Shocks with a Frequency Approach*. PhD thesis, Université Paris Saclay (COmUE), March 2016.

[CBRL15] Alessandro Cattabiani, Andrea Barbarulo, Hervé Riou, and Pierre Ladevèze. Variational theory of complex rays applied to shell structures: In-plane inertia, quasi-symmetric ray distribution, and orthotropic materials. *Computational Mechanics*, 56(6):983–997, December 2015.

[Cet19] Raphael Cettour-Janet. *Modelling the Vibrational Response and Acoustic Radiation of the Railway Tracks*. PhD thesis, Université Paris Saclay (COmUE), September 2019.

[CFB+02] Stephen Corda, Russell J Franz, James N Blanton, M Jake Vachon, and James B DeBoer. In-Flight Vibration Environment of the NASA F-15B Flight Test Fixture. NASA/TM-2002-210719:31, 2002.

[CG16] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, August 2016.

[Che05] Mathilde Chevreuil. *Sur une nouvelle approche en calcul dynamique transitoire, incluant les basses et les moyennes fréquences*. PhD thesis, École normale supérieure de Cachan - ENS Cachan, December 2005.

[Chi96] David Maxwell Chickering. Learning Bayesian Networks is NP-Complete. In Doug Fisher and Hans-J. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics V*, Lecture Notes in Statistics, pages 121–130. Springer, New York, NY, 1996.

[CHM04] David Maxwell Chickering, David Heckerman, and Christopher Meek. Large-Sample Learning of Bayesian Networks is NP-Hard. *Journal of Machine Learning Research*, 5(Oct):1287–1330, 2004.

[CL13] Alice Cicirello and Robin S. Langley. The vibro-acoustic analysis of built-up systems using a hybrid method with parametric and non-parametric uncertainties. *Journal of Sound and Vibration*, 332(9):2165–2178, April 2013.

[CLK05] V. Cotoni, R.S. Langley, and M.R.F. Kidner. Numerical and experimental validation of variance prediction in the statistical energy analysis of built-up systems. *Journal of Sound and Vibration*, 288(3):701–728, December 2005.

[CLR07] Mathilde Chevreuil, Pierre Ladevèze, and Philippe Rouch. Transient analysis including the low- and the medium-frequency ranges of engineering structures. *Computers & Structures*, 85(17):1431–1444, September 2007.

[CML+19] A. Clot, J.W.R. Meggitt, R.S. Langley, A.S. Elliott, and A.T. Moorhouse. Development of a hybrid FE-SEA-experimental model. *Journal of Sound and Vibration*, 452:112–131, July 2019.

[CP09] Andrzej Cichocki and Anh-Huy Phan. Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations. *IEICE Transactions*, 92-A:708–721, March 2009.

[CS207] CS-25 Certification Specifications for Large Aeroplanes, September 2007.

[CvG+14] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, September 2014.

[CVSK15] Siheng Chen, Rohan Varma, Aliaksei Sandryhaila, and Jelena Kovačević. Discrete Signal Processing on Graphs: Sampling Theory. *IEEE Transactions on Signal Processing*, 63(24):6510–6523, December 2015.

[DA16] Mehdi Dehghan and Mostafa Abbaszadeh. Proper orthogonal decomposition variational multiscale element free Galerkin (POD-VMEFG) meshless method for solving incompressible Navier–Stokes equation. *Computer Methods in Applied Mechanics and Engineering*, 311:856–888, November 2016.

[DBB99] Arnaud Deraemaeker, Ivo Babuska, and Philippe Bouillard. Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions. *International Journal for Numerical Methods in Engineering*, 46(4):471–499, October 1999.

[DGD+20] Leo Dostal, Helge Grossert, Daniel A. Duecker, Malte Grube, Daniel Christopher Kreuter, Kai Sandmann, Benjamin Zillmann, and Robert Seifried. Predictability of Vibration Loads From Experimental Data by Means of Reduced Vehicle Models and Machine Learning. *IEEE Access*, 8:177180–177194, 2020.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. July 2011.

[DHTY12] Huo-Yuan Duan, Po-Wen Hsieh, Roger C. E. Tan, and Suh-Yuh Yang. Analysis of a new stabilized finite element method for the reaction–convection–diffusion equations with a large reaction coefficient. *Computer Methods in Applied Mechanics and Engineering*, 247–248:15–36, November 2012.

[Dij59] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[DMBB20] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-Objective Counterfactual Explanations. In Thomas Bäck, Mike Preuss, André Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann, editors, *Parallel Problem Solving from Nature – PPSN XVI*, Lecture Notes in Computer Science, pages 448–469, Cham, 2020. Springer International Publishing.

[DO110] DO-160G Environmental Conditions and test Procedures for Airborne Equipment, December 2010.

[DOP12] Wim Desmet, Atak Onur, and Bert Pluymers. CAE Methodologies for Mid-Frequency analysis in Vibration and Acoustics. *Leuven University Press*, page 31, 2012.

[Doz16] Timothy Dozat. Incorporating Nesterov Momentum into Adam. February 2016.

[DPR15] Cristina Díaz-Cereceda, Jordi Poblet-Puig, and Antonio Rodríguez-Ferran. Automatic subsystem identification in statistical energy analysis. *Mechanical Systems and Signal Processing*, 54–55:182–194, March 2015.

[DS22] Shuyu Dong and Michèle Sebag. From graphs to DAGs: A low-complexity model and a scalable algorithm, April 2022.

[DTFV16] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. Learning Laplacian Matrix in Smooth Graph Signal Representations. *arXiv:1406.7842 [cs, stat]*, February 2016.

[DTRF19] Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard. Learning graphs from data: A signal representation perspective. page 38, 2019.

[DvSV02] W. Desmet, B. van Hal, P. Sas, and D. Vandepitte. A computationally efficient prediction technique for the steady-state dynamic analysis of coupled vibro-acoustic systems. *Advances in Engineering Software*, 33(7):527–540, July 2002.

[Eic09] Michael Eichler. Granger-causality graphs for multivariate time series. September 2009.

[EKX96] Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, OR, 1996. AAAI.

[EWHD16] AbdElRahman ElSaid, Brandon Wild, James Higgins, and Travis Desell. Using LSTM recurrent neural networks to predict excess vibration events in aircraft engines. In *2016 IEEE 12th International Conference on E-Science (e-Science)*, pages 260–269, Baltimore, MD, USA, October 2016. IEEE.

[EY36] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, September 1936.

[FD07] Brendan J. Frey and Delbert Dueck. Clustering by Passing Messages Between Data Points. *Science*, 315(5814):972–976, February 2007.

[FDDC89] Leopoldo Penna Franca and Eduardo Gomes Dutra Do Carmo. The Galerkin gradient least-squares method. *Computer Methods in Applied Mechanics and Engineering*, 74(1):41–54, September 1989.

[FFML97] Leopoldo P. Franca, Charbel Farhat, Antonini P. Macedo, and Michel Lesoinne. Residual-free bubbles for the Helmholtz equation. *International Journal for Numerical Methods in Engineering*, 40(21):4003–4009, 1997.

[FHL20] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep Ensembles: A Loss Landscape Perspective. *arXiv:1912.02757 [cs, stat]*, June 2020.

[FI11] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the beta-divergence, March 2011.

[FNM+22] Stéphane Février, Stéphane Nachar, Lionel Mathelin, Frédéric Giordano, and Bérengère Podvin. Apprentissage de Graphe pour la Reconstruction de l'Environnement Vibratoire. In *15ème Colloque National En Calcul Des Structures*, 83400 Hyères-les-Palmiers, France, May 2022. Université Polytechnique Hauts-de-France [UPHF].

[FNM+23a] Stéphane Février, Stéphane Nachar, Lionel Mathelin, Frédéric Giordano, and Bérengère Podvin. Data-Driven Model to Predict Aircraft Vibration Environment. *AIAA Journal*, 61(10):4610–4622, July 2023.

[FNM+23b] Stéphane Février, Stéphane Nachar, Lionel Mathelin, Frédéric Giordano, and Bérengère Podvin. Data-Driven Prediction of Aircraft Vibration Environment During Unsteady Flight Dynamics. In *1st Annual Aerospace Structures, Structural Dynamics, and Materials Conference (SSDM)*, San Diego (California), United States, June 2023. ASME.

[For56] L. R. Ford. Network Flow Theory. Technical report, RAND Corporation, January 1956.

[FR91] Charbel Farhat and Francois-Xavier Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6):1205–1227, 1991.

[Fri01] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, October 2001.

[GAM87] GAM-EG-13B Essais de Compatibilité à l'Environnement Climatique, Mécanique, Électrique, Électromagnétique et Spécial des Matériels Aéronautiques, June 1987.

[GB02] P. Gardonio and M. J. Brennan. On the Origins and Development of Mobility and Impedance Methods in Structural Dynamics. *Journal of Sound Vibration*, 249:557–573, January 2002.

[GG89] I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):773–780, July 1989.

[GG15] Yuan Gao and Dorota Glowacka. Deep Gate Recurrent Neural Network. 2015.

[GGJA13] Oriol Guasch, Carlos García, Jordi Jové, and Pere Artís. Experimental validation of the direct transmissibility approach to classical transfer path analysis on a mechanical setup. *Mechanical Systems and Signal Processing*, 37(1-2):353–369, May 2013.

[GKBP15] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, January 2015.

[GM77] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, December 1977.

[GM04] O. Guasch and F.X. Magrans. The Global Transfer Direct Transfer method applied to a finite simply supported elastic beam. *Journal of Sound and Vibration*, 276(1-2):335–359, September 2004.

[GS00] F.A. Gers and J. Schmidhuber. Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 3, pages 189–194 vol.3, July 2000.

[Gua09] Oriol Guasch. Direct transfer functions and path blocking in a discrete mechanical system. *Journal of Sound and Vibration*, 321(3):854–874, April 2009.

[GVR+21] Paul Garnier, Jonathan Viquerat, Jean Rabault, Aurélien Larcher, Alexander Kuhnle, and Elie Hachem. A review on deep reinforcement learning for fluid mechanics. *Computers & Fluids*, 225:104973, July 2021.

[HBO+21] Pierre Humbert, Batiste Le Bars, Laurent Oudre, Argyris Kalogeratos, and Nicolas Vayatis. Learning Laplacian Matrix from Graph Signals with Sparse Spectral Representation. *Journal of Machine Learning Research*, 22(195):1–47, 2021.

[HGW+16] Weiyu Huang, Leah Goldsberry, Nicholas F. Wymbs, Scott T. Grafton, Danielle S. Bassett, and Alejandro Ribeiro. Graph Frequency Analysis of Brain Signals. *IEEE Journal of Selected Topics in Signal Processing*, 10(7):1189–1203, October 2016.

[HH92] Isaac Harari and Thomas J. R. Hughes. Galerkin/least-squares finite element methods for the reduced wave equation with non-reflecting boundary conditions in unbounded domains. *Computer Methods in Applied Mechanics and Engineering*, 98(3):411–454, August 1992.

[HKM+01] Harry Himelblau, Dennis L. Kern, Jerome E. Manning, Allan G. Piersol, and Sheldon Rubin. *NASA Handbook 7005: Dynamics Environmental Criteria*, volume NASA-HDBK-7005. March 2001.

[HNR68] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[Hoc91] Josef Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. April 1991.

[Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Basis Expansions and Regularization. In Trevor Hastie, Robert Tibshirani, and Jerome Friedman, editors, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, pages 139–189. Springer, New York, NY, 2009.

[Hug95] Thomas J. R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, 127(1):387–401, November 1995.

[HYL18] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation Learning on Graphs: Methods and Applications. *arXiv:1709.05584 [cs]*, April 2018.

[HZC21] Xin He, Kaiyong Zhao, and Xiaowen Chu. AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems*, 212:106622, January 2021.

[IOCDP03] Sergio R. Idelsohn, Eugenio Oñate, Nestor Calvo, and Facundo Del Pin. The meshless finite element method: THE MESHLESS FINITE ELEMENT METHOD. *International Journal for Numerical Methods in Engineering*, 58(6):893–912, October 2003.

[Jar30] Vojtěch Jarník. O jistém problému minimálním. (Z dopisu panu O. Borůvkovi). *Práce moravské přírodovědecké společnosti 6*, 4:57–63, 1930.

[JGG+11] Karl Janssens, Peter Gajdatsy, Ludo Gielen, Peter Mas, Laurent Britte, Wim Desmet, and Herman Van Der Auweraer. OPAX: A new transfer path analysis method based on parametric load models. *Mechanical Systems and Signal Processing*, 25(4):1321–1338, May 2011.

[JMG+08] K. Janssens, P. Mas, Péter Gajdátsy, L. Gielen, and H. Auweraer. A novel path contribution analysis method for test- based NVH troubleshooting. 2008.

[KB17] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017.

[KDTB01] M. Kamiński, M. Ding, W. A. Truccolo, and S. L. Bressler. Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of significance. *Biological Cybernetics*, 85(2):145–157, August 2001.

[KK95] Eisuke Kita and Norio Kamiya. Trefftz method: An overview. *Advances in Engineering Software*, 24(1):3–12, January 1995.

[KKK16] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! Criticism for Interpretability. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[KLR12] Louis Kovalevsky, Pierre Ladevèze, and Hervé Riou. The Fourier version of the Variational Theory of Complex Rays for medium-frequency acoustics. *Computer Methods in Applied Mechanics and Engineering*, 225–228:142–153, June 2012.

[KLRB12] Louis Kovalevsky, Pierre Ladevèze, Hervé Riou, and Marc Bonnet. The Variational Theory of Complex Rays for Three-Dimensional Helmholtz Problems. *Journal of Computational Acoustics*, 20(04):1250021, December 2012.

[Koh82] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, January 1982.

[Kov11] Louis Kovalevsky. *La théorie variationnelle des rayons complexes version Fourier : application aux problèmes tridimensionnels de vibro-acoustique*. PhD thesis, École normale supérieure de Cachan - ENS Cachan, June 2011.

[KRL13] L. Kovalevsky, H. Riou, and P. Ladevèze. On the use of the Variational Theory of Complex Rays for the analysis of 2-D exterior Helmholtz problem in an unbounded domain. *Wave Motion*, 50(3):428–436, April 2013.

[KRL14] Louis Kovalevsky, Hervé Riou, and Pierre Ladevèze. A Trefftz approach for medium-frequency vibrations of orthotropic structures. *Computers & Structures*, 143:85–90, September 2014.

[Kru56] Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

[KT81] Daniel Kahneman and Amos Tversky. The Simulation Heuristic. Technical report, 1981.

[KT11] Adam Kapela and Nikolaos Michael Tsoukias. Multiscale FEM modeling of vascular tone: From membrane currents to vessel mechanics. *IEEE transactions on bio-medical engineering*, 58(12):3456–3459, December 2011.

[KW22] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, December 2022.

[Lad96] Pierre Ladevèze. A new computational approach for structure vibrations in the medium frequency range. *C. R. Acad. Sci. Paris, Série II*, 12(322):849–856, 1996.

[Lal09a] Christian Lalanne. *Mechanical Vibrations and Shock Analysis - Fatigue Damage*, volume 4. 2009.

[Lal09b] Christian Lalanne. *Mechanical Vibrations and Shock Analysis - Mechanical Shock*, volume 2. 2009.

[Lal09c] Christian Lalanne. *Mechanical Vibrations and Shock Analysis - Random Vibration*, volume 3. 2009.

[Lal09d] Christian Lalanne. *Mechanical Vibrations and Shock Analysis - Specification Development*, volume 5. 2009.

[LARB01] Pierre Ladevèze, Lionel Arnaud, Philippe Rouch, and Claude Blanzé. The variational theory of complex rays for the calculation of medium-frequency vibrations. *Engineering Computations*, 18(1-2):193–214, 2001.

[LB04] R.S. Langley and A.W.M. Brown. The ensemble statistics of the energy of a random system subjected to harmonic excitation. *Journal of Sound and Vibration*, 275(3-5):823–846, August 2004.

[LBCM10] Alain Le Bot, Antonio Carcaterra, and Denis Mazuyer. Statistical Vibroacoustics and Entropy Concept. *Entropy*, 12(12):2418–2435, December 2010.

[LBHOK19] Batiste Le Bars, Pierre Humbert, Laurent Oudre, and Argyris Kalogeratos. Learning Laplacian Matrix from Bandlimited Graph Signals. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2937–2941, Brighton, United Kingdom, May 2019. IEEE.

[LC09] Robin S. Langley and Julio A. Cordioli. Hybrid deterministic-statistical analysis of vibro-acoustic systems with domain couplings on statistical components. *Journal of Sound and Vibration*, 321(3):893–912, April 2009.

[LDD95] Richard H. Lyon, Richard G. DeJong, and Richard G. DeJong. *Theory and Application of Statistical Energy Analysis*. Butterworth-Heinemann, Boston, 2nd ed edition, 1995.

[LEL19] Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. Consistent Individualized Feature Attribution for Tree Ensembles, March 2019.

[LeN19] Alexander LeNail. NN-SVG: Publication-Ready Neural Network Architecture Schematics. *Journal of Open Source Software*, 4(33):747, January 2019.

[LG21] Qin Lu and Georgios B. Giannakis. Probabilistic Reconstruction of Spatio-Temporal Processes Over Multi-Relational Graphs. *IEEE Transactions on Signal and Information Processing over Networks*, 7:166–176, 2021.

[Lip90] Peter Lipton. Contrastive Explanation. *Royal Institute of Philosophy Supplements*, 27:247–266, March 1990.

[LL17] Scott Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4768–4777, November 2017.

[Llo82] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.

[LM62] Richard H. Lyon and Gideon Maidanik. Power Flow between Linearly Coupled Oscillators. *The Journal of the Acoustical Society of America*, 34(5):623–639, May 1962.

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[LPC+93] Larry D. Libersky, Albert G. Petschek, Theodore C. Carney, Jim R. Hipp, and Firooz A. Allahdadi. High Strain Lagrangian Hydrodynamics. *Journal of Computational Physics*, 109(1):67–75, November 1993.

[LRRB03] P. Ladevèze, P. Rouch, H. Riou, and X. Bohineust. Analysis of Medium-Frequency Vibrations in a Frequency Range. *Journal of Computational Acoustics*, 11(02):255–283, June 2003.

[LS99] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[LSH+23] Jiashuo Liu, Zheyan Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards Out-Of-Distribution Generalization: A Survey, July 2023.

[Lüt13] Helmut Lütkepohl. Vector autoregressive models. *Chapters*, pages 139–164, 2013.

[LZV+22] Jessica Leoni, Francesco Zinnari, Eugenia Villa, Mara Tanelli, and Andrea Baldi. Flight regimes recognition in actual operating conditions: A functional data analysis approach. *Engineering Applications of Artificial Intelligence*, 114:105016, September 2022.

[Mac71] Richard H. MacNeal. A hybrid method of component mode synthesis. *Computers & Structures*, 1(4):581–601, December 1971.

[Mac03] B. Mace. Statistical energy analysis, energy distribution models and system modes. *Journal of Sound and Vibration*, 264(2):391–409, July 2003.

[Mag81] F.X. Magrans. Method of measuring transmission paths. *Journal of Sound and Vibration*, 74(3):321–330, February 1981.

[Man18] Vincent Manet. Méthode des Éléments Finis: vulgarisation des aspects mathématiques et illustrationde la méthode. page 429, December 2018.

[MARG21] Francesc X. Magrans, Àngels Aragonès, Antonio Rodríguez-Ferran, and Oriol Guasch. The solution of vibroacoustic linear systems as a finite sum of transmission paths. *Mechanical Systems and Signal Processing*, 153:107464, May 2021.

[Maz05] Vincent Mazet. *Développement de Méthodes de Traitement de Signaux Spectroscopiques : Estimation de La Ligne de Base et Du Spectre de Raies*. These de doctorat, Nancy 1, January 2005.

[Maz22] Vincent Mazet. Background correction, MATLAB Central File Exchange, 2022.

[MB96] J. M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1):289–314, December 1996.

[MDB99] Nicolas Moës, John Dolbow, and Ted Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.

[MG05] F. X. Magrans and O. Guasch. The Role of the Direct Transfer Matrix as Connectivity Matrix and Application to the Helmholtz Equation in 2D: Relation to Numerical Methods and Free Field Radiation Example. *Journal of Computational Acoustics*, 13(02):341–363, June 2005.

[MIL08] MIL-STD-810G Environmental Engineering Considerations and Laboratory Tests, October 2008.

[Mil19] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, February 2019.

[Mol22] Christoph Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. 2 edition, 2022.

[MPR18] Francesc Xavier Magrans, Jordi Poblet-Puig, and Antonio Rodríguez-Ferran. A subsystem identification method based on the path concept with coupling strength estimation. *Mechanical Systems and Signal Processing*, 100:588–604, February 2018.

[MPU21] Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. Uncertainty in Gradient Boosting via Ensembles. April 2021.

[MSC05] Francesc Magrans, Pere Vicens Sorribes, and G.C. Cousin. Low and mid-high frequency Advanced Transmission Path Analysis. *12th International Congress on Sound and Vibration 2005, ICSV 2005*, 4:3292–3299, January 2005.

[NB18] David Najera and Adam Brink. Efficient Random Vibration Analysis of Nonlinear Systems with Long Short-Term Memory Networks for Uncertainty Quantification. *ISMA 2018 International Conference on Noise and Vibration Engineering and USD2018 International Conference on Uncertainty in Structural Dynamics*, SAND2018-9830C, 2018.

[NBBC19] Hoang Nguyen, Xuan-Nam Bui, Hoang-Bac Bui, and Dao Trong Cuong. Developing an XGBoost model to predict blast-induced peak particle velocity in an open-pit mine: A case study. *Acta Geophysica*, 67(2):477–490, April 2019.

[NGO13] Sunil K Narang, Akshay Gadde, and Antonio Ortega. Signal processing techniques for interpolation in graph structured data. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5445–5449, Vancouver, BC, Canada, May 2013. IEEE.

[Nic98] Raymond S. Nickerson. Confirmation Bias: A Ubiquitous Phenomenon in Many Guises. *Review of General Psychology*, 2(2):175–220, June 1998.

[NXN14] NX Nastran Superelement User's Guide. 2014.

[NY06] Kousuke Noumura and Junji Yoshida. Method of Transfer Path Analysis for Vehicle Interior Sound with no Excitation Experiment. In *FISITA 2006 World Automotive Congress*, Yokohama, Japan, October 2006. JSAE.

[OFK+18] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst. Graph Signal Processing: Overview, Challenges and Applications. *arXiv:1712.00468 [eess]*, March 2018.

[OS98] R. Ohayon and Christian Soize. *Structural Acoustics and Vibration: Mechanical Models, Variational Formulations and Discretization*. Academic Press, San Diego, 1998.

[PDCFN18] G. Paneiro, F. O. Durão, M. Costa e Silva, and P. Falcão Neves. Artificial neural network model for ground vibration amplitudes prediction due to light railway traffic in urban areas. *Neural Computing and Applications*, 29(11):1045–1057, June 2018.

[Pea01] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, November 1901.

[PMB13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training Recurrent Neural Networks, February 2013.

[PTB03] E. Perrey-Debain, J. Trevelyan, and P. Bettess. Plane wave interpolation in direct collocation boundary element method for radiation and wave scattering: Numerical aspects and applications. *Journal of Sound and Vibration*, 261(5):839–858, April 2003.

[PTB04] E. Perrey-Debain, J. Trevelyan, and P. Bettess. Wave boundary elements: A theoretical overview presenting applications in scattering of short waves. *Engineering Analysis with Boundary Elements*, 28(2):131–141, February 2004.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[RB18] Marko Robnik-Šikonja and Marko Bohanec. Perturbation-Based Explanations of Prediction Models. In Jianlong Zhou and Fang Chen, editors, *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*, Human–Computer Interaction Series, pages 159–175. Springer International Publishing, Cham, 2018.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA, USA, January 1986.

[Rit09] Walter Ritz. über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik. *Journal für die Reine und Angewandte Mathematik*, 135:1–61, 1909.

[RKJ+19] Jean Rabault, Miroslav Kuchta, Atle Jensen, Ulysse Réglade, and Nicolas Cerardi. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of fluid mechanics*, 865:281–302, 2019.

[RLDV14] Edwin Reynders, Robin S. Langley, Arne Dijckmans, and Gerrit Vermeir. A hybrid finite element – statistical energy analysis approach to robust sound transmission modeling. *Journal of Sound and Vibration*, 333(19):4621–4636, September 2014.

[RLK13] Hervé Riou, Pierre Ladevèze, and Louis Kovalevsky. The Variational Theory of Complex Rays: An answer to the resolution of mid-frequency 3D engineering problems. *Journal of Sound and Vibration*, 332(8):1947–1960, April 2013.

[RLR04] H. Riou, P. Ladevèze, and P. Rouch. Extension of the variational theory of complex rays to shells for medium-frequency vibrations. *Journal of Sound and Vibration*, 272(1):341–360, April 2004.

[RMG17] Daniel Romero, Meng Ma, and Georgios B. Giannakis. Kernel-Based Reconstruction of Graph Signals. *IEEE Transactions on Signal Processing*, 65(3):764–778, February 2017.

[RMU86] David E. Rumelhart, James L. McClelland, and San Diego PDP Research Group University of California. *Parallel Distributed Processing : Explorations in the Microstructure of Cognition*. Cambridge, Mass. : MIT Press, 1986.

[RPZ+18] Bin Ren, Laurent Pueyo, Guangtun Ben Zhu, John Debes, and Gaspard Duchêne. Non-negative Matrix Factorization: Robust Extraction of Extended Structures. *The Astrophysical Journal*, 852(2):104, January 2018.

[RSG16a] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-Agnostic Interpretability of Machine Learning, June 2016.

[RSG16b] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier, August 2016.

[RSG18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-Precision Model-Agnostic Explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018.

[RW06] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass, 2006.

[SAJEH11] D. Sarsri, L. Azrar, A. Jebbouri, and A. El Hami. Component mode synthesis and polynomial chaos expansions for stochastic frequency functions of large linear FE models. *Computers & Structures*, 89(3):346–356, February 2011.

[SBL19] Stefania Sardellitti, Sergio Barbarossa, and Paolo Di Lorenzo. Graph Topology Inference Based on Sparsifying Transform Learning. *IEEE Transactions on Signal Processing*, 67(7):1712–1727, April 2019.

[SCC+22] Ishrat Singh, Philip Conrad, Puja Chowdhury, Jason D. Bakos, and Austin Downey. Real-Time Forecasting of Vibrations with Non-stationarities. In Ramin Madarshahian and Francois Hemez, editors, *Data Science in Engineering, Volume 9*, Conference Proceedings of the

Society for Experimental Mechanics Series, pages 21–29, Cham, 2022. Springer International Publishing.

[SDS19] Ljubiša Stanković, Miloš Daković, and Ervin Sejdić. Introduction to Graph Signal Processing. In Ljubiša Stanković and Ervin Sejdić, editors, *Vertex-Frequency Analysis of Graph Signals*, pages 3–108. Springer International Publishing, Cham, 2019.

[SGS93] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*, volume 81 of *Lecture Notes in Statistics*. Springer, New York, NY, 1993.

[SH97] James R Stewart and Thomas J. R Hughes. H-Adaptive finite element computation of time-harmonic exterior acoustics problems in two dimensions. *Computer Methods in Applied Mechanics and Engineering*, 146(1):65–89, July 1997.

[Sha51] Lloyd S. Shapley. Notes on the N-Person Game — I: Characteristic-Point Solutions of the Four-Person Game. Technical report, RAND Corporation, July 1951.

[Shi55] Alfonso Shimbel. Structure in communication nets. In *Proceedings of the Symposium on Information Networks*, pages 199–203. New York, Polytechnic Press of the Polytechnic Institute of Brooklyn, 1955.

[SHJ+20] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods, February 2020.

[Sho59] The shortest path through a maze. In *Proc. Internat. Sympos. Switching Theory 1957*, pages 285–292, Cambridge, 1959. Massachusetts: Harvard Univ. Press.

[Sin04] T.N. Singh. Artificial neural network approach for prediction and control of ground vibrations in mines. *Mining Technology*, 113(4):251–256, December 2004.

[SJA04] Z. Q Sheng, S Jirathearanat, and T Altan. Adaptive FEM simulation for prediction of variable blank holder force in conical cup drawing. *International Journal of Machine Tools and Manufacture*, 44(5):487–494, April 2004.

[ŠK14] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3):647–665, December 2014.

[SL05] P. J. Shorter and R. S. Langley. Vibro-acoustic analysis of complex systems. *Journal of Sound and Vibration*, 288(3):669–699, December 2005.

[SM00] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[Smi62] P. W. Smith. Response and Radiation of Structural Modes Excited by Sound. *The Journal of the Acoustical Society of America*, 34(5):640–647, May 1962.

[SMLR16] Santiago Segarra, Antonio G. Marques, Geert Leus, and Alejandro Ribeiro. Reconstruction of Graph Signals Through Percolation from Seeding Nodes. *IEEE Transactions on Signal Processing*, 64(16):4363–4378, August 2016.

[SNF+13] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. *IEEE Signal Processing Magazine*, 30(3):83–98, May 2013.

[Soi93] Christian Soize. A model and numerical method in the medium frequency range for vibroacoustic predictions using the theory of structural fuzzy. *The Journal of the Acoustical Society of America*, 94(2):849–865, August 1993.

[Soi98] Christian Soize. Reduced models in the medium frequency range for general dissipative structural-dynamics systems. *European Journal of Mechanics - A/Solids*, 17(4):657–685, 1998.

[SPP+05] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science (New York, N.Y.)*, 308(5721):523–529, April 2005.

[SSE+17] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Transactions on Database Systems*, 42(3):19:1–19:21, July 2017.

[Ste83] Stanley R Sternberg. Biomedical Image Processing. *Computer*, 16(1):22–34, January 1983.

[SWL+19] Xiupeng Shi, Yiik Diew Wong, Michael Zhi-Feng Li, Chandrasekar Palanisamy, and Chen Chai. A feature learning approach based on XGBoost for driving assessment and risk prediction. *Accident Analysis & Prevention*, 129:170–179, August 2019.

[TEOC20] Yuichi Tanaka, Yonina C. Eldar, Antonio Ortega, and Gene Cheung. Sampling Signals on Graphs: From Theory to Applications. *IEEE Signal Processing Magazine*, 37(6):14–30, November 2020.

[TG06] N. Totaro and J.L. Guyader. SEA substructuring using cluster analysis: The MIR index. *Journal of Sound and Vibration*, 290(1-2):264–289, February 2006.

[TGRB20] Andrew S. Tenney, Mark N. Glauser, Christopher J. Ruscher, and Zachary P. Berger. Application of Artificial Neural Networks to Stochastic Estimation and Jet Noise Modeling. *AIAA Journal*, 58(2):647–658, February 2020.

[Tho82] D.J. Thomson. Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*, 70(9):1055–1096, September 1982.

[Tof15] Chris Tofallis. A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society*, 66(8):1352–1362, August 2015.

[Tre26] Erich Trefftz. Ein Gegenstück zum Ritzschen Verfahren. In *Verhandl. 2er Internat. Kongress. Techn. Mechanik Zürich*, pages 131–137, 1926.

[Tur52] Alan Mathison Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, August 1952.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[vdR16] Maarten V. van der Seijs, Dennis de Klerk, and Daniel J. Rixen. General framework for transfer path analysis: History, theory and classification of techniques. *Mechanical Systems and Signal Processing*, 68–69:217–244, February 2016.

[vDV05] B. van Hal, W. Desmet, and D. Vandepitte. Hybrid finite element—wave-based method for steady-state interior structural–acoustic problems. *Computers & Structures*, 83(2-3):167–180, January 2005.

[VGO+20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, SciPy 1.0 Contributors, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Häggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V. Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert-Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, José Vinícius de Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, and Yoshiki Vázquez-Baeza. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020.

[Vib12] Vibrations, failure of the right main landing gear torsion link during landing roll. Accident report g-ld110723a, BEA, Bureau d'Enquêtes et d'Analyses pour la sécurité de l'aviation civile, March 2012.

[VMLH22] J. Viquerat, P. Meliga, A. Larcher, and E. Hachem. A review on deep reinforcement learning for fluid mechanics: An update. *Physics of Fluids*, 34(11):111301, November 2022.

[von73] Chr. von der Malsburg. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14(2):85–100, June 1973.

[von07] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[Web19] Arthur Gordon Webster. Acoustical Impedance, and the Theory of Horns and of the Phonograph. *Proceedings of the National Academy of Sciences of the United States of America*, 5(7):275–282, 1919.

[Wel67] P. Welch. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, June 1967.

[WIG+22] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, July 2022.

[WMR18] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR, March 2018.

[WOdS20] Laura Wilmes, Raymond Olympio, Kristin M. de Payrebrune, and Markus Schatz. Structural Vibration Tests: Use of Artificial Neural Networks for Live Prediction of Structural Stress. *Applied Sciences*, 10(23):8542, November 2020.

[Woo72] A.G. Woods. Design for acceptable aircraft vibration. *Journal of Sound and Vibration*, 20(3):353–358, February 1972.

[XZL+21] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon S. Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks. *International Conference on Learning Representations*, March 2021.

[ZARX18] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning, November 2018.

[ZGB+13] Bin Zhang, Chris Gaiteri, Liviu-Gabriel Bodea, Zhi Wang, Joshua McElwee, Alexei A. Podtelezhnikov, Chunsheng Zhang, Tao Xie, Linh Tran, Radu Dobrin, Eugene Fluder, Bruce Clurman, Stacey Melquist, Manikandan Narayanan, Christine Suver, Hardik Shah, Milind Mahajan, Tammy Gillis, Jayalakshmi Mysore, Marcy E. MacDonald, John R. Lamb,

David A. Bennett, Cliona Molony, David J. Stone, Vilmundur Gudnason, Amanda J. Myers, Eric E. Schadt, Harald Neumann, Jun Zhu, and Valur Emilsson. Integrated systems approach identifies genetic nodes and networks in late-onset Alzheimer's disease. *Cell*, 153(3):707–720, April 2013.

[ziv06] Vector Autoregressive Models for Multivariate Time Series. In Eric Zivot and Jiahui Wang, editors, *Modeling Financial Time Series with S-PLUS®*, pages 385–429. Springer, New York, NY, 2006.

[ZLSL01] Xiong Zhang, Xiao-Hu Liu, Kang-Zu Song, and Ming-Wan Lu. Least-squares collocation meshless method. *International Journal for Numerical Methods in Engineering*, 51(9):1089–1100, July 2001.

[ZTZ10] Olgierd C. Zienkiewicz, Robert L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier, Amsterdam Heidelberg, 6. ed., reprint., transferred to digital print edition, 2010.