Amélioration de la conception de la dynamique structurelle et de la vibroacoustique grâce à des jumeaux numériques utilisant l'apprentissage automatique et l'apprentissage par transfert

Barbara Zaparoli Cunha

Numéro d'ordre NNT: 2023ECDL0045 Année: 2023

THESE de DOCTORAT DE L'UNIVERSITÉ DE LYON
OPÉRÉE AU SEIN DE L'ÉCOLE CENTRALE DE LYON

ÉCOLE DOCTORALE MEGA

(Mécanique, Énergétique, Génie Civil, Acoustique)

Spécialité: MÉCANIQUE

Soutenue le 06/12/2023 par

Barbara Zaparoli Cunha

Enhancing Structural Dynamics and Vibroacoustics Design through Digital Twins Enabled by Machine Learning and Transfer Learning

---

Devant le jury composé de:

Rapporteur:	Prof. Sergio DE ROSA	-	Professeur, UNINA, DII
Rapporteur:	Dr. Scott COGAN	-	Chercheur, UFC, FEMTO-ST
Présidente:	Prof. Marie-Annick GALLAND	-	Professeur, ECL, LMFA
Directeur:	Prof. Mohamed ICHCHOU	-	Professeur, LTDS, ECL
Co-directeur:	Prof. Abdelmalek ZINE	-	Professeur, ICJ, ECL
Co-encadrant:	Dr. Christophe Droz	-	Chercheur, INRIA
Co-encadrant:	Dr. Stéphane Foulard	-	CEO, COMPREDICT GmbH

Abstract

Vibroacoustic design is critical in the early stages of product development to detect and resolve potential issues before the product is put into production, ultimately reducing development time and costs and improving the final product performance. A promising tool that has emerged for vibroacoustic design is the Digital Twin (DT). DTs create high-fidelity virtual replicas of products at a low computational cost, enabling a thorough design search that leads to optimal and robust designs. However, the nonlinear and non-smooth nature of vibroacoustic problems, the lack of interpretability and physics foundation of data-based models, and the data scarcity at the early stages of design pose challenges for building accurate and affordable DTs.

In this context, this thesis proposes a novel data-based framework for modeling DTs of vibroacoustic systems that can speed up system evaluations and improve product design. The proposed framework relies on Machine Learning (ML) techniques to infer system models based on data obtained from high-fidelity simulations. The framework combines various learning methodologies to improve the accuracy, interpretability, and physical consistency of ML algorithms for usage in the vibroacoustic domain. Additionally, since acquiring a substantial amount of data in the early stages of vibroacoustic design is usually prohibitively expensive, this thesis aims to improve the data efficiency of these algorithms to make their implementation feasible.

The thesis evaluates different classes of ML methods in vibroacoustic problems with varying degrees of nonlinearity, non-smoothness, uncertainty, and dimensionality to identify suitable methods. Moreover, a feature engineering approach incorporates expert knowledge into the ML model by creating new inputs based on existing physical models to address ML's accuracy and consistency with the target physical system. The thesis also employs Global Sensitivity Analysis (GSA) to enhance the model's interpretability and physical consistency. The sensitivity analysis provides a comprehensive understanding of how changes in the input variables affect the model's output, facilitating the identification of the most influential variables.

To overcome the scarcity of data at the beginning of vibroacoustic design, a novel solution is introduced: the use of Transfer Learning (TL) to leverage knowledge from past product generations. The effectiveness of TL in problems with low-dimensional tabular data under conditional shift was thoroughly examined to assess the approach's viability in constructing DTs. This analysis identifies TL methodologies suitable for constructing vibroacoustic DTs and demonstrates that the proposed approach significantly enhances data efficiency.

Furthermore, the thesis demonstrates the proposed framework's use in an industrial application, specifically the early robust optimization of the NVH performance of gear transmissions. This demonstration showcases how the framework contributes to modeling DTs of vibroacoustic systems. The developed DT accelerates system

evaluations and aids in product design. Thus, this thesis provides valuable tools to overcome the challenges of constructing DTs for vibroacoustic product design.

Keywords: Digital Twin, Vibroacoustic, Surrogate Models, Transfer Learning, Product Design, Machine Learning, Digital Prototype, Data Efficiency.

Acknowledgments

Here, I embark on the challenging task of expressing, in just a few words, the deep gratitude and appreciation I have for everyone who supported me throughout my thesis journey. It was your guidance and affection that made this milestone possible.

First, I would like to express my gratitude for the financial support provided by the Marie Skłodowska-Curie Actions to the LIVE-I project. I am honored to have benefited from an initiative that promotes research, scientific dissemination, and sustainable development.

I extend my heartfelt thanks to my supervisors, Dr. Foulard, Dr. Droz, Prof. Ichchou, and Prof. Zine, for their invaluable guidance and for fostering both my professional and personal growth. Whether you played the role of the good cop or the bad cop, I learned a great deal from each of you.

I am also profoundly grateful to all the colleagues I've met along the way, many of whom have become cherished friends. Special thanks to my friends at Ecole Centrale de Lyon for bringing joy to E6 and for joining me daily in the essential, coffee-powered recharges—especially Thomas, Xeufeng, Firas, Daniel, Ali, Giulia, Xavier, Leo, and Quentin. I am equally thankful to my peers from the LIVE-I project for the mutual support and the learning we shared. To the entire team at COMPREDICT, who welcomed me, opened their doors, and from whom I continue to learn each day, my deepest gratitude goes out to you.

My appreciation also extends to my friends in the Brazilian community, both in France and Germany, and to those who supported me from afar in Brazil. Thanks to you, I never felt alone on this journey. A special mention to Allan, who deserves pages of gratitude for helping me through the most difficult moments of my thesis—I will always be thankful to you.

Finally, I thank my family, the Zaparolis and Cunhas, for providing me with the education and support needed to achieve this dream and, most importantly, for their unwavering love and encouragement despite the distance and longing. Lele e Papi, muito obrigada, eu amo vocês. Mãe, você sempre será minha maior fonte de força e exemplo de luta. Seu amor me move. Essa conquista é para vocês.

Contents

List of Acronyms

Introduction

Motivation

Structural Dynamics and Vibroacoustics (SD&V) is the study of the dynamic behavior of structures under mechanical vibrations and their interaction with acoustic phenomena. The SD&V design focuses on optimizing mechanical systems to minimize vibrations and noise while maintaining structural integrity. The design process tries to achieve a well-balanced compromise among these objectives and additional factors such as product weight, cost, and environmental impact. This comprehensive approach is essential for developing high-performing and competitive products in multiple industries. However, the ever-shortening design cycles and the necessity to meet strict regulations [1, 2] have made the design of vibroacoustic products increasingly challenging. Consequently, it has become imperative to address vibroacoustic aspects early in development to avoid costly redesigns, saving valuable time and resources.

In the late 20th century, the introduction of virtual prototyping led by Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) revolutionized the product development process [3]. Virtual prototypes leverage computational simulations to provide an efficient and accurate means of early vibroacoustic design, as it is less resource-intensive to experiment and conduct tests on virtual models than on physical machines. Over the past four decades, virtual prototypes have been widely adopted in vibroacoustic design [4]. However, as requirements continue to increase and more aspects and parameters must be considered, the cost of extensive domain exploration using vibroacoustic simulations becomes prohibitive. This is particularly true because vibroacoustic simulations tend to be costly since they are often broadband, multi-scale, and have high-resolution (space or frequency) requirements. Furthermore, CAE models fail to incorporate the vast amount of real-world data collected from existing products in today's digital age, hindering their ability to capture the complexities of practical scenarios. Therefore, new approaches are needed to overcome these limitations and further enhance the effectiveness of vibroacoustic design methodologies.

A promising solution gaining momentum in virtual prototyping is the concept of DT. DTs combine physical-based modeling and data-based techniques to create virtual counterparts of physical products, enabling cost-effective and accurate analysis that can incorporate real measurements and adapt throughout the product's life cycle. While many researchers have emphasized the significance of DTs in the current digitalization of the industry [5–10], their potential for vibroacoustics and their practical implementation still requires further exploration. Additionally, most existing DTs in SD&V primarily focus on applications in the operational phase of the product, such as Structural Health Monitoring (SHM) and Active Control, with

limited research dedicated to their utilization in the product design phase [11].

Less expensive copies of CAE simulations, known as surrogate models, play a crucial role in implementing DTs for the design process [12]. Previous studies in SD&V surrogates relied mainly on Response Surface Model (RSM) [13–15], which approximates the true model using polynomial fitting and fails to capture the arbitrary nonlinearities in the system. In contrast, ML models make fewer assumptions about the function class and stand out for learning complex data relationships and nonlinearities, making them suitable for constructing vibroacoustic DTs. While some recent studies have applied ML for this purpose, many challenges remain. The non-smooth nature of SD&V problems poses difficulties in constructing accurate ML-based surrogates, as ML methods typically assume the local smoothness of data [16, 17]. Furthermore, ML methods are often seen as black-box approaches, lacking interpretability and physical constraints, which raise skepticism in the industry and in the scientific community regarding their application in applied sciences.

Another challenge with ML-based approaches in vibroacoustics is their data dependency, requiring a large amount of data for accurate learning of the system's behavior. This poses a significant problem, particularly in the early stages of product development when data is scarce and expensive to acquire. While Active Sampling (AS) techniques have been explored to select the most informative samples [18–20], leveraging existing data and physical knowledge to increase data efficiency remains unexplored, presenting an opportunity to explore TL and Physics-Guided Machine Learning (PGML) research fields.

The objective of this thesis is to introduce a novel ML-based framework for modeling DTs of vibroacoustic products readily applicable in the early stages of product development. The framework will apply innovative solutions using ML-based surrogate models and TL to overcome the aforementioned challenges. Furthermore, the approach should remain versatile and be applicable to general SD&V problems. By investigating the implementation of DTs in the vibroacoustic design process, this research aims to establish novel approaches for efficient and effective product development, where virtual prototypes serve as valuable tools for designers and engineers to ensure optimal performance, enhanced user experience and ultimately achieve cost and time savings.

Research Questions and Contributions

Given the above discussion on the thesis goal and challenges, the key research questions to be answered by this thesis are:

1. How can data-based DTs' accuracy, interpretability, and physical consistency be enhanced for their application in vibroacoustic design?
2. What strategies can be employed to overcome data scarcity in early-stage vibroacoustic design?

To address these research questions, the proposed DT solutions rely on ML techniques to infer system models using data obtained from high-fidelity simulations

and on TL to leverage already available data and improve data efficiency. The contributions of this thesis can be summarized as follows:

1. Comprehensive review of the applications of ML models in SD&V, highlighting the strengths and drawbacks of ML approaches and discussing current trends and perspectives in the field. The review presented in Chapter 1 is a valuable guide for structural and vibroacoustic engineers interested in exploring ML techniques.
2. Identification of ML algorithms suitable to tackle non-smooth vibroacoustic problems in Chapter 2, resulting in more accurate and reliable predictions. The suitability and robustness of the algorithms are investigated through a study with the Sound Transmission Loss (STL) problem with varying degrees of complexity, design spaces, and materials.
3. Analysis of the use of Physics-Guided Features (PGF) to readily incorporate expert knowledge into ML models. The findings from Chapter 2 show that adding PGF can improve the accuracy and physical consistency of the ML model.
4. Implementation of GSA techniques to improve the DT interpretability in Chapter 3. While classical variance-based GSA and ML-based GSA were successfully applied, the ML-based approach is more flexible and accurate than classical variance-based GSA for scarce data conditions.
5. The thesis addresses the issue of data scarcity in early-stage vibroacoustic design by introducing the exploitation of already available data from previous generations of the product using TL. In Chapter 4, a viability study demonstrates the ability of TL to enhance data efficiency in problems with low-dimensional tabular data by transferring information learned from correlated domains. It was shown that TL enables deeper neural network models under data scarcity and can significantly reduce prediction errors.
6. The real-world effectiveness of the proposed framework is demonstrated in Chapter 5 for an industrial application: the early robust optimization of the weight and NVH performance of gear transmission. The DT framework is applied to perform a multi-objective robust optimization of the gear micro- and macro-geometries in a fraction of the time from traditional methods. Furthermore, the use of deep TL enables the DT to adapt across different gear transmission generations.

In summary, this thesis addresses the research questions by exploring different ML and TL techniques to propose a framework that is an effective tool for implementing DTs in the early stages of vibroacoustic product development. The thesis' contributions encompass advancements in ML techniques, handling non-smooth vibroacoustic problems, incorporating expert knowledge, enhancing DT interpretability, overcoming data scarcity through TL, and demonstrating the practical application of the framework in an industrial context. The framework's versatility allows for easy application to different problems, facilitating its implementation in industrial contexts. Through these contributions, the thesis aims to advance the field of vibroacoustic design and provide valuable insights for industrial practitioners and

researchers alike.

Thesis outline

This thesis is structured into six chapters, each addressing different aspects of implementing the proposed DT framework and its practical applications in industrial contexts.

Chapter 1 presents the concept of DT, its construction based on the merge of physics- and ML-based models, and its applications and perspectives in SD&V. The chapter also introduces the fundamentals of ML, specifically focusing on the most commonly used ML algorithms for surrogate modeling in SD&V, highlighting their strengths and drawbacks. The chapter also discusses future trends as TL and PGML. Overall, Chapter 1 provides an overview of the research field of DT in SD&V design and identifies its research gaps and prospects.

In Chapter 2, suitable ML approaches for handling challenges in vibroacoustic problems are investigated through benchmarking. The concept of PGFs is introduced, and its effect on DT accuracy is studied. The canonical vibroacoustic problem used for these studies is the STL, which is modeled with varying levels of complexity and non-smoothness, as well as different materials, to evaluate the performance of ML approaches under different scenarios.

Chapter 3 focuses on describing variance-based and Mean Decrease in Impurity (MDI) methods used for GSA to improve the interpretability of the surrogate models. The application of GSA methods is showcased using the STL problem, and a comparison is made between the variance-based and MDI approaches.

Chapter 4 investigates the viability of applying TL with low-dimensional tabular data under conditional shift. The analyses are performed on function samples from the Matern kernel, representing a wide range of functions with tunable non-smooth parameters. The chapter compares the performance of different TL approaches, examines how TL improvement changes with the distance between source and target domains, and explores the impact of function smoothness. It also aims to identify a distance metric between source and target domains that correlates with the improvement achieved through TL.

Chapter 5 focuses on implementing the inter-generational DT framework for an industrial application: the Noise, Harshness, and Vibration (NVH) of gear transmission. The chapter introduces the industrial case, explaining the acoustic mechanism and its source, the transmission error. It describes the high-fidelity models used and presents a workflow for applying the proposed framework to the design process. The DT framework is implemented for the gear transmission problem, which is applied to the robust multi-objective optimization of the gear micro- and macro-geometry parameters. Finally, the DT evolves to account for the modifications of the gear transmission next generation through deep TL.

The final chapter comprehensively discusses the research findings, contributions, and implications. It summarizes the key insights gained from each chapter and

presents the conclusions drawn from the thesis. This chapter also highlights potential avenues for future research in vibroacoustic design.

In conclusion, this thesis progresses through chapters that cover the state-of-the-art, investigate suitable ML approaches, explore GSA, analyze TL, and implement the DT framework for an industrial application. This comprehensive structure provides a cohesive and detailed exploration of implementing a DT approach for vibroacoustic systems, offering valuable insights and practical applications in industrial settings.

Mathematical Notation

Scalars are represented by non-bold variables, e.g., x Lowercase, bold variables, e.g., $\boldsymbol{x}$, denote vectors, and uppercase, bold variables, e.g., $\boldsymbol{X}$, represent matrices in a Euclidean space. Sets are represented by uppercase, calligraphic variables, e.g., $\mathcal{X}$. The notation $P(\cdot)$ denotes the probability of a specific event in a given space.

For a given problem, $\mathcal{X}$ represents the input space, $\mathcal{Y}$ represents the output space, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ is the problem domain, and $f : \mathcal{X} \to \mathcal{Y}$ is the target function. A dataset with N samples sampled from a distribution $\mathcal{D}$ over $\mathcal{Z}$ is denoted by $S = \{\boldsymbol{z}_i\}_{i=1}^{N} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, where $\boldsymbol{z}_i \in \mathcal{Z}, \boldsymbol{x}_i \in \mathcal{X}$, and $\boldsymbol{y}_i \in \mathcal{Y}$. The symbol X represents an input instance set, i.e., $X = \{\boldsymbol{x_i}, \ldots, \boldsymbol{x_N}\} \in \mathcal{X}$, and Y represents the corresponding output instance set, i.e., $Y = f(X) = \{\boldsymbol{y_i}, \ldots, \boldsymbol{y_N}\} \in \mathcal{Y}$. The problem domain $\mathcal{D}$ consists of its feature space $\mathcal{X}$ and marginal probability distribution $P(X)$. The problem task $\mathcal{T}$ consists of its output space $\mathcal{Y}$ and the target function $f(\cdot)$.

Chapter 1

Digital Twins and Machine Learning in Structural Dynamics and Vibroacoustic

This chapter presents a comprehensive overview of ML and DT in the context of SD&V prototyping. It covers the fundamental concepts of DT and ML, providing a thorough understanding of the topics necessary for comprehending the thesis. The chapter discusses the current state of ML-based surrogates in vibroacoustics, identifying research gaps and potential ML techniques for the ongoing development of the DT framework. The chapter emphasizes the role of DTs in enabling design space exploration, uncertainty quantification, and optimization in SD&V. Lastly, emerging trends and perspectives in the field, including Physics-Guided Machine Learning and Transfer Learning, are presented. Overall, this chapter establishes the foundation for the subsequent methodologies employed in this study.

Part of this chapter was published in the authors' comprehensive review paper in reference [11] from the journal Mechanical Systems and Signal Processing.

Contents

Introduction

Digital twins (DTs) are virtual representations of physical systems, integrating measured data with advanced modeling and simulation techniques. By accurately replicating the behavior and characteristics of their physical counterparts, DTs offer a powerful platform for monitoring, analyzing, and optimizing the performance of complex systems. The concept of DTs has gained prominence with Industry 4.0 and the era of digitalization, which has resulted in an unprecedented generation and storage of information.

However, such a large amount of data requires processing and translation abilities beyond human capacity. Thus, ML algorithms have become essential to enable the automatic processing of massive datasets, extracting patterns, making inferences, and delivering predictions based on available information. In this context, ML plays a crucial role as an enabler of DTs, augmenting their capabilities and enhancing their potential: digitalization and connectivity collect raw system data, while ML translates this data into meaningful insights.

This chapter examines the relationship between ML and DTs in the context of leveraging digitalization and big data for product design in SD&V analysis. This overview aids in understanding the current state-of-the-art and identifies appropriate approaches and research gaps in the field. Additionally, this chapter provides a comprehensive introduction to ML fundamentals and offers guidance on selecting suitable algorithms while considering crucial factors such as the application objective, database size, computational resources, problem dimensionality, data nature, uncertainty management, and system nonlinearity.

This chapter also provides an overview of numerous ML applications in SD&V, highlighting their various benefits. However, it also acknowledges the presence of drawbacks, misuses, and challenges of the ML-based methods, indicating the potential for further advancement in the field. For example, the lack of interpretability and physical basis raises great apprehension about using ML in SD&V and other physical sciences. Furthermore, although the behavior of mechanical waves in SD&V systems encloses frequency information, which is well explored in applications such as SHM, it also leads to non-monotonic and rough function behaviors, posing challenges for ML models. The current implementation of ML in the industry is also hindered by the need for substantial amounts of labeled data required in deep learning and the cost of ML simulations. Additionally, there is an ongoing debate about when the usage of ML is justifiable and provides gains in terms of time and precision with an adequate confidence level. This chapter discusses these issues alongside relevant references and approaches that have attempted to address them.

Finally, this chapter examines the current trends in ML applications in the context of SD&V analysis, with specific emphasis on two main approaches: PGML and TL. PGML refers to the integration of physical principles and domain knowl-

edge into ML models, addressing the concern regarding the lack of interpretability and physical basis in ML applications. On the other hand, TL methods leverage knowledge acquired from one task or domain and apply it to another related task or domain with limited labeled data. TL can potentially enhance the efficiency and effectiveness of ML models in SD&V, particularly when labeled data is scarce or costly to obtain.

Section 1.1 defines DTs and outlines their current implementation in SD&V, along with a discussion of the perspectives in the field. Section 1.2 offers guidelines for ML usage, highlighting the strengths and weaknesses of various ML algorithms in SD&V product design. Section 1.2 also introduces the ML algorithms used in this thesis. Section 1.3 presents the workflow of DT modeling in SD&V prototyping and reviews its utilization for uncertainty quantification and design optimization. Finally, Section 1.4 reviews and discusses emerging trends and perspectives, with specific emphasis on two main approaches: PGML and TL. In summary, this chapter provides a comprehensive background and future opportunities in the joint research field of ML and SD&V design, laying the foundation for the methodologies applied throughout this thesis.

1.1 Digital Twins: Definition and Applications in Structural Dynamics and Vibroacoustics

1.1.1 Definition of Digital Twins

DT is a time-evolving high-fidelity replica of a product/process with a bidirectional information connection. The concept was first introduced in 2003 in a presentation on Product Lifecycle Management (PLM) by Michael Grieves [5], but it only spread with the famous article by Glaessgen and Stargel [9]. Grieves and Vickers [5] envisioned DT as an integrated multiphysics and multiscale simulation of the real system using the best physical models and data available to create a virtual copy of the system. By doing so, DT is intended to be able to continuously forecast the system's health and create plans to mitigate the damage or improve performance while accounting for the system's associated uncertainties.

The concept of DT is still evolving as DT enablers are under ongoing development, and DT applications are spreading to many sectors, such as buildings [21], smart cities [22], manufacturing [23], healthcare [24]. Several articles have focused on reviewing the characteristics and achievements of the DT [10, 25–37].

According to the existing literature, comprehensive DTs should:

- store and manage product data;
- incorporate data-driven and high-fidelity simulations;
- establish a bidirectional live data connection;
- integrate multiphysics and multiscale simulation;
- evolve throughout the product lifecycle;
- associate other DTs within the same fleet;

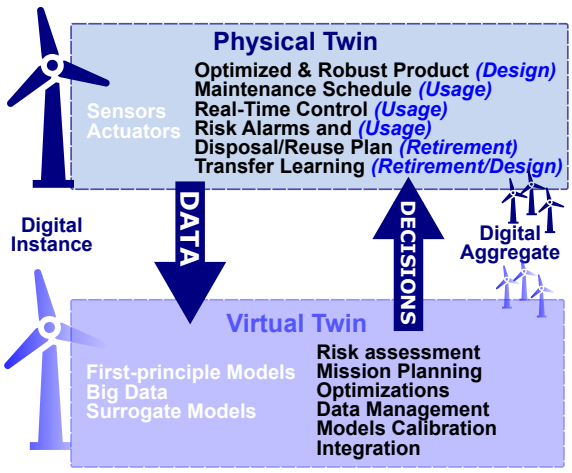


Figure 1.1: Digital twin framework: the data from the physical twin is processed by physics and data-driven methods by the DT, which supports optimized and robust decisions throughout the product lifecycle.

- make inferences from data;
- assimilate and adapt to environmental and operational information;
- handle physical variability, data uncertainty, and model errors;
- provide real-time predictions;
- optimize operational parameters;
- enhance decision-making processes;
- reduce potential risks.

Figure 1.1 serves as a visual representation of a comprehensive DT, showcasing its components, behavior, and uses across the product lifecycle. However, to the best of the authors' knowledge, a DT with all the above functionalities does not currently exist, and its full implementation may require several decades of further development, as predicted by Glaessgen and Stargel [9]. Nevertheless, recent implementations have resulted in notable "incomplete" DT that meet some of these criteria, as discussed throughout this chapter.

According to Gardner et al. [38], the DT is built from components from four main categories: *simulations*, which model the physics of the system; the *knowledge* from experts and previous experiences about the product and the environment variables; the available *data* of the physical twin; and the *connectivity* which links the other elements and gives DT the ability to adapt with information. These components and their interconnections are the building blocks for creating a DT, as illustrated in Figure 1.2.

Figure 1.2 illustrates that ML can be one component of a DT, although it is not a mandatory requirement. An example is the DT implemented by Aivaliotis et al. [39], which incorporates physics-based modeling, virtual sensor modeling, and automatic calibration of model parameters to provide Remaining Useful Life predictions without utilizing ML algorithms. Although ML usage is not imperative

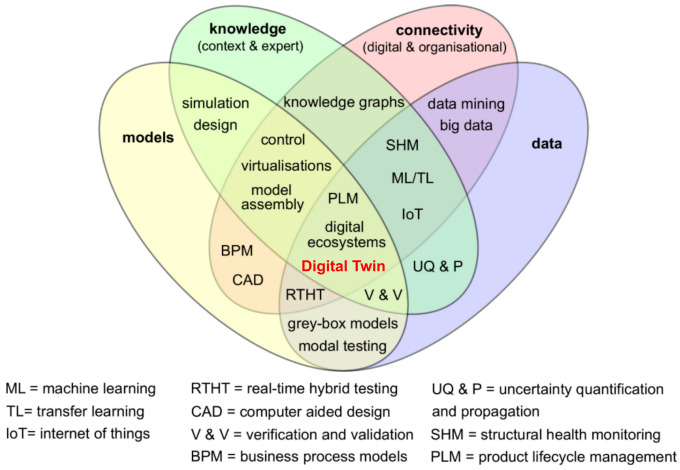


Figure 1.2: Main components and interconnections of a Digital Twin framework as building blocks. Adapted from: Gardner et al. [38].

for constructing a DT, their capabilities in extracting valuable insights from large datasets, handling uncertainties, and making fast predictions with good generalization are extremely advantageous. Consequently, ML is considered a core enabler of DTs [33]. The subsequent discussions will primarily focus on ML-enabled DTs.

1.1.2 Classification and Applications of Digital Twins

DTs can be classified based on the stage of product development [5, 10], as depicted in Figure 1.3. During the conceptualization or design stage, the DT is called a DT Prototype. DT Prototypes can rely on CAD, physical, and data-based models to create a virtual representation of the physical prototype. According to Grieves and Vickers [5], the DT Prototype should be able to verify and validate the system requirements as predicted and mitigate unforeseen undesirable responses. Thus, the DT Prototype should enable cost-effective predictions of the system responses, facilitating the extensive exploration of the design space, system optimizations, and uncertainty analyses. Therefore, surrogate models for DT Prototyping are one particularly advantageous type of data-based model, as they accelerate computationally expensive simulations based on simulated data and can also incorporate measured data.

Furthermore, the DT Prototype should evolve and be applicable throughout various stages of the product lifecycle [5, 36, 10, 32, 30]. For example, leveraging the Digital Prototype's ability to readily predict the system response, it can be effectively employed in real-time model-based control and SHM. Nevertheless, according to Jones et al. [10] and as depicted in Figure 1.4, the design phase receives relatively less attention in DT research. This thesis helps to address this research gap, as it aims to develop a DT specifically for SD&V product design. Section 1.3 presents a detailed discussion on this topic.

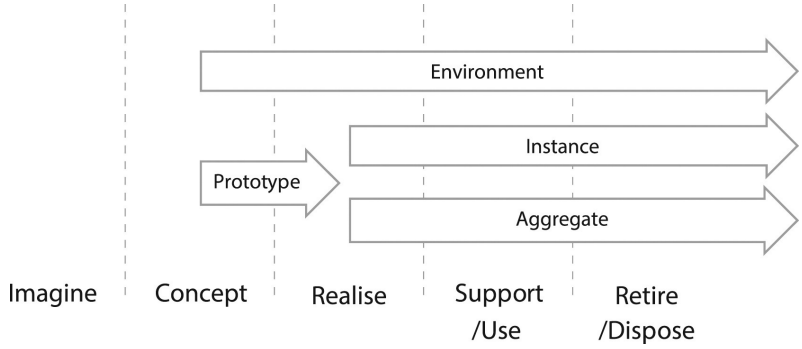


Figure 1.3: Digital Twin evolution throughout the product's life. (Source:Jones et al. [10]).

During the subsequent phases of the product lifetime, the DT is classified into DT Instance or DT Aggregate (see Figure 1.3) [5]. A DT Instance replicates a specific physical counterpart, capturing its unique characteristics and detailed behavior, such as in [38, 40, 41]. On the other hand, the DT Aggregate represents a group of related physical instances, incorporating data and insights from multiple products to approximate the overall expected behavior of the population, such as in [42, 43]. Thus, DT Instances provide detailed information about individual entities, and DT Aggregates offer a higher-level perspective for system-level predictions, decision-making, and optimization.

As depicted in Figure 1.4, many DTs implemented in the literature concentrate on the realization phase, i.e., manufacturing and production processes [23, 44, 45]. However, within the domain of SD&V, the majority of ML-based DTs are predominantly utilized in the operational stage [11], named as support/use stage in Figures 1.3 and 1.4. At this phase, the DT is employed for various purposes, such as early failure detection, control strategies, product and environment monitoring, and mission planning. There exists a vast literature and a diverse range of ML approaches applied to address these objectives in SD&V that, in general, can be categorized into SHM and Active Control of noise and vibration. For a comprehensive discussion on ML applications in the operational phase of SD&V systems, please refer to Appendix A.

An illustrative example of integrating ML and physics-based models in SHM is presented by [41]. They developed a calibrated DT using measured and simulated data to identify damage severity and location in a bar structure. The proposed bi-directional DT allows updating the physical twin's operation parameters and control strategies. The DT implemented in [40] also combines data- and physics-based models to detect Euler buckling failure and assess structural reliability. Zhang and Sun [46] utilized a Neural Network (NN) guided by Finite Element Method (FEM) results to improve generality and physical consistency in damage detection.

DT's damage detection and prediction capability has also been used for mission planning. Karve et al. [47] presents an exemplary DT applied in 'what if?' scenarios.

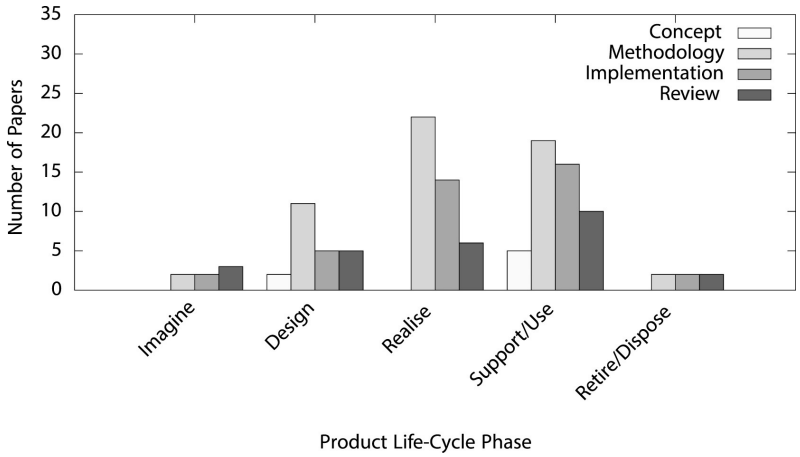


Figure 1.4: Research papers on DT distribution across the product life-cycle. (Source:Jones et al. [10]).

Their DT model incorporates time-evolving dynamics, bi-directional connections, and a fusion of data-driven and physics-based methods. Their DT facilitates mission planning strategies that minimize potential damage by accounting for both aleatory and epistemic uncertainties. Another significant contribution to mission planning is demonstrated by Kapteyn et al. [48], who implemented a DT framework for aircraft monitoring and real-time decision-making. Their approach involves an ML-based classification and a surrogate model selected from a library of physical models shared by different components. Informed by the estimated damage severity, the DT dynamically adjusts the planned maneuvers to avoid structural failure, showcasing the DT's ability to enhance decision-making processes.

A comprehensive example of DT implementation for active control is demonstrated by Gardner et al. [38]. Initially, measured data were employed to calibrate the parameters of the physical model. Subsequently, the outputs of this model were utilized as inputs for a Gaussian Process Regressor (GPR) technique, which enhanced output prediction by accounting for uncertainties and non-modeled physics through online AS. The methodology is demonstrated in the study case of a three-storey structure. Remarkably, despite the physical model's linearity, the DT could predict the nonlinear behavior arising from the contact between the column and bumper under specific excitations. Besides that, DT can make predictions steps ahead of time because it was trained with lagged information and, therefore, is appropriate for the control of the structure.

DT applications during the middle-of-life stage of a product are diverse and extend beyond the examples mentioned above. One notable application that has gained significant momentum is virtual sensing. *Virtual sensing leverages the power of DT to estimate or infer physical measurements without relying solely on direct measurements from physical sensors*, leading to more informative operations and eliminating the need for additional hardware [49, 39].

Finally, the DT should also collaborate for the product end-of-life. For instance,

the ability of certain DTs to estimate the remaining useful life of components plays a vital role in making informed decisions about end-of-life scenarios, including disposal, reuse, and market value assessment [50, 51]. However, very few studies focus on the product end-of-life, as Figure 1.4 supports.

Another valuable yet under-explored potential of DTs is their use in leveraging information gathered throughout the product lifecycle to assist the design of the product's next generation. By evaluating components that were over- or under-designed based on insights provided by the DT, future designs can be optimized accordingly. Additionally, the DT can aid in investigating the causal factors behind observed phenomena, enhancing our understanding of complex systems. This thesis will delve further into using DT to leverage inter-generational information in the product design employing TL-enabled DT Prototypes (see Section 1.4).

1.2 Overview of Machine Learning Algorithms

1.2.1 Definition and Classification of Machine Learning

An ML algorithm is an artificial intelligence algorithm that makes an inference from data and experiences without explicit programming. The classical definition by Mitchell and Mitchell [52] states that ML is a class of computer programs that: *"learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."*

Three key elements can describe an ML algorithm: *representation*, which defines the hypothesis space $\mathcal{H}$ of all possible models $h \in \mathcal{H}$ considered to represent the relations or patterns in the dataset $\mathcal{S}$, e.g., Decision Trees, neural networks, hyperplane representations; *evaluation*, which defines the cost function $C(\mathcal{S}, h)$ that assesses the model performance, e.g., accuracy, squared error, K-L divergence; and the *learning algorithm*, which is the method to identify $h \in \mathcal{H}$ that best fits the training dataset according to the evaluation criterion, e.g., gradient descent, greedy search, and Bayesian inference.

The ML algorithms can be classified according to the dataset and learning approach as supervised learning, unsupervised learning, and reinforcement learning [53]:

- **Supervised learning**: the ML algorithm learns a function $h : \mathcal{X} \to \mathcal{Y}$ that maps the input space $\mathcal{X}$ to the output space $\mathcal{Y}$ based on a training dataset that comprises a labeled set of input-output pairs $\mathcal{S} = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}\}$. The goal in supervised learning is to use the model fitted with the training data to predict the output of new unseen inputs $\boldsymbol{x}^*$, i.e., $y^* = h(\boldsymbol{x}^*)$. Supervised models can be categorized as regression models if the outputs are continuous values or as classification models if the outputs are categories or discrete values.
- **Unsupervised learning:** given a dataset only with inputs $\mathcal{S} = \{\boldsymbol{x} \in \mathcal{X}\}$, the goal is to unveil underlying patterns and hidden structures in the data. There-

fore, unsupervised models can simplify and describe unlabeled data. Popular classes of unsupervised learning are clustering, used to classify the data into groups with maximum similarity, density estimation, used to find the data distribution, and dimension reduction, used to discover lower dimensional space of latent variables that capture the data essential information.

- **Reinforcement learning:** class of ML algorithms in which an agent interacts with an environment and learns from the success and errors of these experiences. The agent performs actions A that transform the environment state S, which generates a direct reward R. The goal is to find a policy $\pi : S \rightarrow A$ that maps actions to states to maximize the expected future reward [54].

1.2.2 ML in SD&V: General Guidelines and Algorithms

As stated by the "*no free lunch*" theorem, no learning algorithm outperforms the others in any domain [55]. Usually, investigating a good ML algorithm for a given problem involves trial-end-error experiments. Nevertheless, the algorithms considered should be selected accordingly to the volume and nature of the data, the resources available, and the purpose of the task. The first step to defining appropriate ML models is identifying the learning category (supervised, unsupervised, or reinforcement) and the analysis purpose, e.g., group data, reduce data dimensionality, and perform regression. Subsequently, one may consider how the algorithm assumptions relate to the many aspects of the data, such as complexity, nonlinearity, input dimensionality, time-dependency, spatial dependency, continuous or discrete variables, independent or dependent variables, and uncertainty level. In general, simple ML models with satisfactory accuracy should be preferred over complex ones because they tend to generalize better to new data (avoid overfitting), require fewer data, and be more interpretable [56, 57]. Observing the algorithm's popular applications can give a good indication of the circumstances in which an ML algorithm excels and is suitable.

In the context of SD&V, ML algorithms have been widely applied in SHM, active noise and vibration control, and product design. Based on an extensive search of recent publications in this field using the Scopus database, the authors identified the most relevant ML algorithms are NN, Support Vector Machine (SVM), and Gaussian Process (GP)-based models. In the product design application, most ML algorithms employed are supervised regression algorithms, as the algorithms usually should predict a continuous system response. In particular, GPR, NN, and Decision Trees-based models are among the most commonly used for DT prototyping. While SVM and linear regression models are also popular, SVM is primarily suitable for classification problems, and linear regression requires predetermined nonlinear combinations of problem parameters and has stronger assumptions about the function class.

Therefore, this chapter will focus on reviewing NN, GPR, and Decision Trees-based models in Sections 1.2.2.2, 1.2.2.3, and 1.2.2.4, respectively, due to their popularity and suitability to the thesis topic. Section 1.2.2.1 introduces the bias-variance

trade-off property of ML supervised algorithms, as it is an important concept applicable to the ML algorithms under consideration. Appendix A provides an overview of other commonly used supervised, unsupervised, and reinforcement learning methods in SD&V. Additionally, the appendix presents a comprehensive literature review of ML applied in SHM and active control of noise and vibration.

1.2.2.1 Bias-Variance Trade-off

Supervised learning algorithms aim to discover a prediction model that can generalize well on unseen data rather than fitting a model to a specific training dataset. Therefore, minimizing the cost function during training is insufficient to ensure a good predictive model. The true evaluation of the algorithm's performance should be based on its predictions on unseen data from a test dataset. This evaluation considers the generalization error, which consists of three components: bias, variance, and irreducible errors [58].

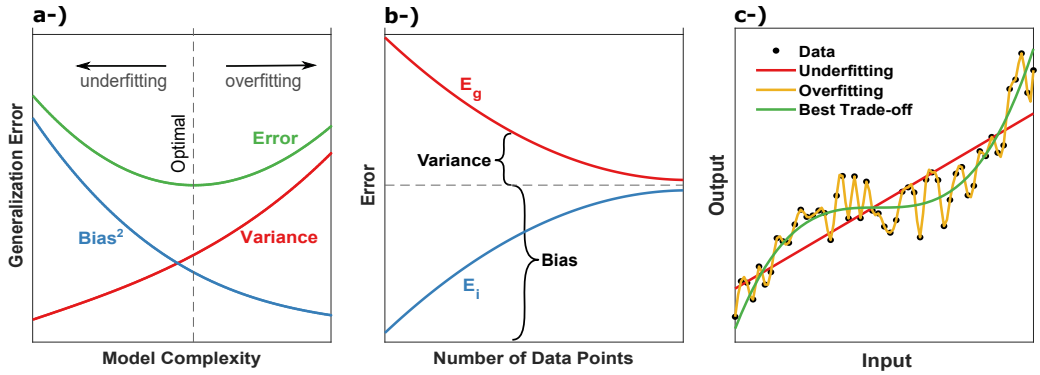


Figure 1.5: (a) Bias-variance trade-off on Machine Learning for a given number of data points. Bias, variance, and irreducible errors sum up to the generalization error. The optimal bias-variance compromise should minimize the generalization error, avoiding underfitting and overfitting; (b) Illustration of how the error in the training dataset E_i is smaller than the true generalization error E_g and how the prediction accuracy improves with more samples in the dataset for a given model, without changing model' complexity; (c) Example of models underfitting, overfitting, and with an appropriate bias-variance trade-off. Figures adapted from [58].

The bias error measures the level of incorrect hypotheses in the model and tends to decrease with model complexity. The variance error measures the variability of model predictions and typically increases with model complexity. Therefore, a high-bias model oversimplifies the problem, leading to bad predictions in the training and test datasets (underfitting). In contrast, a low-bias model performs well in the training dataset but might lead to high-variance error (overfitting). Figure 1.5-a illustrates this bias-variance trade-off for a given number of training points. Figure 1.5-b shows how complex models with low bias become viable with increasing database size.

To address overfitting, regularization techniques can be applied to penalize model complexity and increase robustness to ill-posed problems. Additionally, selecting proper hyperparameters for the ML algorithm, either through expert reasoning or search algorithms, enables a good balance between bias and variance [59–61].

While these principles apply to all supervised ML algorithms, each algorithm has its specificity, including different hypothesis spaces, evaluation metrics, and learning algorithms, leading to different advantages and drawbacks. The following sections provide more details about NN, GPR, and Decision Trees-based models.

1.2.2.2 Neural Networks

NN is a network of artificial neural units inspired by the human brain [62]. Most Deep Learning (DL) models are based on NN with multiple layers, which enables high-level feature extraction from raw data. NN-based algorithms are also the most used ML algorithms in the main applications of SD&V, namely SHM, active control, and product design. The NNs' popularity is due to their suitability to approximate a function without strong assumptions on its format, their flexibility, and their easy implementation supported by popular libraries.

The vanilla NN architecture is the Multilayer Perceptron (MLP) [63], in which the neurons of one layer are fully connected to the neurons in the next layer [64]. The connection between each neural unit from layer i to each neural unit in the subsequent layer j is parameterized by the weight $w_{i,j}$. Additionally, each layer is parameterized by a bias term b. Each neural unit in the MLP computes the weighted sum of its inputs given by $z_j = \sum_i (w_{ji} \cdot o_i) + b_j$, as illustrated in Figure 1.6-a. This weighted sum then undergoes an activation function denoted as $\varphi(\cdot)$, such as the sigmoid function, which introduces nonlinearity into the network. The outputs from one layer serve as the inputs for the next layer in a feed-forward procedure until the output layer. In regression problems with unbounded outputs, a linear activation function is typically used in the output layer.

The training or learning procedure consists of optimizing the weights and biases using algorithms such as gradient descent, aiming to minimize the defined loss function that represents the prediction error on the training dataset. This optimization is made possible by the backpropagation algorithm [65], which efficiently computes the gradient of the loss function with respect to the weights and biases using automatic differentiation [66]. Figure 1.6 illustrated the neural unit and the training procedure of an MLP.

Once trained, the NN can readily make new predictions as it becomes a system of algebraic equations. Several factors should be considered to achieve accurate predictions and good generalization. Proper data scaling is essential for optimal performance. Regularization techniques, such as weight regularization and dropout, can help prevent model overfitting [67]. In addition, an essential step in improving NN performance is tuning its hyperparameters, which are the parameters set before the learning process, including the number of hidden layers, nodes per layer, and activation functions. Besides, validation sets are commonly employed to monitor

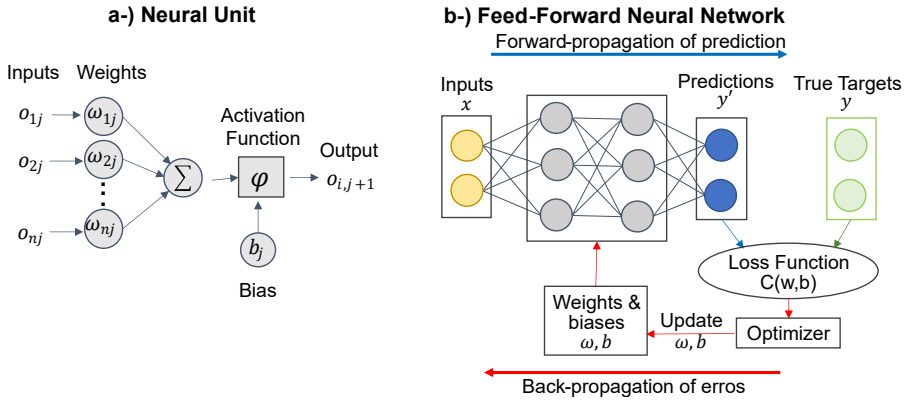


Figure 1.6: In the neural unit of a Neural Network, the weighted sum of the inputs is added to a bias and goes through a nonlinear activation function, firing the neuron output (a). Supervised training workflow of an MLP with backpropagation of the errors (b).

generalization during training.

DL architectures can be constructed by stacking several neural layers; in this way, the DL algorithm can learn higher-level representations of the data at each layer. DL algorithms have high capabilities of automatically extracting features and learning complex representations from large amounts of data [67]. Thus, DL can automate data preprocessing stages and process raw data in a general-purpose procedure [65]. However, DL models have many parameters to be optimized, being computationally demanding and prone to overfitting. Consequently, DL became viable and made a series of breakthroughs with the advent of big data and the increase in computational resources [68]. Although DL models tend to outperform shallow ML algorithms if sufficient data and computational power are provided [69], they have black-box properties, may suffer from convergence problems, and normally need a large dataset of labeled data [70], which is rare in SD&V applications.

Different types of NN architectures excel at specific tasks. For instance, Convolutional Neural Network (CNN) are designed to capture spatial patterns, while Recurrent Neural Networks (RNN) are suitable for sequential data. However, when it comes to surrogate modeling, which focuses on rearranging inputs to mimic system transfer functions rather than extracting information from high-dimensional data, the most commonly applied NN architecture is the shallow or deep MLP. Another NN architecture choice for surrogate modeling is the Radial Basis Function (RBF)-based network, which uses the RBF kernel as an activation function and, thus, can interpolate data by increasing the feature vector dimensions [64]. Probabilistic NNs [71, 72] are suitable for surrogate applications that require prediction uncertainties.

Although the literature on NN is dense and expands fast, several references cover the topic in a didactic way. The book in [62] contains comprehensive explanations of the NN main elements, while the book by Goodfellow et al. [59] has equally good NN introductions but covers more detailed and advanced aspects. Implementation

guides are available along with dedicated libraries for NN in MATLAB [73] and Python [67]. LeCun et al. [74] discussed practical recommendations for implementing NN. The authors recommend the survey in [75] to get a broad vision of DL, from their basic concepts to state-of-the-art algorithms, and the publications in [65, 76] for relevant DL applications and perspectives.

1.2.2.3 Gaussian Process Regressor

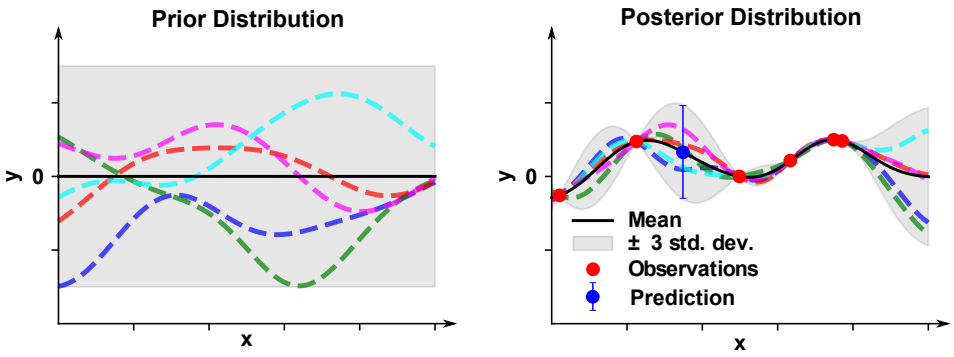


Figure 1.7: In Gaussian process regression, kernel functions define the prior distribution (left), and the posterior distribution (right) is updated with information from the observation points using Bayesian inference.

Gaussian process is a stochastic process that assumes a joint Gaussian distribution over all variables and is, thus, a distribution over functions. In GPR, it is first determined by a prior probability distribution, which embeds the beliefs about the unknowns before observing any data. Therefore, for any finite set of inputs $\boldsymbol{x}$:

$$m(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K}), \tag{1.1}$$

where $\mu_i(x_i)$ is the mean function, and $K_{ij}(x_i, x_j)$ is the covariance function, also known as kernel.

Given the training dataset $\{\boldsymbol{x}, \boldsymbol{y}\}$ and considering that new input-output pairs $\{x_*, y_*\}$ belong to the same joint Gaussian distribution, Equation (1.1) can be rewritten as

$$\begin{bmatrix} \boldsymbol{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \mu_* \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} & K_* \\ K_*^T & K_{**} \end{bmatrix} \right), \tag{1.2}$$

where $(K_*)_{ij} = K(x_i, (x_*)_j)$ and $(K_{**})_{ij} = K((x_*)_i, (x_*)_j)$.

During the regression task, the initial belief of the Gaussian Process $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$ is continuously updated as new observations become available. This process involves conditioning the prior distribution of the Gaussian Process on the observed data $\boldsymbol{y}_*$. The outcome of this conditioning step is the posterior probability distribution.

Figure 1.7 illustrates updating the prior with observed data leading to the posterior distribution. To account for noisy data, it is common practice to augment the prior distribution with Gaussian noise $\mathcal{N}(0, \sigma^2)$, where σ^2 represents the noise

variance. As the GPR depends on the kernel function used to model the prior distribution, the kernel hyperparameters can be optimized to maximize the marginal likelihood of the posterior distribution, or a prior distribution over hyperparameters can be defined for more flexible models [77]. A distinction of GPRs is that they provide a measurement of the uncertainty of the prediction once they output the prediction's mean and variance.

According to the literature review undertaken in this thesis, the GPR, also known as Kriging, is the most used ML algorithm in the SD&V design process. The GPR suitability for this application is due to it being a powerful predictor with small datasets, allowing to embed domain knowledge in the prior, its interpretability, and mainly because it provides probabilistic predictions used to maximize the information gained during sampling, as in the Bayesian optimization framework [78]. Furthermore, while parametric algorithms such as linear regression and NN make assumptions on the format of the underlying function $m(\boldsymbol{x})$, GPR makes a much less strong assumption of a prior probability to every function. However, GPR may result in poor prediction due to a bad choice of the kernel and problems in the optimization of the hyperparameters. Additionally, GPR does not scale well computationally with large datasets due to the need to evaluate the kernel at all training points.

1.2.2.4 Decision Trees-Based Methods

Decision Trees [79, 80] are statistical learning algorithms that apply recursive partitions of the space to perform classification or regression and that are the basic element of ensemble models, such as Random Forest (RF) and Gradient Boosted Decision Trees (GBT), as depicted in Figure 1.8. There are different Decision Tree algorithms, with CART (Classification and Regression Trees) being the most famous. The CART algorithm performs successively binary partitions of the input feature space using simple rules at each of their branches. Each split at a node n is defined by a threshold t_n applied to an input feature q_{i_n} and divides the space at the current branch $\mathcal{C}$ into two new subspaces (two new branches):

$$\mathcal{C}^-(q_{i_n}, t_n) = \{(\mathbf{q}, \mathbf{y}) \in \mathcal{C} : q_{i_n} \leq t_n\},$$
$$\mathcal{C}^+(q_{i_n}, t_n) = \{(\mathbf{q}, \mathbf{y}) \in \mathcal{C} : q_{i_n} > t_n\}.$$

Each node split aims to aggregate samples with similar outputs. Therefore, during the training process, the pair (q_{i_n}, t_n) that minimizes the splitting error at node n, also called impurity criterion $G_n(q_{i_n}, t_n)$, is chosen to split the domain. A common impurity criterion used in regression problems is based on the Mean Square Error (MSE) of each subspace, defined as

$$G_n(q_{i_n}, t_n) = \frac{L^-}{L}\left(\frac{1}{L^-}\sum_{y\in\mathcal{C}^-}(y-\mu^-)^2\right) + \frac{L^+}{L}\left(\frac{1}{L^+}\sum_{y\in\mathcal{C}^+}(y-\mu^+)^2\right), \tag{1.3}$$

where μ^- and μ^+ denote the mean output value of the subspaces $\mathcal{C}^-$ and $\mathcal{C}^+$, respectively, and L, L^- and L^+ are the number of samples in subspaces $\mathcal{C}$, $\mathcal{C}^-$

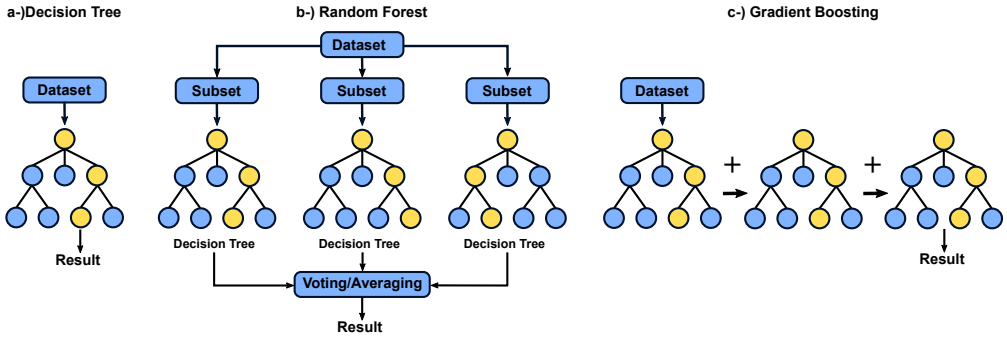


Figure 1.8: Individual Decision Tree partitions the space minimizing the classification/regression impurity (a). Random Forest uses Bagging to ensemble Decision Trees and gets the final prediction by majority or average voting (b). Gradient Boosting ensembles Decision Trees in sequence so that the individual prediction of one Decision Tree is improved by the next and so on (c).

and $\mathcal{C}^+$. The binary divisions of the Decision Tree continue successively until an established criterion is met, e.g., when the Decision Tree reaches its maximum depth. At the terminal nodes of regression Decision Trees, a scalar is assigned as output prediction for each final subspace.

Decision Trees are interpretable, fast, handle heterogeneous data and outliers, exempt data scaling, implicitly perform feature selection, and are nonparametric, allowing for models with nonlinear complex relations [81]. However, Decision Trees extrapolate poorly and are prone to overfitting, leading to high-variance and greedy solutions [58]. Although pruning is a regularization technique commonly applied to reduce the overfitting problem [82], it is usually through the combination of individual Decision Trees in ensemble methods that good generalization is achieved.

Any ML method can be used to create ensemble methods, but ensembles of Decision Trees are popular due to their randomized structure and low-cost training. The most popular ensemble methods are Bagging and Boosting. In Bagging, each predictor collaborates with a vote in the final prediction, and the predictors are constructed independently in a parallelized framework, reducing mainly the variance, as is the case of RF. In Boosting, the predictors are connected and sequentially organized, each trying to improve the prediction of the previous one, reducing mainly the bias, as is the case of GBT. Louppe [81], Chapter 8 by Mehta et al. [58], and Chapter 14 by Bishop [82] present the theory of ensemble methods and their close relation with the bias-variance trade-off.

RF is a powerful ensemble model that leverages the Bagging approach to combine multiple Decision Trees, reducing the variance of single tree predictions. For regression problems, the RF final prediction is the average of the trees' predictions, while for classification problems, it is determined through majority voting. To ensure improved generalization compared to individual trees, it is crucial that each Decision Tree within the RF ensemble is unique. This uniqueness is achieved by training each tree with a distinct subset of the training dataset, obtained through

random sampling with replacement from the available training data. Additionally, RF enhances diversity and decreases computational time by randomly selecting a subset of variables as partition criteria candidates at each node. Compared with NNs [83], RFs are computationally cheaper during training and prediction, demand fewer data, and are more robust to missing data and hyperparameters choice. Besides, RFs intrinsically perform sensitivity analyses, which increases their interpretability [84–86], as further discussed in Section 1.3.2.

GBT is an ensemble of Decision Trees based on the Boosting technique ([53], Chapter 16.4). In GBT, Decision Trees are trained sequentially, and each Decision Tree trains on the mispredicted data of the previous one. Thus focusing mainly on reducing the predictor bias. As the prediction error is differentiable, it can be minimized using a gradient descent optimizer.

In a regression problem, the first tree predicts the mean of the data, while the next tree tries to predict the residuals of the prediction of the first tree while minimizing the loss function. Therefore, the sum of the prediction of all the trees in the sequence will minimize the error and the problem complexity in a process called additive training. There are two main methodologies to implement GBT: XGBoost (eXtreme Gradient Boosting) [87] and AdaBoost (Adaptive Boosting) [88]. GBT methods usually outperform RF [67], with the downside that the GB algorithm cannot be fully parallelized.

1.3 ML-Based Digital Twin Surrogates for Structural Dynamics and Vibroacoustics Design

In recent decades, advancements in virtual prototyping in SD&V have helped meet the ever-growing consumer demands, providing higher quality products and speeding-up development times. However, with SD&V products becoming more complex, their design involves solving large optimization problems with many parameters. Therefore, while restricting the exploration to only a limited subset of parameter value combinations can result in sub-optimal outcomes, simulating high-fidelity models for the entire parameter space is impractical, as it requires immense computational resources. In light of these challenges, surrogate models have emerged as a crucial tool in prototyping SD&V DTs. Surrogate models offer the advantage of being significantly faster to simulate and more easily incorporate new data than traditional CAE models.

One of the primary obstacles in constructing surrogates for SD&V DTs lies in the intricate behavior of mechanical waves within the underlying physical system. In numerous studies, the response of the SD&V system has been approximated by polynomials using the RSM [13–15]. Typically, second-order polynomials are favored in RSM due to their sampling efficiency and high interpretability. However, these methods cannot capture nonlinearities that may exist within the system response. On the other hand, higher-order polynomials are rarely employed due to the costly surrogate training [89].

Therefore, in recent SD&V DT prototypes, ML-based surrogates have emerged as a promising solution. As discussed in Section 1.2, ML models can learn complex data relationships and nonlinear functions using a set of system observations. Once trained, ML surrogates demonstrate great computational efficiency when predicting the system's output given the input values. Thus, they can enable the straightforward and fast quantification of output uncertainties concerning the uncertainties in the physical system. ML surrogates have found diverse applications within SD&V DTs, including product optimization [90–92], uncertainty analyses [93, 85, 16], and knowledge discovery from measured databases [94].

However, two well-known drawbacks of ML surrogates are their poor interpretability and difficulty assimilating long-term historical data [12]. The harsh and discontinuous behavior of many SD&V analyses, particularly close to the system resonances, can also hinder the implementation of ML surrogate models [16, 17] as the local smoothness of the data is one of the main assumptions in ML [56, 58]. For example, the NN surrogate implemented in [95] showed good overall agreement in predicting the acceleration response of a structure as a function of the excitation and geometry but with inaccuracies at peak amplitudes, which usually are regions of interest in SD&V analyses. The next sections present the workflow for ML-based surrogate construction, related methods to overcome the above mentioned obstacles, and the main applications of surrogate modeling in SD&V.

1.3.1 Workflow of ML-Based Surrogate Modeling

Surrogate modeling is a statistical model that mimics the behavior of a true function $f(\boldsymbol{x}) = \boldsymbol{y}$. The surrogate models use statistical methods to map the relationship between a sample of inputs and the corresponding outputs, known as support points or training data. In this way, the surrogate generates a new approximate function $\tilde{f}(\boldsymbol{x}) \approx f(\boldsymbol{x})$ that generalizes the observed behavior of the true function and predicts outputs for new inputs $\tilde{f}(\boldsymbol{x}^*) = \tilde{\boldsymbol{y}}^*$. Thus, the surrogate provides a compromise between computational cost and fidelity.

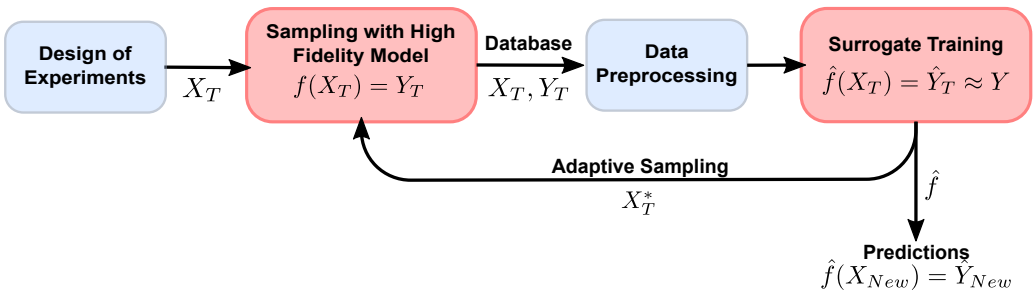


Figure 1.9: Steps to build a surrogate model: apply the design of experiments to define supporting points' location; sample results with the high-fidelity model; preprocess the data; train the surrogate model; predict new outputs using the surrogate model. Adaptive sampling is optional and may be applied to select new supporting points to update the surrogate model in regions of interest or uncertainty.

Figure 1.9 describes the basic steps to build a surrogate model. The initial step is to sample informative support points, which can be accomplished using the design of experiments theory. Commonly employed techniques include Latin Hypercube Sampling (LHS) and quasi-Monte Carlo, known for their favorable space-filling properties, low computational cost, and ability to capture parameter interactions [96]. Optionally, AS, also known as Active learning (AL), [97] can be employed to sample new optimally informative support points to update the surrogate that, consequently, can achieve good accuracy with fewer points. The selection of these new support points is guided by an acquisition function, which leverages information from previously sampled points to determine the best set of points to sample. AS can be applied to improve predictions locally where the surrogate may fail, and it can be framed as a Bayesian optimization approach for design optimization (see Section 1.3.3).

The next step is the selection of a suitable algorithm for surrogate modeling, considering various factors outlined in Section 1.2. In SD&V, statistical methods are commonly used to construct surrogates. These methods include Polynomial Chaos Expansion (PCE) [98, 99], RSM [15, 13, 100, 14], RBF interpolation [101, 102], low-rank tensor approximations [103], and spectral expansions [17].

Additionally, using supervised regression algorithms from ML has gained popularity in SD&V surrogate modeling. Algorithms such as SVM [104], GPR [105, 38], NN [106], Decision Trees-based algorithms [107] are increasingly being utilized. However, unlike many ML applications, surrogate models often learn from small datasets with low-dimensional inputs that represent the parameters of a physical system. Hence, feature extraction and selection techniques are less commonly used in surrogate modeling than in SHM and Active Control applications.

A popular ML algorithms choice for a surrogate is GPR since it has good performance with small and low-dimensional datasets, and its adaptability and probabilistic properties are appropriate for AS. Recently, Liu et al. [108] developed a methodology for informative power loss measurements in transmissions based on subset simulation, GPR surrogates, and AS. As discussed in [17], localized surrogates as GPR rely on the proximity of the support points and are suitable for interpolating. In contrast, global surrogates like NN extrapolate better but may have lower local accuracy. Additionally, global approximations with local refinements or domain-decomposition-based methods are suitable for functions with highly localized behavior in specific regions of the input space, as demonstrated in the analysis of a damped oscillator by Marelli et al. [17].

The highly localized behavior encountered in SD&V problems presents a significant challenge for surrogate modeling. To assess the ability of different algorithms to capture this non-smooth behavior, the present thesis conducts a benchmark study of ML algorithms for canonical SD&V problems in Chapter 2. AS approaches have been proposed in [18–20] to enhance surrogate accuracy in regions with high prediction error or irregular and nonlinear response behavior. Another promising trend to overcome this issue is PGML [29], discussed in Section 1.4.

1.3.2 Uncertainty Quantification with ML-Based Surrogates

Uncertainties are an inherent part of every phenomenon and computational analysis, and their quantification improves the comprehension of the phenomenon and the analysis's reliability level. Uncertainty quantification encompasses two key aspects: Uncertainty Propagation and Sensitivity Analysis, as indicated in Figure 1.10. Both analyses require multiple evaluations of the system model, making the use of surrogate models beneficial for reducing analysis time.

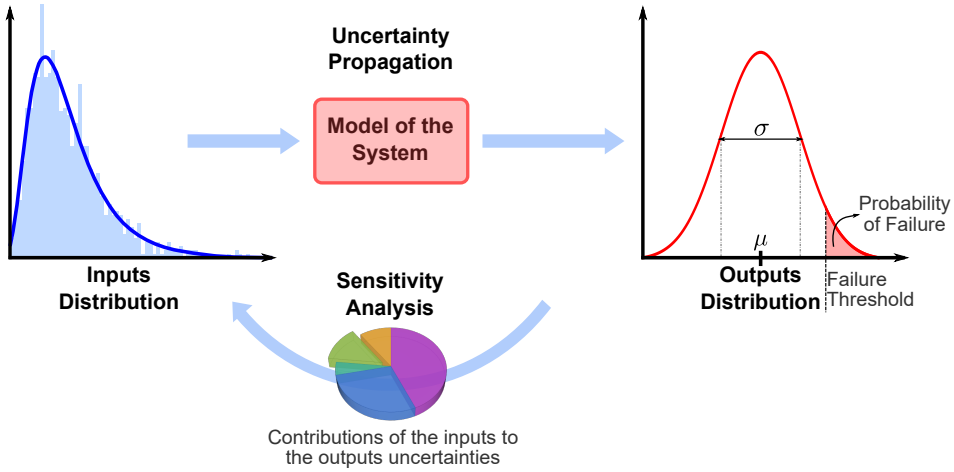


Figure 1.10: Uncertainty quantification framework: the uncertainty propagation analysis propagates the uncertainties of the input through the system model to obtain the output distribution, and the sensitivity analysis evaluates the input contributions to the output uncertainties.

1.3.2.1 Sensitivity Analysis

Sensitivity Analysis (SA) is a technique that examines the impact of input uncertainties on output variability. There are two main approaches to evaluating input influence: local SA, which examines the inputs individually at a specific point in the input space, and GSA, which captures input interaction effects and individual influences.

SA enhances the interpretability of surrogate models by revealing the influence of inputs on the system response. Certain surrogate models have intrinsic properties to evaluate sensitivity indices. For example, methods such as PCE, low-rank tensor, and RF [103, 86] provide the sensitivity indexes as a by-product of their training process. Additionally, sensitivity indices can be evaluated with minimal effort in NNs [109] and GPR [110].

Cheng et al. [111] presented an overview of GSA evaluated with surrogate models and compared their performances. Pizarroso et al. [112] listed several methods to analyze input-output relationships in ML-based surrogates to improve their interpretability. Gradient interpretability has also been studied in [113–115].

In [116], a GSA framework for surrogates that merge physical and numerical substructures is presented and applied to a structural dynamic problem modeled by a PCE-based surrogate. Stender et al. [117] implemented explainable surrogate models to identify and quantify uncertainties from a dataset of acoustic measurements. The proposed approach helped to find the sources of the measurements' variability, such as manufacturing and mounting aspects and specimen geometry. The out-of-bag-based GSA based on RF surrogates was performed by Chai et al. [85] in a STL problem. Despite the bias and smoothing effects presented by the RF surrogate, the out-of-bag-based sensitivity indexes agreed with the ones obtained by Fourier Amplitude Sensitivity Testing (FAST), with the former being more easily interpretable.

Apart from using SA embedded within the surrogate algorithm, classical SA approaches can be accelerated by replacing expensive model evaluations with surrogate predictions. In Chapter 3, we investigate both approaches, accelerated classical variance-based GSA and surrogate-embedded GSA, to enhance the interpretability of the surrogate, comparing their advantages and drawbacks. The MDI [86], a by-product of RF training, is employed as a surrogate-embedded GSA approach. Additionally, SA is applied to perform feature selection, enabling design optimizations with fewer variables (see Chapter 5).

1.3.2.2 Uncertainty Propagation

Uncertainty Propagation (UP) involves the propagation of input uncertainties through a model to assess the statistical moments and probability density function of the system response or failure probability [103]. Spectral stochastic methods like PCE and direct simulation methods such as Monte Carlo Simulation (MCS) are commonly employed for uncertainty propagation [118]. However, these methods often require many model evaluations, leading to significant computational costs. Thus, surrogate modeling techniques have been utilized to accelerate UP in SD&V.

For instance, Nobari et al. [119] used a surrogate based on polynomial and GPR to enable UP and GSA of the squeal instability analysis modeled with complex eigenvalue analysis. Similarly, Diestmann et al. [93] used GPR and PCE-based surrogates to accelerate the uncertainty quantification of NVH indicators in gear transmissions.

In the field of reliability and risk assessment, surrogate models can be highly beneficial, particularly in SD&V, where the non-monotonic behavior makes traditional vertex methods impractical for UP, requiring computationally expensive statistical methods [16]. Surrogates based on GPR and AS have been vastly applied in the literature to support these analyses with reduced costs. Cicirello et al. [16] used AS to build GPR surrogates that predicted the upper and lower bounds of the system response, reducing the number of true function evaluations required for UP. While the proposed method has good accuracy and is faster than a sub-interval method, it may encounter difficulties when applied to more complex analyses with higher dimensionality. Other studies, such as Guo et al. [120] and Guo et al. [121], also

employed GPR-based surrogates with AS to improve computational efficiency in reliability analysis and failure mode SA, respectively.

NN-based surrogates were employed in analyses of structural failure probability in [122], where the RBF-based network performed better in cases under static load, while MLP performed better in nonlinear dynamic analysis. Wang et al. [123] explored automatic differentiation in an NN surrogate to evaluate its first and second-order derivatives, which were then used to obtain response bounds using the subinterval method. Extremum RSM has been effective for nonlinear and time-varying analysis since just the extreme responses of the system are considered [124–126]. In some cases, surrogate models replace the entire UP framework by directly predicting the probability of failure [127].

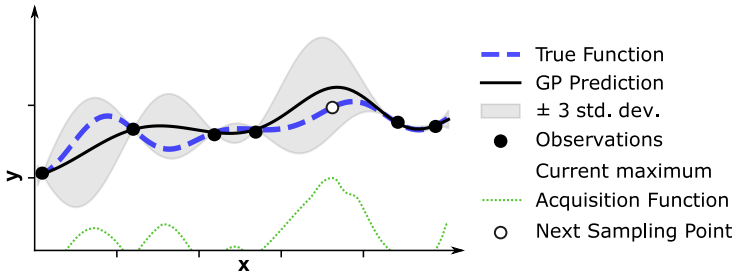
1.3.3 Design Optimization with ML-Based Surrogates

Design optimization is the process of finding the best design solution within given constraints. It utilizes mathematical algorithms to explore the design space and identify the optimal design configuration. The design optimization in the SD&V domain is commonly made feasible by surrogates, as otherwise, it may be prohibitive due to the expensive cost of the simulations. In surrogate-enabled optimization, there are two main approaches commonly used: direct replacement of the system function within the optimization algorithm and Bayesian optimization.

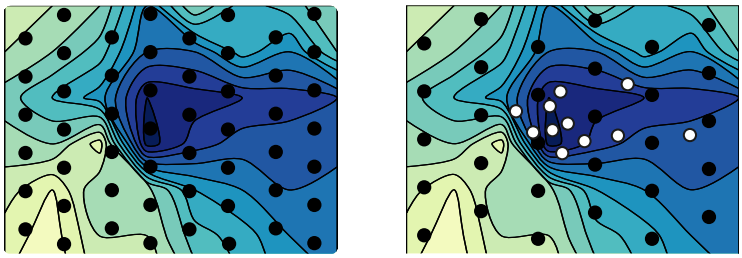
In the direct replacement approach, a surrogate model is integrated directly into the optimization algorithm, providing inexpensive evaluations of the system's behavior. Various types of surrogate models have been employed to enable optimization in SD&V using this approach, such as quadratic polynomial regression [15, 100, 128, 13], RBF-based interpolation [102, 129], GPR [130], and NN [131–134]. For example, in [135], the acoustic optimization of an electric motor was tackled through local surrogates replacing FEM, and different ML algorithms were evaluated as surrogates, namely linear regression, Decision Tree, SVM, and GPR. Bacigalupo et al. [92] carried out bandgap optimization of metamaterials supported by an RBF-based surrogate. GPR-based surrogates were applied to optimize transmission loss in intake systems [91], isolation of metamaterials [136], and vehicle NVH [130].

An NN-based surrogate was used instead of costly FEM simulations in [131] to perform parametric optimization of vehicle suspension hardpoints to minimize structure-borne road noise. The optimization approach used a criterion combining an up-limit and a matching target to the FRF curve to control amplitudes at specific frequencies and the frequency shift. The NN-driven optimization allowed good time-saving and an increase of problem dimensionality in comparison with polynomial and RSM surrogates previously used in [132]. Li et al. [133] used a surrogate based on Elman NN to minimize the vehicle sound pressure while constraining mass, side-impact intrusion, and first-order global modal. Instead of using the vehicle parameters, Tsokaktsidis et al. [134] used time-domain acceleration at the component level as input of an NN surrogate to predict the sound pressure level in the passenger

cabin.



(a) Gaussian Process Regressor (GPR) surrogate model and its acquisition function used to select new sampling points.



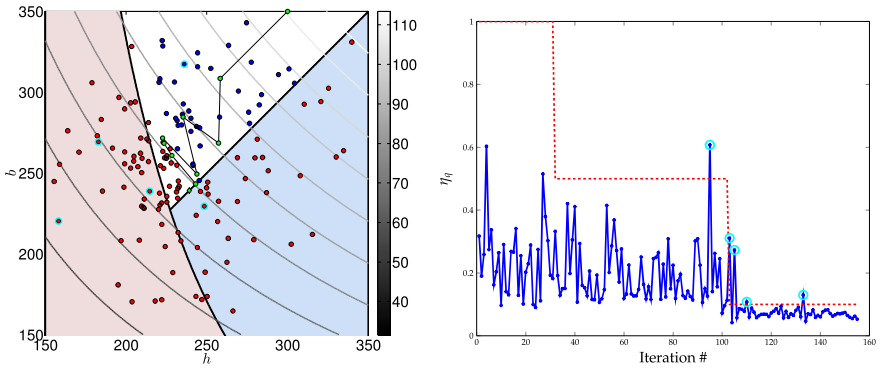
(b) Illustration of the initial observation points (black) and the new sampling points (white) for the cases without (left) and with (right) adaptive sampling.

Figure 1.11: Illustration of adaptive sampling or active learning.

However, sampling expensive-to-evaluate functions across the entire domain to build an accurate surrogate for global optimization can also be prohibitively expensive. Bayesian Optimization (BO) [137, 138] is an advantageous approach in this scenario, as it maximizes sample efficiency for expensive black-box models and, thus, enables global optimization with minimal function evaluations. BO relies on the prediction mean and variance to guide the decision-making of new informative sample locations. Beyond its application in surrogate-based product design [97], a vast field of research focuses on BO for optimizing expensive black-box models [138].

The acquisition function, often based on the Expected Improvement function[139], balances the exploration of new regions with high prediction variance and the exploitation of promising regions where the prediction mean is optimal [138]. This approach helps avoid getting trapped in local optima and facilitates accurate optimization within identified regions of interest.

One of the first studies applying BO to experimental design was the article *"Efficient Global Optimization of expensive black-box functions"* by Jones et al. [139]. Since then, BO has been applied to several optimization problems, such as to minimize disc-pad shape under squeal noise criteria with Efficient Global Optimization in [140], to optimize the modal characteristics of an engine using adaptive hierarchical GPR [141], and to optimize a mechanical metamaterial modeled by RBF-based surrogate in [142]. GPR is often the regressor used in BO as it provides the required



(a) Points sampled during CMA-ES in the design space

(b) Quantile accuracy criterion during optimization

Figure 1.12: Surrogate based RBDO of a column under compression performed by Moustapha [89]. Figure (a) shows the optimization path, including the enrichment points selected during adaptive sampling. Figure (b) shows how the accuracy criterion decreases with optimization iterations to guarantee the surrogate accuracy near the optimal point. The blue points are admissible; green points are the successive best points; the red points are unfeasible; cyan points are those around which enrichment has been done during optimization.

probabilistic outputs and performs well with sparse data. Figure 1.11 illustrates a GPR prediction with the respective acquisition function.

Surrogate models also play a significant role in Reliability-Based Design Optimization (RBDO), as both reliability analysis and optimization tasks involve multiple evaluations. Moustapha and Sudret [104] presented a complete survey on surrogate-assisted RBDO with detailed implementation details and several approaches to tackle the reliability analysis. Fei et al. [143] performed an RBDO of turbine blade radial deformation under dynamic loads using an extreme SVM surrogate and importance degree model. Zhang et al. [144] used a fuzzy multi-extremum RSM to perform an RBDO of fatigue and creep failures of a turbine blisk and achieved accuracy similar to the MCS in a fraction of the time.

A Reliability Efficient Global Optimization approach was implemented to optimize friction-type tuned mass damper-controlled structures in [145]. PCE-based Kriging was used to speed up the dynamic simulations in the RBDO of a passive control device to mitigate vibration [146]. In his thesis, Moustapha [89] used AS with adaptive accuracy criterion to minimize computational cost and improve accuracy in the RBDO of the crash analysis of a lightweight vehicle. This methodology was also applied to the RBDO of the buckling analysis of a column illustrated in Figure 1.12, which aimed to minimize the column cross area while keeping the probability of failure under 5%. Global approximations with local refinements and domain-decomposition-based approaches are also used to ameliorate surrogate accuracy in regions of interest [17].

In Chapter 5 of this thesis, surrogate models are applied to accelerate optimization and UP. The surrogates directly replace gear transmission analysis to enable

robust design optimization of the gear transmission NVH performance. The approach considers gear teeth manufacturing uncertainties and evaluates the design robustness through the MCS.

1.4 Future Trends and Perspectives of Digital Twins and Machine Learning in Structural Dynamics and Vibroacoustics

Digital transformation is already a reality and has been changing how to solve several industrial problems, including virtual prototyping of mechanical systems, traditionally analyzed solely by physical models. The works referenced in this chapter illustrate how this transformation is taking place and bringing advantages to the SD&V field. Despite the progress, much should be done to scale and take full advantage of the benefits offered by digital transformation. This section aims to explore two emerging topics in ML research that can support the implementation of DTs in SD&V: *Physics-guided ML*, in which physics knowledge is embedded into data-driven methods to support the learning of consistent representations; and *Transfer Learning*, which make use of existing data from related tasks to enhance accuracy performance in scenarios with limited available data.

1.4.1 Physics-Guided Machine Learning

The big drawback of ML models, mainly when applied to physical problems, is the lack of a theoretical foundation and interpretability, raising skepticism by part of the scientific community. Indeed, ML models may lead to physically inconsistent results, may fail to generalize to unseen scenarios and depend on the availability of big data. However, physics-driven models rely on hypotheses and simplifications of the real boundary conditions and struggle to account for uncertainties and historical and environmental conditions. PGML is an incipient but fast-growing research field that suggests merging physics-driven and data-driven models to take the best of both worlds, as shown in Table 1.1 [147, 148, 29, 149, 150].

Recent reviews in [147, 151, 29, 148, 152] classify and describe PGML works developed in different domains. The state-of-art of PGML is described here according to the configurations in which the physical knowledge is merged with the ML algorithm, as illustrated in Figure 1.13. Two categories can be defined: *Physics Leveraged by ML*, in which ML models are used to improve the results from the simplified physical models; and *ML Leveraged by Physics*, in which physical laws and constraints are intrinsically embedded into the ML, guiding it to have consistent physical results. Willard et al. [29] presented a similar categorization and associated each PGML configuration with applications for which they are appropriate.

One way ML can leverage the results of physical simulations is when the results of the latter, and possibly its inputs, are used as ML input, which was named In-Series Hybrid Model in Figure 1.13. The ML is trained to correct the results of

Table 1.1: Advantages of PGML distinguishing the contributions of the data-driven approach and the embedded physics knowledge.

Contributions from data-driven models to PGML	**Contributions from physics embedded in PGML**
Improve state-of-the-art physical models by comprising unknown relations;	Improve ML predictions with domain knowledge and inductive bias;
Computationally cheap to evaluate;	Provide physically consistent models;
Handle noisy input;	Reduce or end need of data;
Reduce model order;	Increase interpretability of ML model;
Estimate aleatory and epistemic errors bounds;	Improve ML generalization for unseen scenarios;
Mitigate instability issues in time integrators;	Reduce search space of ML algorithm
Provide lagged predictions to active control;	Improve long term-forecasting.
Discover governing equations and unknown physics;	
Solve inverse problems and lead to better parameter identification in the physical model.	

the physical model by using the real system output as the training target [38, 153]. Similarly, in residual modeling, ML learns to model the error of the physics-driven model, and therefore, the ML can correct the model output or classify its validity, as in [154–157, 149, 158]. Finally, the ML model can be used just as a sub-process of the physics-driven model to evaluate one of its parameters [159, 47, 160–163].

In the configurations that physics improves ML, the structure is case-specific since it depends on the physical equations that govern the system. The most common approach is physics-guided loss [153, 164–170], in which the loss function contains penalization terms for non-physical predictions, e.g., an unexpected non-monotonic behavior. A thorough case of physics-guided loss is the Physics-Informed Neural Network (PINN) [171–181], in which the loss function is solely composed of the residue of a partial differential equation formulated in its derivative form. The equation variables are also the NN inputs, and therefore, the residue, i.e., the loss function, is minimized by using automatic differentiation [66], and the equation is solved with no data needed.

Another popular approach is *physics-guided architecture*, in which the physical behavior is incorporated in the ML model architecture, as is the case of sequential

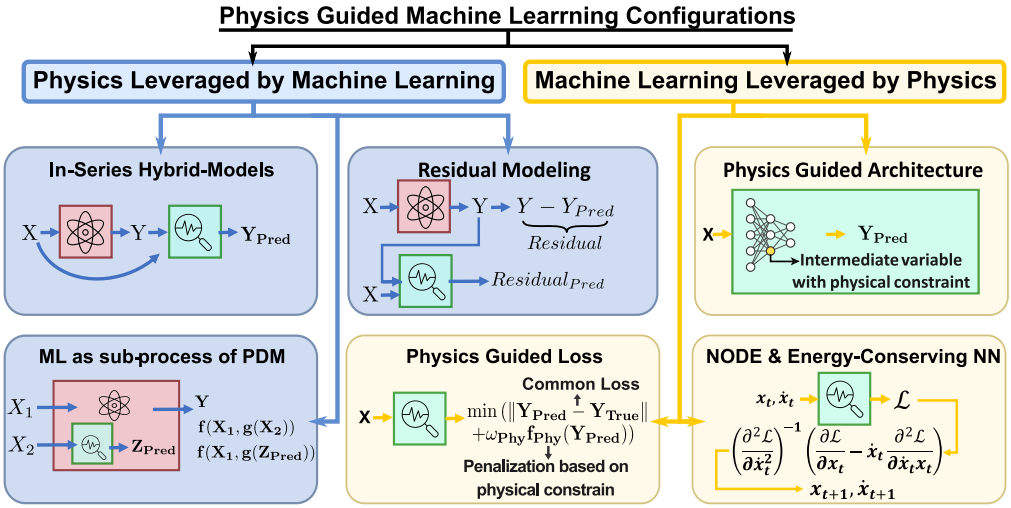


Figure 1.13: Configurations of physics-guided ML merging data-driven and physics-driven models. The representations are illustrative and do not show the arrangement precisely nor include all configurations.

behavior in RNN or spatial perception in CNN. Expected physical behavior can be embedded through constraints in the weights and biases [182] or intermediate variables [183, 167] of NN architectures. Zhang et al. [184] used Long Short-Term Memory (LSTM) and graph-based tensor differentiator to enforce physical constraints in the architecture and loss-function of metamodels of nonlinear structural systems. Besides improving the prediction accuracy and robustness, the PGML implemented in [184] models non-observable latent nonlinear state variables, such as the hysteretic metric and nonlinear restoring force, delivering a more interpretable surrogate.

Domain knowledge should also be incorporated into GPR by selecting a kernel that best defines the correlation of the different components of the underlying physics of the problem. The authors of [185] present a framework to select stationary and non-stationary kernels based on the characteristics of the domain, such as symmetry and periodicity. The kernel of a GP can also be designed using knowledge of the system's equations of motion, which can lead to more interpretable hyperparameters and better prediction accuracy compared to non-physics-aware kernels when few data points are available, as shown in [186]. Other methods to embed physical knowledge into GP priors for SHM problems are addressed in [187], including using simple physical models as the prior of the mean function and residual modeling with GP-NARX [188].

Elements of physics-guided loss and physics-guided architecture are used in Neural Ordinary Differential Equation (NODE) and energy-conserving neural networks. In NODEs, explicit integration steps are performed in each layer of the NN as one step evaluation of a standard ODE solver [189–193]. In energy-conserving NN, the structure of Lagrangian and Hamiltonian equations have been embedded into the NN construction to ensure an energy-conservative behavior, as reviewed by Lutter

and Peters [194] and implemented in different structures in [195–202, 189, 203].

The survey by Willard et al. [29] presents other PGML approaches, while Ba et al. [204] merged several PGML approaches to create an NN able to generalize well to different mechanical problems. Although it is a new topic, several recent works employed PGML, underlying its potential. However, most PGML research concentrates on other fields, e.g., fluids dynamics [155, 205, 206], lake modeling [207, 167, 164, 165, 153], climate modeling [208, 209], and material science [210–212]. The extensive use of PGML techniques in the SD&V is a burgeoning research field with emerging opportunities.

For example, NODE and energy-conserving NN can be used as time integration solvers for modeling SD&V problems, and the physics information could help solve the difficulties posed by the non-smooth behavior of these problems. Recently, Yu et al. [213] created a PGML model of a structural dynamic system using an RNN encoding the equation of motion. The PGML model showed superior results even for scarce and noisy data, with better generalizability and robustness than solely data-driven models. Besides, it allowed computational time-saving by applying big time-steps without facing stability issues from the purely physics-based approach. NODE and energy-conserving NN are suitable for introducing inductive biases in dynamic systems. Examples of how this approach can improve ML performance under high nonlinearities and discontinuities are presented in recent works which applied energy-conserving NN to improve data efficiency in non-smooth contact dynamics problems [214, 215].

Yin et al. [216] introduced the APHYNITY framework to augment physical models with data information applied to dynamics forecasting. The residual modeling approach takes into account the contributions of both physics and ML models to the final response. Besides, the developed approach ensures that the ML response has minimal influence on the final prediction. Therefore the physics-based model explains as much of the prediction as possible. In addition, the APHYNITY framework applied energy-conserving NN to ensure physical consistency. In problems such as reaction-diffusion equations, wave equations, and nonlinear damped pendulums, the study demonstrates how APHYNITY outperformed both the simplified physical-based approach and the solely data-driven approach. Moreover, it improved the identification of physical parameters.

The PGML techniques hold great potential for enhancing the performance of ML models in SD&V applications mentioned throughout this chapter. By incorporating PGML, ML models' coherence, interpretability, data efficiency, and reliability can be significantly improved.

In this thesis, we explore the use of domain knowledge to enhance DT modeling in SD&V by employing PGFs. The PGF approach performs basic operations on input data using feature engineering techniques, thereby imbuing the ML model with domain-specific insights. Thus, the approach is an instance of an In-Series Hybrid Model case of PGML (see Figure 1.13). The simplicity of the PGF approach allows it to be seamlessly integrated into the DT framework, applicable to different ML methods and different SD&V problems.

1.4.2 Transfer Learning in SD&V: Bridging the Data Gap

One of the most significant hurdles faced when utilizing ML in the early stages of product development is the scarcity of data. Traditional ML algorithms typically demand substantial amounts of labeled data to perform satisfactorily. However, acquiring such data can be costly, time-consuming, or unfeasible in many real-world scenarios. Consequently, exploring alternative solutions to overcome data scarcity in ML has become an active research area. A prominent solution that has gained significant attention in recent years is TL, which is also being explored in the SD&V domain.

TL is a learning framework in ML that aims to use the knowledge acquired from one or multiple source domains to improve performance in a related target domain [217]. By leveraging the information learned from the source domain, where data is relatively abundant, TL reduces the need for labeled data in the target domain. In recent years, several TL algorithms have been proposed in the literature [217, 70, 218], demonstrating their potential to enhance the performance of DL models that traditionally rely on large datasets, particularly in computer vision and natural language processing.

In the field of SD&V, TL has shown promise in various applications. For instance, TL has been employed in SHM to transfer damage information and inference between similar instances within a population, which is also known as Population-Based SHM (PBSHM) [42, 43, 219–222] and aligns with the concept of DT Aggregates (Section 1.1). PBSHM can be categorized into homogeneous and heterogeneous populations [42]. In homogeneous PBSHM, instances are nominally-identical but are subject to variability in the parameters and in the environmental and operational conditions. For example, [43] utilized GPR to approximate the behavior of a wind farm population with structurally-equivalent wind turbines. In heterogeneous PBSHM, instances differ but share certain similarities. Feature-based TL approaches have been employed in both homogeneous and heterogeneous PBSHM to reduce feature distribution discrepancies between source and target domains [223, 70, 224, 225]. Alternatively, feature-based TL combined with graph representations, which identify structural similarities between instances, have been used to transfer information among distinct instances [219, 221, 226].

Another relevant application of TL in SD&V is the transfer of knowledge from numerical simulations to realistic scenarios, i.e., virtual models generate abundant numerical data in the source domain, which TL leverages to enhance predictions based on limited experimental data in the target domain. For example, Ai et al. [227] used FEM to generate numerical acoustic emission signals and utilized unsupervised domain adaptation to reduce the distribution difference between the numerical and the realistic domains. Their results suggest that the proposed approach can accurately localize acoustic emission signals without labeled training data, being helpful for damage detection. Similarly, Trinh et al. [228] employed TL to augment scarce experimental datasets with simulated data. Their approach effectively integrated experimental measurements with multiscale numerical models,

enabling accurate prediction of acoustic responses parameterized by microstructural properties despite the scarcity of experimental data.

While TL has demonstrated success in various applications and the growing use of ML to create surrogates, few studies are exploring the use of TL to enhance the creation of DT during product design. Recent developments have shown the effectiveness of TL for constructing multi-fidelity surrogate models. For instance, Jiang and Durlofsky [229] successfully applied TL techniques to develop a surrogate model for subsurface flow prediction. Their approach involved training an NN with low-fidelity simulation data and fine-tuning it using a limited number of high-fidelity simulation data. The resulting surrogate model achieved a remarkable 90% reduction in training simulation costs while maintaining accurate predictions of subsurface flow.

Another noteworthy study by Geng et al. [230] employed a deep reinforcement learning algorithm in conjunction with a multi-fidelity surrogate model constructed using TL to optimize propeller blade design while considering aerodynamic and aeroacoustic performance. The multi-fidelity surrogate was constructed using samples from a low-fidelity frequency domain method and a high-fidelity Computational Fluid Dynamics model in the time domain. TL techniques were leveraged to enhance both the multi-fidelity surrogate and the deep reinforcement learning-based optimization algorithm, demonstrating significant computational cost savings and noise reduction. Alternatively, Kim and Lee [231] proposed an instance-based TL method using modified domain-adversarial NN. This method utilized domain adaptation techniques to generate auxiliary target-domain data from a source domain and selects useful data to improve model accuracy. The proposed method was validated in study cases for drone surrogate modeling and bearing fault diagnosis, showing significant improvements in prediction accuracy despite limited data availability.

Despite these developments, a recent survey by Borisov et al. [232] highlighted the absence of well-established TL methodologies for low-dimensional tabular data, which is common in most surrogate models. Additionally, there is often a conditional shift between source and target tasks in surrogate applications, whereas many TL methods, such as feature-based methods, assume a covariate shift in the data Liu et al. [233]. These research gaps underscore the need for further exploration of TL algorithms within the surrogate context, including searching for suitable algorithms and exploring the viability of new applications, such as the transfer of knowledge between related product designs.

1.5 Conclusions

This chapter explores the integration of ML algorithms and DTs in the realm of product design in Structural Dynamics and Vibroacoustics (SD&V) analysis, within the context of digitalization and big data. It emphasizes the crucial role of ML in processing and interpreting the vast amounts of data generated in the digital age, enhancing the capabilities of DT surrogates.

The chapter first provides an extensive introduction to ML fundamentals and offers guidance on selecting suitable algorithms for DT surrogates in the SD&V domain, considering factors such as application objectives, database size, computational resources, problem dimensionality, data nature, uncertainty management, and system nonlinearity. Then, it presents how ML algorithms can enhance the performance of DT surrogates for SD&V design. At last, it discusses the current trends in ML applications for SD&V surrogates, focusing on two approaches: Physics-Guided Machine Learning (PGML) and Transfer Learning (TL).

The current state-of-the-art examination reveals the major challenges when applying ML to SD&V design. These include the challenge that ML encounters when dealing with non-smooth behavior, the lack of interpretability and physical constraints of ML models, as well as the heavy reliance on big data—often a scarce and expensive resource, particularly during the first stages of product design.

Consequently, the following chapters address these concerns by leveraging the relevant techniques identified in this chapter. Chapter 2 explores ML algorithms for non-smooth behavior, while Chapter 3 improves ML interpretability through classical and ML-based GSA. Chapter 4 investigates the viability of deep TL in surrogate modeling and propose its application to enhance inter-generational knowledge transfer and data efficiency. Physics-Guided Featuress (PGFs) enhance accuracy physical consistency and efficiency in all chapters. In Chapter 5, the integration of these techniques culminate in an inter-generational DT framework, applicable to practical industrial SD&V projects. The DT serves as a tool to support SD&V design and overcome major limitations from current ML-based surrogates.

Chapter 2

Machine-Learning Based Surrogates for Canonical Vibroacoustic Analyses

Surrogate models serve as data-based approximations for computationally expensive simulations, enabling efficient exploration of a model's design space and facilitating informed decision-making across various physical domains. However, employing surrogate models in the Structural Dynamics and Vibroacoustics (SD&V) domain poses challenges due to the non-smooth and complex behavior of wave phenomena. This chapter investigates four ML approaches for modeling SD&V analyses, using the canonical surrogate Sound Transmission Loss (STL) problem as a study case. The influence of the STL analyses' model complexities, materials, and design spaces are considered. Furthermore, Physics-Guided Featuress (PGFs) are employed to enhance the ML models' accuracy and physical consistency. The results obtained demonstrate that NN surrogates incorporating PGFs outperform other ML models in accuracy across different STL models. The transferability of the proposed techniques to other problems within the SD&V domain is discussed, along with an analysis of potential limitations associated with the models.

A portion of this chapter's work has been previously published by the authors in reference [107], Applied Sciences journal, and reference [234], International Conference on Noise and Vibration Engineering.

Contents

Introduction

Surrogate models have been widely used as effective decision-making and risk management tools in various domains [235–237]. These surrogates allow for the construction of DTs and enable the creation of optimal and reliable designs, online monitoring, and real-time control.

Recently, surrogates in the SD&V domain have increasingly relied on ML techniques. This shift is mainly due to the improved accuracy and reduced evaluation times offered by ML approaches compared to traditional methods. However, to ensure the soundness of further surrogate-based developments in the SD&V domain and the reliability of surrogate-enabled solutions, it is imperative to develop guidelines for constructing accurate ML-based surrogates considering the particulars of the domain.

One of the key challenges in constructing accurate ML-based surrogates in the SD&V domain is the presence of non-smooth response functions. ML methods typically assume the local smoothness of the data, thus making modeling non-smooth functions particularly challenging. While several studies have applied ML-based surrogates in SD&V, there is a lack of comprehensive comparisons regarding the overall performance of different ML algorithms when dealing with vibroacoustics analyses.

This chapter has two main objectives. Firstly, we aim to identify effective ML methods in modeling non-smooth functions within the SD&V domain. To accomplish this, we conduct a benchmark analysis of four popular regression ML models and determine which one is best suited for creating surrogates in SD&V. Our second objective is to explore techniques for incorporating domain knowledge into SD&V surrogates, to enhance physical consistency and potentially improve the accuracy of ML models. To achieve this, we apply algebraic transformations to the system inputs, aligning them with existing SD&V physical models. Throughout this chapter, our investigations are centered around the classical vibroacoustic problem of STL. By focusing on this problem, we can provide concrete examples and demonstrate the effectiveness of our approaches in the context of SD&V.

The main contributions of this chapter can be summarized as follows:

1. Series of numerical experiments to assess the accuracy of ML-based surrogates in effectively modeling non-smooth SD&V problems.

2. Identification of ML algorithms that are well-suited for different levels of smoothness and complexity present in the target function.
3. Implementation of physics-guided features to embed domain knowledge into otherwise black-box ML models.
4. Study of the impact of changing the input space on the performance of ML surrogates.

The chapter is organized as follows: Section 2.1 presents the methodology and implementation details of the ML-based surrogates. Section 2.2 describes the STL phenomena and the various methodologies employed to model them. Benchmarking analyses comparing the performance of ML algorithms to predict diverse STL problems are covered in Sections 2.3. Specifically, Section 2.3.1 focuses on STL of infinite plates with different material properties, while Section 2.3.2 compares the performance of different physics-based models, which result in varying degrees of non-smoothness in the STL response. The strengths and limitations of each ML method are discussed, along with an investigation into the effects of incorporating PGF in the surrogates. In Section 2.3.3, we analyze the influence of design space size and location on the ML-based surrogate performance. Finally, Section presents the chapter's conclusion and discusses the transferability of the findings to other vibroacoustic problems. The data and algorithms used in this work are available on GitHub (Accessed 14 June 2023).

2.1 Guidelines for Implementing Physics-Guided ML Surrogates

A surrogate model is a computationally inexpensive mathematical approximation of a complex simulation. More formally, let $f : \mathcal{X} \to \mathcal{Y}$ be the simulated function. The surrogate model can be defined as a function $\hat{f} : \hat{\mathcal{X}} \to \hat{\mathcal{Y}}$, where $\hat{\mathcal{X}} \subset \mathcal{X}$ is the set of interest for the surrogate model, and $\hat{\mathcal{Y}}$ is the associated codomain. The relation between the simulated function and the surrogate is given by:

$$f(x_i) = y_i = \hat{f}(x_i) + \varepsilon_i, \ \forall x_i \in \hat{X}, \tag{2.1}$$

where ε_i is the surrogate's prediction error. The predictor $\hat{f}(x)$ is fitted with N input/output pairs $\{x_i, y_i\}$ sampled from the simulated function to minimize the error ε_i for all inputs in the set of interest. In the following, the steps for constructing the ML surrogate models are illustrated in Figure 2.1.

2.1.1 Sampling Methods

The first stage in designing a surrogate model is to define informative supporting points using an adequate sampling strategy to keep the number of samples N small due to the high computational cost of evaluating $f(x)$. In random sampling, all points in the interval of interest have an equal probability of being drawn. Although

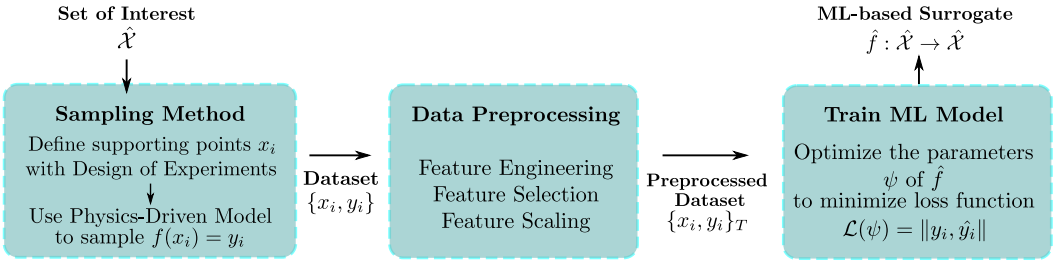


Figure 2.1: Workflow of ML-based surrogate models.

having low bias and being easy to deploy, random sampling can lead to gaps in the input domain. Thus, we employ LHS to define the supporting points for the training of the surrogates. The LHS approach segments the input space into M intervals along each dimension, and randomly samples from each interval, better depicting the target function's input domain [235].

2.1.2 Physics-Based Ground Truth Function

A physics-based function $f(x)$ provides the physical guidance to train the surrogate. Therefore, it should provide a close representation of the physical problem. In the literature, different physical-based models capture the complex behavior of the STL problem with distinct levels of detail [238]. This chapter implements four STL methodologies described in Section 2.2 to evaluate the ML-based surrogate capabilities.

2.1.3 Data Preprocessing and Physics-Guided Features

The accuracy of data-driven models can improve significantly with the proper operations over the feature set. Here, features refer to the different dimensions of the input data $x \in \mathcal{X}$. In feature engineering, domain knowledge is used to select the most informative features of an input set and to add PGFs through basic operations over the raw data to improve the predictor's accuracy. Feature scaling is another crucial preprocessing step, especially for ML predictors which are distance-based, for example, GPR, or gradient-based ones, such as NN. The goal of feature scaling is to remove the effect of the different range of values in the input so that equal importance is given to each feature in the fitting of the regressor. In addition, it guarantees that the gradient descent is updated with similar rates for all features, improving convergence. Normalization and standardization are the main methods for feature scaling.

2.1.4 Machine Learning-Based Predictor

The predictor function $\hat{f}(x)$ can be fitted with different ML models. This chapter investigates four popular ML regression algorithms that do not impose strong assumptions on the form of the modeled function, namely NN (see Section 1.2.2.2),

GPR (see Section 1.2.2.3), RF, and GBT (see Section 1.2.2.4). The ML models' hyperparameters were selected based on a trial-and-error search and are as follows:

NN-based surrogate models. The NN were implemented using *Keras* with *tensorflow* as backend. One regressor was trained to output all STL predictions at $n_{outputs}$ frequencies. Standardization was applied to all input features, while the outputs were scaled in the range [0, 1]. The fully connected NNs had an architecture with n_{inputs} nodes in the input layer, five hidden layers with 32 nodes each, and $n_{outputs}$ nodes in the output layer, where n_{inputs} is the number of input features. All hidden layers had the *sigmoid* as activation functions and an L2 regularization penalty of $1e^{-7}$. The training was performed with batches of 32 samples and 1500 epochs, except for the MS and FEM models, in which the NN trained for 2500 epochs. The Adam algorithm was applied to minimize the average MSE of the STL curve prediction.

GPR-based surrogate models. The kernel k used to construct the GPR used a constant kernel k_C, a Matern 3/2 kernel k_M, a RBF kernel k_{RBF} and a white noise kernel k_σ combined as $k(x_i, x_j) = k_C(x_i, x_j) * k_M(x_i, x_j) + k_{RBF}(x_i, x_j) + k_\sigma(x_i, x_j)$. The kernel hyperparameters were optimized to maximize the log-marginal likelihood of the posterior distribution with ten restarts. One GPR predicted the STL at $n_{outputs}$ frequencies. The preprocessing of the inputs and outputs was the same as the NN-based surrogate models.

RF-based surrogate models. An RF was trained to output the STL for the analyzed frequency range. The RF was made up of 200 CARTs without a depth limit. The impurity criterion was the mean of the averaged squared error among the $n_{outputs}$ outputs.

GBT-based surrogate models. The GBT predictor comprised 125 decision trees with a maximum depth of 10 nodes. The learning rate of the training was set to 0.05. A different GBT predictor was trained for each predicted output willing to minimize the MSE.

2.2 Case Study: Sound Transmission Loss (STL) - Definition and Methodologies

When an acoustic wave hits a plate, it is partially reflected in the incident acoustic field and partially absorbed by the plate itself. Most of the absorbed portion excites the plate, making it vibrate and radiate noise in both the incident and transmitted fields, as illustrated in Figure 2.2. The STL is defined by the ratio between the incident sound power P_I and the sound power transmitted through the plate P_T, being usually evaluated in decibels as

$$STL = 10 \log_{10} \left| \frac{P_I}{P_T} \right|^2. \tag{2.2}$$

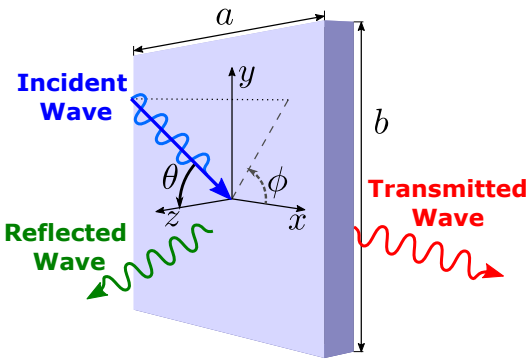


Figure 2.2: Illustration of the STL analysis of a plate.

The transmitted noise is inversely proportional to the system effective impedance, which is the sum of the mechanical impedance of the plate and the fluid-loading impedance [239]. The effective impedance depends on the fluid properties, the incident θ and azimuth ϕ angles of the incident wave, the material, dimensions, and boundary conditions of the plate. At frequencies below the first natural frequency of the plate, the plate dynamics influence on the STL is negligible, and the STL is controlled mainly by the plate bending stiffness D.

When the driving frequency is close to the first natural frequencies of the plate, the resonant behavior of the plate governs the STL. As the frequency increases and the resonant behavior attenuates, the plate behavior approaches that of a limp mass, and thus, the STL depends solely on the surface mass density

$$m = h\rho, \tag{2.3}$$

where h is the thickness and ρ is the density of the plate.

The STL can be approximated by the mass law at this frequency range, as demonstrated in [239]. The limp mass assumption and the mass law approximation are valid for thin plates in frequencies where the bending stiffness is irrelevant.

With the increase in frequency, the coincidence phenomenon occurs, and the mass law is no longer valid. The coincidence happens when the trace velocity of the acoustic wave projected in the plate is equal to the velocity of its natural bending wave. These waves superpose and create a scenario of minimal impedance, maximizing sound transmission. The coincidence frequency ω_{coinc} in which the trace and the bending wavenumbers coincide is evaluated by:

$$\omega_{coinc} = \frac{c_0^2}{\sin^2\theta}\sqrt{\frac{\rho h}{D}}, \tag{2.4}$$

where c_0 is the characteristic sound speed, and ρ is the plate density. The minimum frequency in which coincidence occurs is for grazing incidence ($\theta = 90°$) and is called critical frequency ω_{crit}, where the plate damping controls the amplitude of the valley in the STL. Above the critical frequency, the coincidence phenomenon occurs for gradually smaller incidence angles, and the STL is controlled by the plate bending

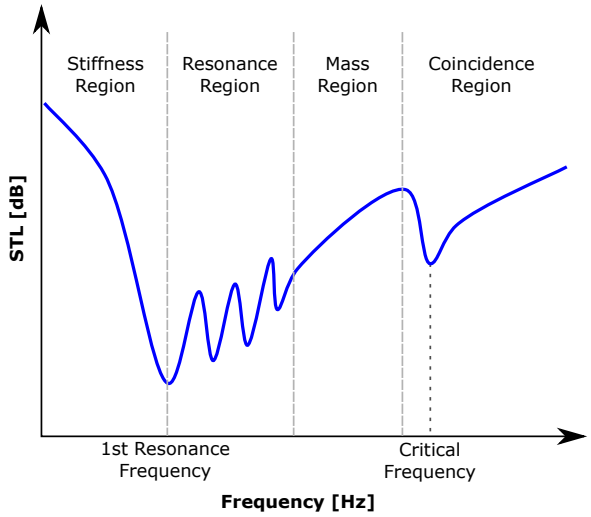


Figure 2.3: Typical STL curve of an isotropic plate as a function of frequency. The stiffness, resonance, mass, and coincidence regions are represented.

behavior. Figure 2.3 illustrates the typical STL curve of an isotropic plate and its influence regions.

Four different methodologies with increasing complexities will be considered to model the STL of a plate. The higher the complexity, the higher the expected level of non-smoothness, allowing for the evaluation of ML performance with increasingly challenging levels. The first and simplest methodology considered is the analytical STL solution for an infinite plate, for which different material properties are considered, namely isotropic, orthotropic, and sandwich materials. As the unbounded infinite plate is not affected by the resonances, the STL in low frequencies relies mainly on the surface density mass. To account for the plate width a and length b, the radiation efficiency of a finite plate may be considered instead of the radiation efficiency of the infinite plate. Spatial windowing and Rayleigh-integral-based techniques can be used for this purpose, resulting in a correction factor applied to the infinite transmission transparency [238] that modifies mainly the results below the critical frequency, where the plate modal behavior determines the sound radiated.

However, detailed STL evaluation of finite plates must account for its structural response under forced vibrations. The current work implements Modal Summation (MS) and the FEM to fully model the vibroacoustic phenomena in the STL of an isotropic finite plate with simply supported boundary conditions. In these approaches, the STL has a highly rough behavior as the STL is controlled by the plate modes' natural frequencies and radiation efficiencies control. Furthermore, in practical scenarios, only the overall STL behavior defines the plate design; thus, the band-averaged STL can be considered. Therefore, the results obtained with MS and FEM are also evaluated in one-third octave frequency bands. In summary, this study employs various physics-driven models of STL using four different methodolo-

gies and considers three different materials as the basis for establishing the accuracy of surrogate models, as enumerated below:

1. Analytical solution for infinite plates (Section 2.2.1) with:
   (a) isotropic materials,
   (b) orthotropic materials, and
   (c) sandwich materials.
2. Correction factor approach for STL of finite isotropic plates (Section 2.2.2).
3. Analytical MS approach of finite isotropic plates (Section 2.2.3)
   (a) without band average, and
   (b) with band average.
4. Numerical FEM of isotropic plates (Section 2.2.4)
   (a) without band average, and
   (b) with band average.

The following sections describe the above methodologies conform to the notation in Figure 2.2. The plate is in the plane $z = 0$, separating the source field $P|_{z>0}$ and the transmission field $P|_{z<0}$, both filled with a light fluid with density ρ_0 and characteristic sound speed c_0. Considering that a harmonic plane wave with wavenumber vector

$$\boldsymbol{k} = \begin{pmatrix} k_x & k_y & k_z \end{pmatrix}^T = \frac{\omega}{c_0}\begin{pmatrix} \sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \end{pmatrix}^T \tag{2.5}$$

impinges the plate with incident angle θ and azimuth angle ϕ, the resultant incident acoustic pressure field is

$$P|_{z>0} = P_I + P_R = p_I e^{i(\omega t - k_x x - k_y y - k_z z)} + p_R e^{i(\omega t - k_x x - k_y y + k_z z)}, \tag{2.6}$$

where p_I is the amplitudes of the incident wave in the incident field P_I, and p_R is the amplitude of the wave reflected by the plate, which forms the reflected field P_R. The plate vibration also radiates a plane wave of magnitude p_T in the transmission field P_T, for which acoustic pressure is given by:

$$P|_{z<0} = P_T = p_T e^{i(\omega t - k_x x - k_y y - k_z z)}. \tag{2.7}$$

The structural-acoustic coupling equation states that there must be continuity of acoustic and mechanics velocities at the interface of the plate with the acoustic fields:

$$\frac{\partial P}{\partial z}\Big|_{z=0_\pm} = \rho_0 \omega^2 w, \tag{2.8}$$

where $w = W(x, y)e^{(i\omega t)}$ is the plate transverse displacement. For the sake of brevity, the factor $e^{(i\omega t)}$ will be omitted from the equations hereafter. Therefore, the coupling equations in the source and transmission side are, respectively:

$$ik_z(P_R - P_I) = \rho_0 \omega^2 w, \tag{2.9}$$

$$-ik_z P_T = \rho_0 \omega^2 w, \tag{2.10}$$

from which the following relationship between the pressure fields' amplitudes is obtained:

$$P_R = P_I - P_T. \tag{2.11}$$

By definition, the acoustic transparency τ is the ratio between transmitted and incident sound intensity, where the sound intensity can be written as:

$$I_i = |P_i \cdot v_i| = \frac{|P_i|^2}{\rho_0 c_0}, \tag{2.12}$$

where the subscript i indicates the acoustic field, which can be the incident, transmitted, or reflected field, and v_i is the local fluid velocity. Therefore, the acoustic transparency is:

$$\tau(\phi, \theta, \omega) = \left| \frac{P_T}{P_I} \right|^2. \tag{2.13}$$

The acoustic transparency for a diffuse field $\tau_d(\omega)$ is the weighted average transmission coefficient of all possible incident and azimuth angles:

$$\tau_d(\omega) = \frac{\int_0^{2\pi} \int_0^{\pi/2} \tau(\phi, \theta, \omega) \cos\theta \sin\theta d\theta d\phi}{\int_0^{2\pi} \int_0^{\pi/2} \cos\theta \sin\theta d\theta d\phi}. \tag{2.14}$$

The STL is the inverse of the acoustic transparency measured on the dB scale (Equation (2.2)). To find the relationship between the pressure field amplitudes in Equation (2.13), the constitutive equation of motion of the plate should be solved combined with Equation (2.11).

2.2.1 Analytical Approach for Infinite Plates with Different Materials

The plate constitutive equation can be formulated as a function of the plate impedance Z as

$$iZ\omega w = F = P|_{z=0^-} - P|_{z=0^+} \tag{2.15}$$
$$= (P_I - P_R) - (P_T) \tag{2.16}$$
$$= 2(P_I - P_T), \tag{2.17}$$

Combining Equations (2.10) and (2.15), it is possible to find the relation:

$$\frac{P_T}{P_I} = \left(\frac{Z \cos\theta}{2\rho_0 c_0} + 1 \right)^{-1}, \tag{2.18}$$

which is straightforwardly applied in Equation (2.13) to find the acoustic transparency for a plane wave. As Equation (2.14) does not have a close analytical solution, it is numerically evaluated to output STL for a diffuse field.

2.2.1.1 Isotropic Thin Plate

The constitutive equation of motion of an isotropic thin plate can be written as:

$$D\nabla^4 w - \omega^2 m w = q, \tag{2.19}$$

where q is the external force per unit area, D is the the isotropic plate's bending stiffness,

$$D = \frac{Eh^3}{12(1-\nu^2)}(1+i\eta), \tag{2.20}$$

E is the elastic modulus, ν is the Poisson's ratio, and η is the damping loss factor of the plate.

For the case of an infinite isotropic plate proposed by Cremer [240] and implemented in accordance to Christen et al. [241], the plate response is independent of ϕ. Therefore, one can write:

$$\nabla = \frac{\partial}{\partial x} = -ik_x = -i\frac{\omega}{c_0}\sin\theta, \tag{2.21}$$

and Equation (2.19) is rewritten as:

$$\left(D\frac{\omega^4}{c_0^4}\sin^4\theta - m\omega^2\right) w = q. \tag{2.22}$$

Thus, the radiation impedance of the finite-size plate reads as:

$$Z(\omega,\theta) = \left(1 - \frac{\omega^2 D}{mc_0^4}\sin^4\theta\right) i\omega m. \tag{2.23}$$

As observed in [239], while the first term of $Z(\omega,\theta)$ in Equation (2.23) is dominated by the plate inertia, the second term is stiffness-controlled. The natural propagating wavenumber of the thin isotropic plate is the bending wavenumber $k_p = \frac{\omega^2\rho h}{D}^{1/4}$ [239]. Therefore, its coincidence frequency occurs at

$$\omega_{coin,t}(\theta) = \frac{c_0^2}{\sin^2\theta}\sqrt{\frac{\rho h}{D}}. \tag{2.24}$$

2.2.1.2 Orthotropic Plate

An orthotropic plate has different properties in the x and y directions. Therefore, the plate response depends on the wave azimuth direction ϕ. Thus the structural impedance of an infinite orthotropic plate in the STL problem is given by [242]:

$$Z(\omega,\theta,\phi) = \left(1 - \omega^2\frac{D(\phi)}{mc_0^4}\sin^4\theta\right) i\omega\rho h, \tag{2.25}$$

where $D(\phi)$ is the heading bending stiffness of the plate and is defined as

$$\begin{aligned} D(\phi) &= D_x \cos^4\phi + 2D_{xy}\sin^2\phi\cos^2\phi + D_y\sin^4\phi \\ &= \left(E_x\cos^4\phi + 2(E_y v_{xy} + 2G_{xy})\sin^2\phi\cos^2\phi + E_y\sin^4\phi\right)\frac{h^3(1+i\eta)}{12\left(1-\frac{E_y}{E_x}v_{xy}^2\right)}. \end{aligned} \tag{2.26}$$

In Equation (2.26), the sub-indices correspond to the direction relative to each material property, and G_{xy} is the in-plane shear modulus.

The behavior of the plate-bending wave on an orthotropic plate varies according to the wave's direction. Thus, the coincidence frequency occurs at different azimuth angle ϕ, and is given by

$$\omega_{coin,o}(\phi,\theta) = \frac{c_0^2}{\sin^2\theta}\sqrt{\frac{\rho h}{D(\phi)}}. \tag{2.27}$$

Two special cases of Equation (2.27) are important in the STL analysis of orthotropic plates, namely the critical frequencies corresponding to the directions x and y, i.e., when $\phi = 0°$ and $\phi = 90°$, respectively, and are calculated as

$$\omega_{crit,x} = c_0^2\sqrt{\frac{\rho h}{D_x}} \quad \text{and} \quad \omega_{crit,y} = c_0^2\sqrt{\frac{\rho h}{D_y}}. \tag{2.28}$$

2.2.1.3 Sandwich Panels

Sandwich structures are composite structures composed of two thin and stiff plates, the skin layers, interspersed with a thick layer of lightweight material, the core layer, and are widely used due to their good stiffness-weight ratio. Sandwich material can be designed to shift the critical frequency to higher frequencies and increase the transmission loss at coincidence due to the dampening effects of the core layer.

As the core layer is thick, its transverse shear deformation must be considered. To include these effects, the constitutive equation of motion of sandwich plates becomes [243, 242]:

$$Dk_x^4 - \rho\omega^2\left(1 + \frac{Dk_x^2}{N}\right)w = q, \tag{2.29}$$

where N represents the transverse shear rigidity of the plate. Consequently, the structural impedance of the sandwich plate can be described by the following equation:

$$Z(\omega,\theta) = im\omega\left[1 + \left(\frac{D}{Nc_0^2}\sin^2\theta - \frac{D}{mc_0^4}(1+i\eta)\sin^4\theta\right)\omega^2\right]. \tag{2.30}$$

In the proposed formulation in [242], the bending stiffness and damping contribution are given by the skin layers, and the shear stiffness contribution by the core layer. Thus, for an isotropic sandwich material with a honeycomb core, the bending stiffness can be written as:

$$D = E_s h_c^2 h_s\left(1 + \frac{h_s}{h_c}\right)^2(1+i\eta), \tag{2.31}$$

and the shear stiffness is given by [243]:

$$N = G_c h_c \left(1 + \frac{h_s}{h_c}\right)^2, \tag{2.32}$$

where G_c is the shear modulus of the core. The subindices c and s relate respectively to core and skin. When the shear stiffness approaches infinity, this formulation simplifies to that of a thin plate.

The coincidence frequency for a thick isotropic sandwich plate is given by [244] is given by:

$$\omega_{coin,s}(\theta) = \frac{c_0^4 m}{D \sin^4\theta \left(1 - \frac{c_0^2 m}{N \sin^2\theta}\right)} \tag{2.33}$$

2.2.2 Correction Factor Approach for Isotropic Thin Plates

Based on the Rayleigh-integral method, Atalla et al. [245] propose to apply a correction factor to the acoustic transparency of the infinite plate τ_∞ to account for effects of the plate size, such that the acoustic transparency of a finite plate could be approximated by:

$$\tau_{fin} = (\sigma_R \cos\theta)\tau_\infty, \tag{2.34}$$

where the radiation efficiency σ_R is given by

$$\sigma_R(\theta, \phi, a, b) = \frac{\Re(Z_{fin}(\theta, \phi, a, b))}{\rho_0 c_0}. \tag{2.35}$$

The structural impedance Z_{fin} is evaluated as:

$$Z_{fin} = \frac{j\rho_0\omega}{S} \int_S \int_S e^{-i(k_x x + k_y y)} G(x, y; x', y') \times \; e^{-i(k_x x' + k_y y')} dx dy dx' dy', \tag{2.36}$$

where S designates the surface of the panel, and G is the half-space Green's function. Rhazi and Atalla [246] proposes analytical simplifications, so that σ_R simplifies into:

$$\sigma_R(\theta, \phi, a, b) = \frac{\omega b}{16 c_0} \int_{-1}^{1} \frac{a}{b} F(\theta, \mu(\phi), a, b) d\mu, \tag{2.37}$$

where $\mu = (4\phi/\pi - 1)$. This equation can be evaluated using Gaussian numerical integration. The function $F(\theta, \mu(\phi), a, b)$ and the details of its derivation are found in [246]. Although the correction factor approach is simplified, the resultant modified STL curve is coherent with experimental curves with band average [238].

2.2.3 Modal Summation Approach for Isotropic Thin Plates

An analytical solution of the fluid-structure iteration problem of a simply supported plate can also be obtained by modeling the plate displacement w in terms of a modal summation [247, 248]:

$$w(x, y, z) = \sum_{m,n} \alpha_{m,n} \varphi_{m,n}(x, y), \tag{2.38}$$

where $\sum_{m,n}$ is the short format for $\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}$, the subindices m and n indicate the mode index, $\alpha_{m,n}$ is the coefficient of the contribution of each mode for the displacement field, and $\varphi_{m,n}(x,y)$ (Equation (2.39)) is the modal function that satisfies the simply supported boundary conditions (Equation (2.40)).

$$\varphi_{m,n}(x,y) = \sin\left(\frac{m\pi x}{a}\right)\sin\left(\frac{n\pi y}{b}\right). \tag{2.39}$$

$$\begin{aligned} w &= \frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2} = 0, \quad \text{for } x = \pm a/2, \\ w &= \frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2} = 0, \quad \text{for } y = \pm b/2. \end{aligned} \tag{2.40}$$

Using the 2D Fourier Transform, each pressure field P_i can also be described as a modal summation:

$$P_i(x,y,z) = \sum_{m,n} p_{i_{m,n}}\varphi_{m,n}(x,y)e^{\pm k_z z}, \tag{2.41}$$

where the signal of the exponential function depends on the propagation direction of the plane wave in the field, and in which the coefficient of the contribution of the pressure field $p_{i_{m,n}}$ reads as:

$$p_{i_{m,n}} = \frac{4}{ab}\int_0^b\int_0^a p_i e^{-i(k_x x + k_y y)}\varphi_{m,n}(x,y)dxdy. \tag{2.42}$$

Substituting Equations (2.38), (2.41) and (2.42) into Equation (2.19) and applying the weighted residuals method:

$$\begin{aligned} \int_0^b\int_0^a \Bigg[D\nabla^4\sum_{m,n}\alpha_{m,n}\varphi_{m,n} - \omega^2 m\sum_{m,n}\alpha_{m,n}\varphi_{m,n} - \\ \sum_{m,n} 2\left(p_{I\,m,n} - p_{T\,m,n}\right)\varphi_{m,n}\Bigg]\varphi_{m,n}dxdy = 0. \end{aligned} \tag{2.43}$$

As the modal functions have orthogonal properties $\{\varphi_{j,k}\}^T M\{\varphi_{h,g}\} = 0$ for $j \neq h$ or $k \neq g$, Equation (2.43) reads as

$$\alpha_{m,n}D\frac{\int_0^b\int_0^a\left[\nabla^4\varphi_{m,n}\right]\varphi_{m,n}dxdy}{\int_0^b\int_0^a\varphi_{m,n}dxdy} - \alpha_{m,n}\omega^2 m - 2\left(p_{I\,m,n} - p_{T\,m,n}\right) = 0 \tag{2.44}$$

if inter-modal coupling is neglected. The integrals on the first term can be analytically evaluated, and the relation from Equation (2.9) is used, resulting in:

$$\left[\omega_{m,n}^2 - \omega^2 + 2\frac{i\omega\rho_0 c_0}{m\cos\theta}\right]\alpha_{m,n} = \frac{2}{m}p_{I\,m,n}, \tag{2.45}$$

where $\omega_{m,n}$ are the natural frequencies of the simply supported plate, and $p_{I\,m,n}$ can be analytically evaluated by solving the integrals from Equation (2.42). Finally, the solution of the system of equation in Equation (2.45) provides the coefficients of modal participation $\alpha_{m,n}$ and, consequently, the displacement w (Equation (2.38)). The coefficients $p_{T\,m,n}$ are evaluated with Equation (2.10), and the acoustic transparency is obtained by:

$$\tau(\theta,\phi) = \frac{\sum_{m,n} |p_{T\,m,n}|^2}{\sum_{m,n} |p_{I\,m,n}|^2}. \tag{2.46}$$

2.2.4 Finite Element Method Approach for Isotropic Thin Plates

The numerical solution of the STL problem was implemented in COMSOL Multiphysics® software [249] using FEM. The plate was modeled by homogeneous shell elements with simply supported boundary conditions, while the receiver acoustic field was modeled by 3D finite elements with perfectly matched layers in its external interfaces to simulate the free-field condition. The FEM model does not include the acoustic field on the source side, but an equivalent pressure load is applied to the plate to account for the fluid load. The fluid pressure is modeled as the summation of 200 components in random directions to simulate the diffuse field. The frequency domain formulation is used to solve the coupled plate constitutive equation and acoustic wave equation. The incident and transmitted power are then calculated as the integral of the incident and transmitted intensity on the plate surface, respectively. Details on the theory and numerical implementation can be found in [250].

2.3 Benchmarking ML Surrogates for Modelling STL in Plates

2.3.1 Infinite Plates with Material Variability

This section compares the accuracy of NN, GPR, RF, and GBT-based surrogate models with and without PGF in predicting the STL for infinite plates with different material properties such as isotropic, orthotropic, and sandwich structures. To generate the datasets, LHS was used to sample up to 500 input-output pairs from the physics models described in Section 2.2.1, considering the diffuse field as the incident acoustic field. The training of the ML models was performed using 80% of the dataset, while the remaining 20% was used as a test dataset to evaluate the surrogate performance using the Root Mean Square Error (RMSE) obtained with five-fold cross-validation as the evaluation metric. The computations in this and following chapters were conducted on a desktop with a hexa-core 3.1 GHz processor and 32 GB of RAM. *Python* was used as the primary programming language, and *Matlab* functions provided the analytical STL solutions.

The selected design spaces have a broad variation range to train surrogates suitable for design exploration. For the isotropic thin plate case, the design space

Table 2.1: Design space considered for the isotropic thin plate analyses.

Inputs	$\boldsymbol{h}$	$\boldsymbol{E}$	$\boldsymbol{\nu}$	$\boldsymbol{\rho}$	$\boldsymbol{\eta}$
Units	mm	GPa	-	kg/m^3	%
Lower Bound	7	55	0.20	2300	0.1
Upper Bound	8	85	0.35	3300	1

Table 2.2: Design space considered for the orthotropic plate analyses.

Inputs	$\boldsymbol{h}$	$\boldsymbol{E_x}$	$\boldsymbol{\nu}$	$\boldsymbol{\rho}$	$\boldsymbol{\eta}$	$\boldsymbol{E_y}$	$\boldsymbol{G_{xy}}$
Units	mm	GPa	-	kg/m^3	%	GPa	GPa
Lower Bound	3.25	190	0.25	1300	0.1	5	50
Upper Bound	4	260	0.35	1800	1	10	65

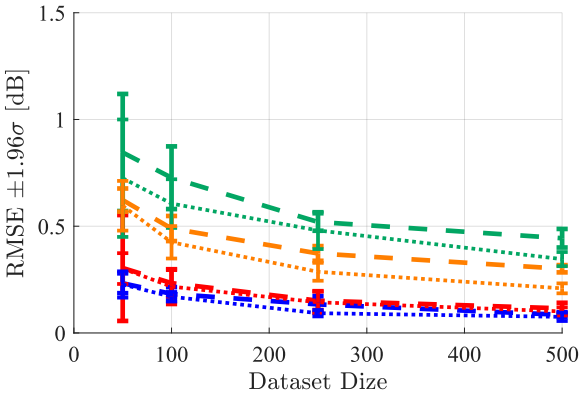
Table 2.3: Design space considered for the sandwich plate analyses.

Inputs	$\boldsymbol{h_s}$	$\boldsymbol{E_s}$	$\boldsymbol{G_c}$	$\boldsymbol{\rho_s}$	$\boldsymbol{h_c}$	$\boldsymbol{\rho_c}$	$\boldsymbol{\eta}$
Units	mm	GPa	MPa	kg/m^3	mm	kg/m^3	%
Lower Bound	1	55	0.10	2300	15	60	0.1
Upper Bound	2	85	0.70	3300	25	120	1

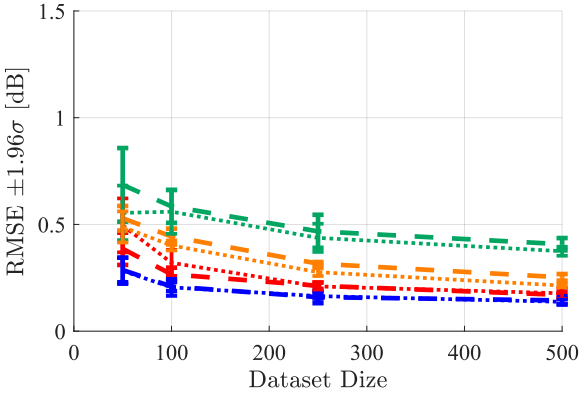
encompassed a wide range of aluminum grades and is presented in Table 2.1. The investigations of the STL for an orthotropic plate considered a unidirectional fiber-reinforced composite plate in which carbon fibers are oriented along the x axis with resin as matrix material, resulting in different Young modulus in the in-plane directions. Table 2.2 provides the orthotropic plate properties ranges. The sandwich plates considered consisted of thin aluminum skin layers and a thick honeycomb core, and the design space for this case is provided in Table 2.3.

Figure 2.4 illustrates the results of the benchmarking study, showing the RMSE of the ML-based surrogate predictions without the use of PGFs, represented by dashed lines (**- - -**). All four ML algorithms achieve satisfactory accuracy in predicting the STL, even with relatively small datasets. However, the NN and GPR models outperform the RF and GBT models in accuracy.

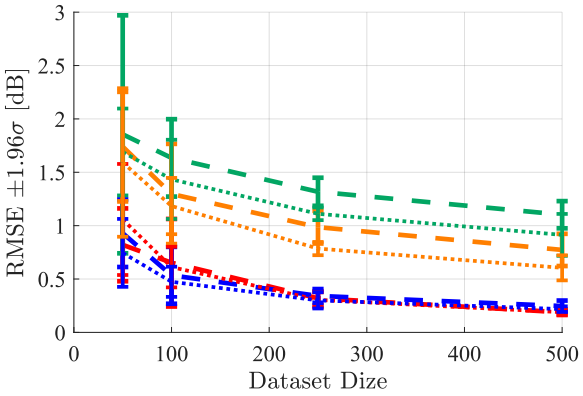
The prediction errors for isotropic thin plates are smaller than those for the sandwich and orthotropic plates. This disparity can be attributed to the lower dimensionality of the isotropic problem, which involves fewer variables. Additionally, the phenomena being predicted for the sandwich and orthotropic cases are more complex, with coincidence phenomena occurring in multiple directions for the orthotropic case and transverse shear phenomena considered in the sandwich case. Despite the more significant errors in the sandwich case, the NN and GPR-based surrogates trained with 500 supporting points maintain an average RMSE of approximately $0.25dB$ across all frequencies, which is satisfactory.



(a) Isotropic thin plates.



(b) Orthotropic thin plates.



(c) Sandwich plates.

Figure 2.4: Benchmark of ML-based surrogate for different material properties regarding the accuracy in relation to the dataset size: — Neural Network; — Gaussian Process Regressor; — Random Forest; — Gradient Boosting Trees; without - - - and with ····· Physics-Guided Features.

Next, we investigate the influence of Physics-Guided Featuress (PGFs) on ML prediction performance. The selection of PGFs should be based on the underlying physics of the problem. For the isotropic thin plates, two PGFs were considered: mass density m and the real part of the bending stiffness D_R from Equation (2.20). Similarly, for the orthotropic plates, the PGFs included mass density m and the coefficients of the heading bending stiffness, D_x, D_x, and D_x from Equation 2.26. In the case of sandwich plates, in addition to mass density and the real part of the bending stiffness from Equation (2.31), the shear stiffness given by Equation (2.32) and the geometric term

$$g = \left(1 + \frac{h_s}{h_c}\right)^2, \tag{2.47}$$

were included as PGFs. According to STL theory, the mass density m controls the STL in the mass region, the bending stiffness has a significant influence on the coincidence region, and the shear stiffness influences the coincidence frequency [239, 243].

The results of the ML models' predictions with PGFs are depicted by the dotted lines (·····) in Figure 2.4. A comparison of the prediction errors with and without PGFs reveals that including PGFs improves ML accuracy, particularly for RF and GBT algorithms. The results with and without PGFs show similar performance for NN and GPR models.

2.3.2 Isotropic Plate Modeled with Different Complexities

In the previous section, we established that ML-based surrogate models provide accurate predictions for the STL analysis of infinite plates. This section aims to extend the analysis by comparing the performance of NN, GPR, RF, and GBT models in terms of accuracy and training time when predicting the STL of both infinite and finite thin isotropic plates with increasing complexity. As complexity rises, the non-smoothness degree of the analysis increases, allowing us to assess the ML models' robustness in the face of the challenges that arise in SD&V.

The benchmarking study covers the four STL physical modeling approaches described in Section 2.2. This analysis allows us to evaluate the robustness of ML surrogates' accuracy in capturing non-smooth behavior. Additionally, we discuss the elapsed time required for training the ML models and examine the influence of the selected design space and the PGF. The database used in this section has 2000 supporting points sampled with LHS from the design space of Table 2.4. The design space encompasses a wide range of aluminum grades and aluminum matrix composites.

Figure 2.5 shows the RMSE of each surrogate's prediction for all six STL models at different dataset sizes. The results reveal that the surrogates' performance improves considerably when datasets of up to 1000 samples are used. However, the improvement is reduced for large dataset sizes, suggesting that sampling additional points may not justify the increase in accuracy.

Table 2.4: Design space with the set of interest used to sample supporting points.

	ρ $\mathrm{kg/m^3}$	E GPa	ν -	η %	h mm	a m	b m
min	2000	60	0.25	0.1	5	0.3	0.3
max	3000	150	0.35	2.0	7	0.6	0.6

Among the ML models, the NN-based surrogates consistently achieve better accuracy when big datasets are used. This can be justified because NN can produce a better fit on non-smooth functions than other ML approaches [251]. The GPR model, widely used for constructing surrogates, performs well for the infinite plate and correction factor models and struggles with highly non-smooth models due to its reliance on spatial correlation of input features [252].

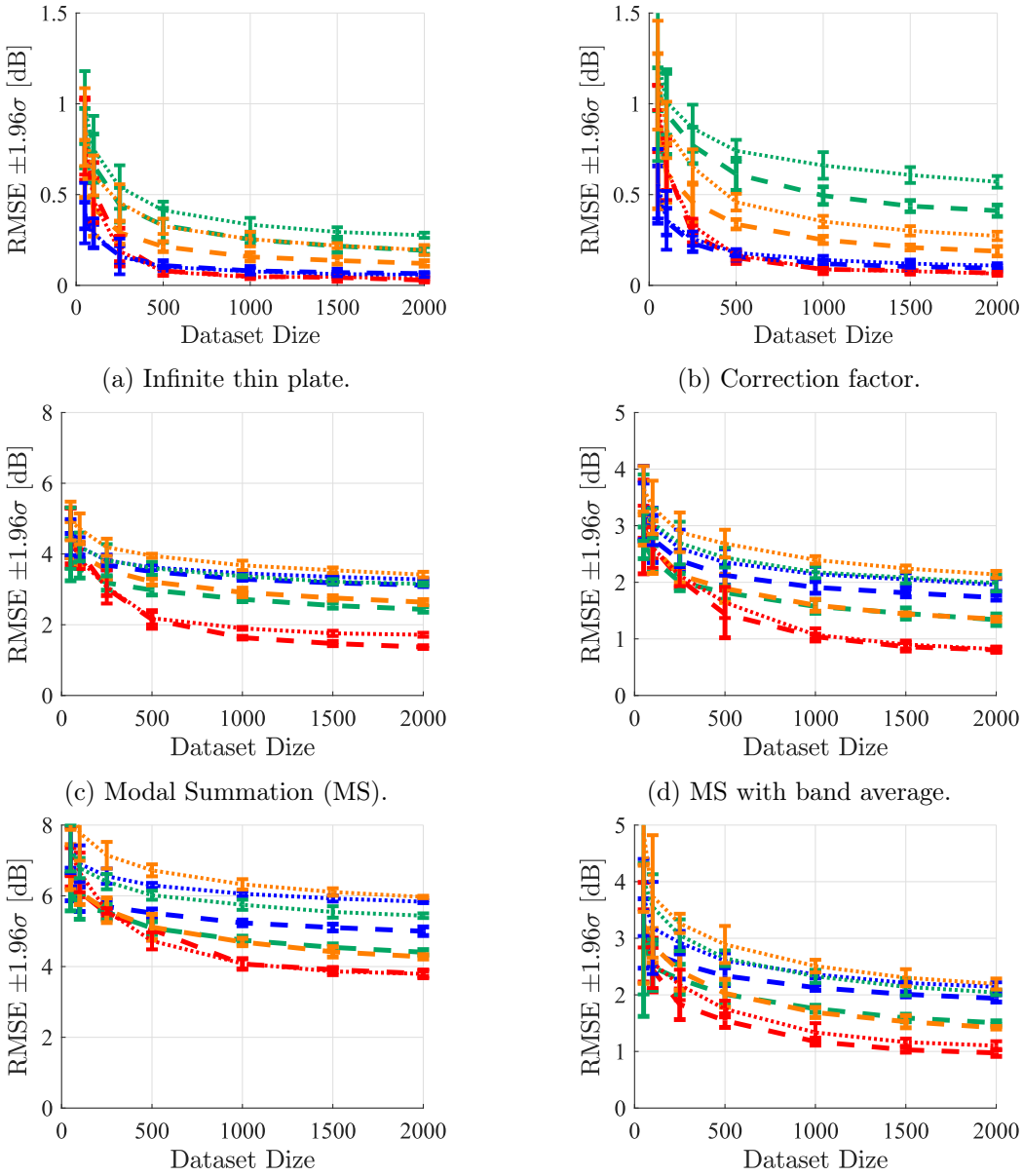
Furthermore, we analyze the influence of PGFs on the surrogate models' performance. The mass density m and the real part of the bending stiffness D_R are included as PGFs for all STL approaches. For the cases with finite plate, the non-smoothness of the modal behavior poses difficulties for the ML models. Therefore, we also include the resonance coefficient term

$$R = \frac{D}{m(a^4 b^4)} \tag{2.48}$$

as a PGF in the surrogate model for the MS and FEM analyses. The resonant term R is the coefficient of the first term of Equation (2.45) and relates with the simply supported plate natural frequencies and, thus, its modal behavior.

As observed in Figure 2.5, the addition of PGFs consistently produces more accurate models. The improvement obtained due to the PGFs in the NN-based surrogate performance is not as significant as for the other ML models, highlighting NN's ability to learn complex representations of the data even without expert guidance. Except for the FEM model, the NN-based surrogates of all STL models achieved RMSE below 3 dB, which are satisfactory results since measurements errors in STL experiments usually also go up to 3 dB. The RMSEs for the FEM model are higher due to the overall more complex simulation and stochastic excitation, which renders the data less reliable.

In addition to accuracy, training time is an important consideration when selecting a surrogate model. Figure 2.6 presents the training time for each ML model. RF and GBT surrogates are significantly faster to train than the other two methods since the training of decision-tree-based models scales loglinearly with the dataset size [253]. As no efficient multi-output implementation of GBT is available, it was required to fit one GBT model for each input frequency, increasing the overall training time compared to the RF model. On the other hand, GPR is computationally expensive to train for large datasets as its operations scale as $\mathcal{O}(N^3)$ [254]. It is worth noting that GPR, GBT, and RF provide sensitivity analysis as a by-product, making them more interpretable than NN. However, NN and GPR exhibit better extrapolation capabilities compared to decision-tree-based methods.



(a) Infinite thin plate.

(b) Correction factor.

(c) Modal Summation (MS).

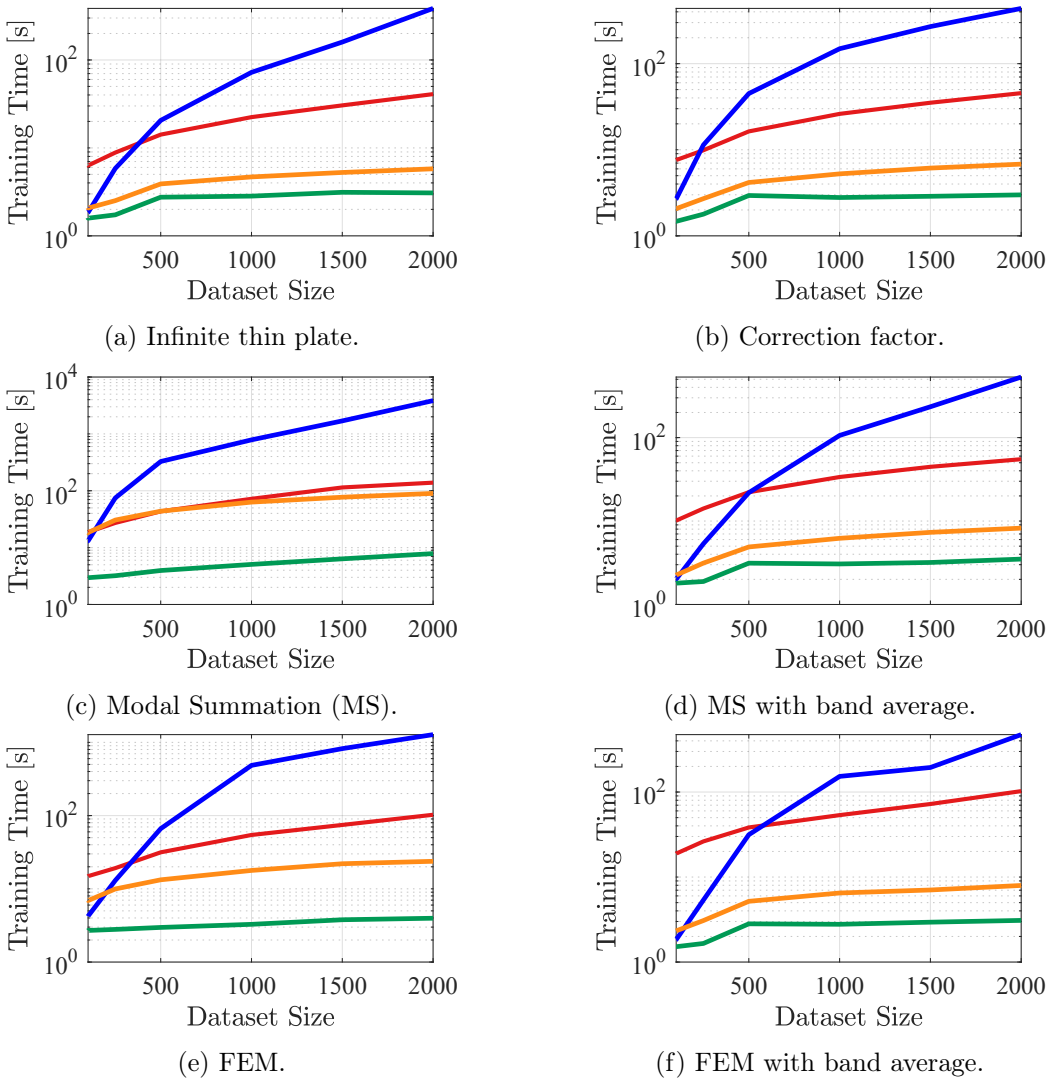
(d) MS with band average.

(e) FEM.

(f) FEM with band average.

Figure 2.5: Benchmark of ML-based surrogate regarding the accuracy in relation to the dataset size for different model complexities. — Neural Network; — Gaussian Process Regressor; — Random Forest; — Gradient Boosting Trees; without ····· and with - - - Physics-Guided Features.

To visually assess the accuracy of the predicted STL curves, Figure 2.7 shows an example of the predicted STL curve for each STL model using the most accurate surrogate which is the NN model with PGFs trained over 2000 supporting points. The predicted curves are remarkably similar to the simulated ones, indicating that NN-



(a) Infinite thin plate.

(b) Correction factor.

(c) Modal Summation (MS).

(d) MS with band average.

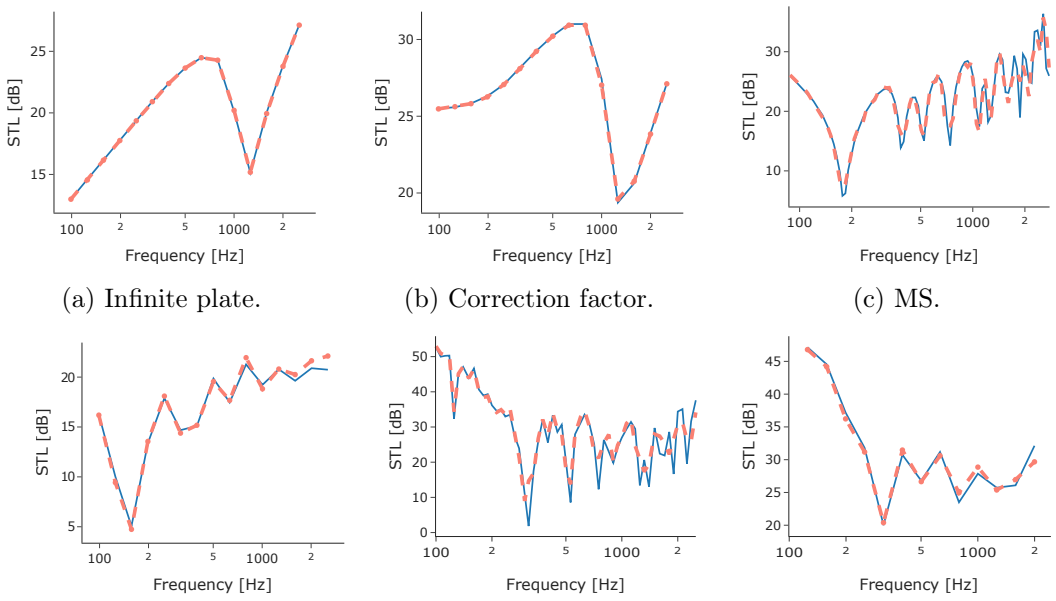
(e) FEM.

(f) FEM with band average.

Figure 2.6: Benchmarking of ML-based surrogate regarding the training time in relation to the dataset size: — Neural Network; — Gaussian Process Regressor; — Random Forest; — Gradient Boosting Trees.

based surrogates can capture the non-smooth behavior of vibroacoustic problems. Figure 2.8 provides a deeper look into the NN-based surrogate model errors for each STL model, additionally indicating the Mean Absolute Error (MAE) and the Mean Maximum Error (MME). For instance, the MME indicates high localized errors in MS and FEM models without band average. Therefore, although these surrogates adequately predict the overall STL behavior, they may considerably mispredict local maxima and minima of a rough response signal.

Overall, the developed surrogate models perform satisfactorily for most studied cases. Even highly discontinuous and rough functions could be adequately predicted. Although the ML algorithms' performance is problem-specific, they usually



(a) Infinite plate. (b) Correction factor. (c) MS.

(d) MS (band averaged). (e) FEM. (f) FEM (band averaged).

Figure 2.7: Example of one predicted STL curve for each STL model using the neural network (NN)-based surrogate with PGF: — simulated; - - - predicted.

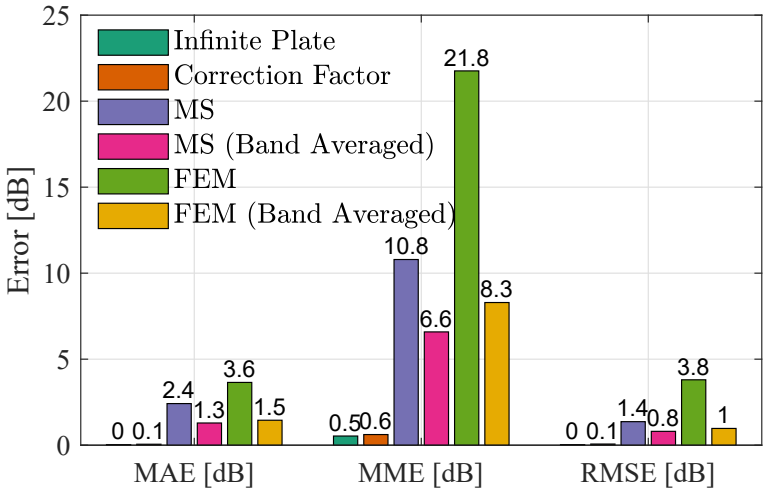


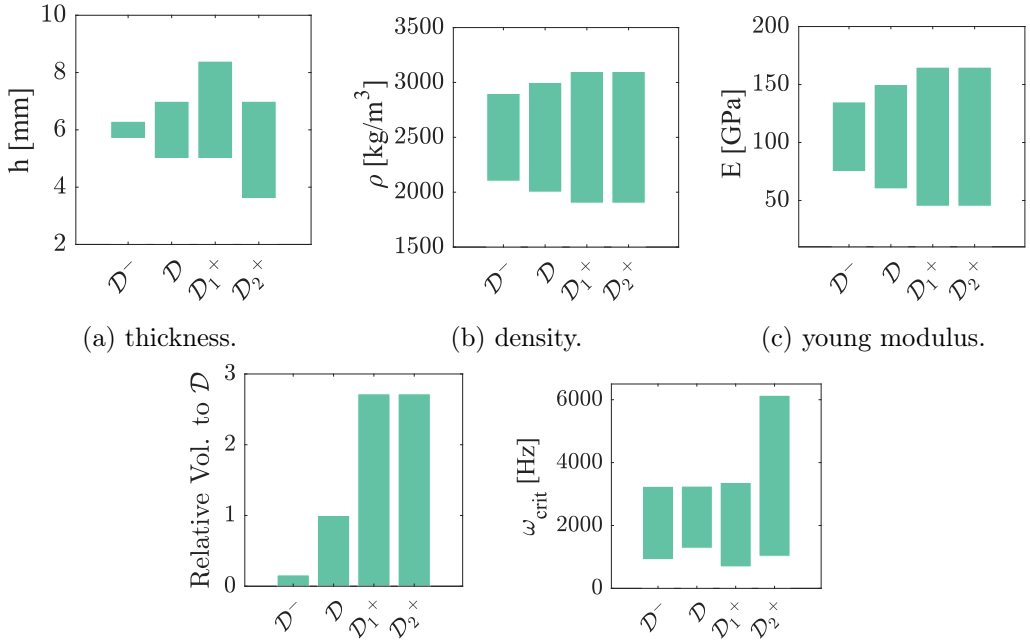
Figure 2.8: Mean absolute error (MAE), mean maximum error (MME) per STL design, and RMSE of the NN-based surrogate model with physics-guided features.

perform well in related tasks; therefore, the benchmarking provides insights into the suitability of the ML methods to model vibroacoustic problems with non-smooth behavior.

2.3.3 Influence of Selected Design Space on the ML Prediction Performance

To further explore the performance of NN-based surrogates, an investigation was conducted to evaluate their accuracy regarding the set of interest $\hat{\mathcal{X}}$. Four design spaces were considered: the standard design space $\mathcal{D}$ from Table 2.4, the smallest design space $\mathcal{D}^{-}$, and the two largest design spaces $\mathcal{D}_1^{+}$ and $\mathcal{D}_2^{+}$. The design spaces $\mathcal{D}_1^{+}$ and $\mathcal{D}_2^{+}$ have the same size but are located in different places in the domain $\mathcal{X}$, as illustrated in Figure 2.9. The range of damping, Poisson's ratio, and the width and length of the plates remained consistent across all design spaces, following the values specified in Table 2.4. Figure 2.9 shows the resulting range of critical frequencies.

For each design space, 500 supporting points were sampled using LHS for the STL models of the infinite plate, correction factor, and MS with and without one-third octave band average.

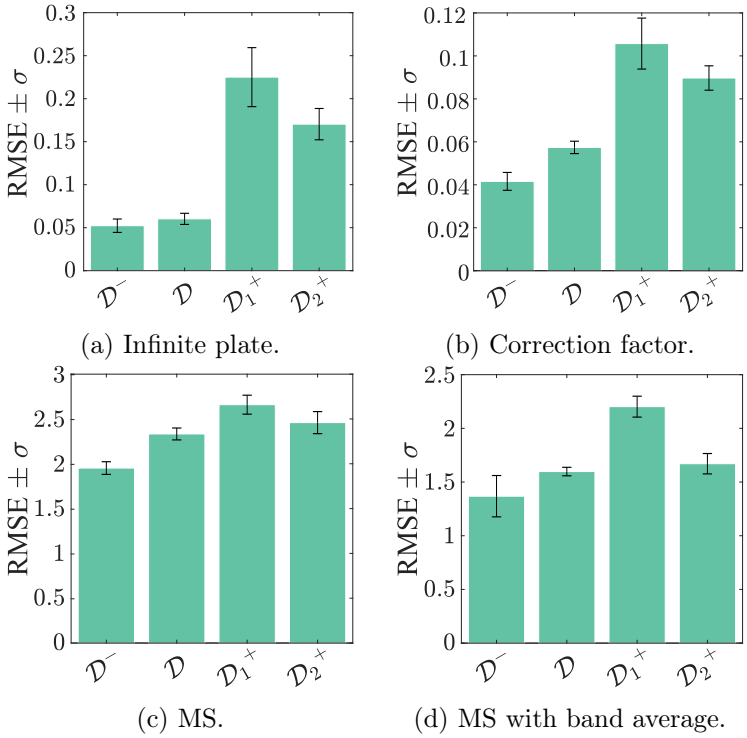


(a) thickness. (b) density. (c) young modulus.

(d) design space volume. (e) critical frequency.

Figure 2.9: Range of the input variables used for each design space considered (**a**–**c**), the volume of the design spaces relative to the volume of the standard design space $\mathcal{D}$ (**d**) and the range of critical frequencies resulting from each design space (**e**).

Figure 2.10 presents the accuracy obtained with an NN-based surrogate model trained with physics-guided features for each set of interest and the STL method. The results indicate that the surrogate error tends to increase with the design space size, especially for the infinite plate and correction factor models. However, for the MS model without band average, the difference in the surrogates' accuracy is not relevant.



(a) Infinite plate. (b) Correction factor.

(c) MS. (d) MS with band average.

Figure 2.10: NN-based surrogate accuracy for the design spaces $\mathcal{D}$, $\mathcal{D}^-$, $\mathcal{D}_1^+$ and $\mathcal{D}_2^+$ when modeling the STL problem with different approaches:

Furthermore, it can be observed that a considerable difference in accuracy is obtained for design spaces $\mathcal{D}_1^+$ and $\mathcal{D}_2^+$, although they have the same size. This is because the samples from $\mathcal{D}_2^+$ have, on average, a simpler behavior than the samples from $\mathcal{D}_1^+$, once many of them present critical frequency higher than 2500 Hz, which is above the frequency range modeled by the surrogates. Thus, the location of the design space can have more of an effect on surrogate accuracy than its size.

2.4 Conclusions and Discussion

This chapter focused on investigating the use of ML-based surrogate modeling in the context of SD&V systems, specifically for STL analyses. The surrogates were evaluated for different material properties and at varying complexity levels, including analytical and numerical approaches.

The ML-based surrogates demonstrated their ability to achieve satisfactory accuracy for STL predictions, even in the presence of highly non-smooth behavior. However, as the complexity and irregularity of the system response increase, these surrogates require additional support points to maintain accuracy.

Among the four ML methods studied, NN and GPR surrogates outperformed RF and GBT-based surrogates for the infinite STL analyses of thin isotropic plates, orthotropic plates, and sandwich plates. However, when considering more complex

scenarios, such as STL of finite plates, NN models outperformed GPR, RF, and GBT, while the Decision Tree-based algorithms (RF and GBT) exhibited notably faster training times.

Furthermore, including domain knowledge through PGFs generally improved the surrogate performance, highlighting the importance of incorporating physical understanding into the ML modeling process. The influence of PGF was less significant for simpler problems and NN-based models.

The surrogate errors are also shown to be proportional to the design space size for the surrogate with less complex STL models but do not change significantly with the design space size for the STL with high non-smooth responses. On the other hand, the location of the design space in the input domain has a relevant impact on the accuracy in all cases.

Although the results were generated for the case of STL, some conclusions can be generalized to guide the implementation of accurate surrogates for other SD&V problems with similar behavior. The capability of the ML-based surrogate, especially the NN-driven, to predict non-smooth behavior observed in STL analyses indicates their potential effectiveness in other SD&V analyses.

Therefore, this chapter presents good practices and methods to create ML-based surrogate models for SD&V problems with improved accuracy, and physical guidance, increasing confidence in the model. The vibroacoustic surrogates can serve as informed and efficient decision-making tools, particularly suitable for constructing the DT Prototype proposed in this thesis.

Chapter 3

On the Interpretability of Digital Twins Using Global Sensitivity Analysis

Although ML-based DT have shown promising results in accurately speeding up vibroacoustic simulations, the lack of interpretability and physics foundation inhibit the widespread usage of these models by the scientific community. Therefore, this chapter aims to enhance the interpretability of these ML-based DTs. To achieve this goal, we employ GSA and PGF to improve the interpretability and physical consistency of ML-based simulations for the STL problem across various plate materials, model complexities, and incident fields. Computationally efficient sensitivity analysis is obtained via the MDI, which is the byproduct of the training of the RF surrogate models. The resulting sensitivity indices were similar to the traditional Sobol indices and more accurate than FAST for small datasets. Moreover, introducing basic expert knowledge into the ML inputs helped interpret the physical meaning of sensitivity indices throughout the frequency spectrum. The GSA also assists in dimension reduction and, thus, surrogate accuracy.

The authors have previously published a part of this chapter's content, referring to [107] in the Applied Sciences journal, as well as [234] presented at the International Conference on Noise and Vibration Engineering.

Contents

Introduction

Recently ML has enhanced SD&V analyses by, for example, detecting and predicting failures with high accuracy [255] and speeding up costly simulations [135]. These methods, however, are classified as "black-box", meaning that the causal relationship between the model's inputs and outputs is not explicitly determined. As understanding the underlying physics of the model is crucial in product design to support decision-making and to advance innovation, the adoption of ML in SD&V has been lagging. This chapter addresses this limitation by employing SA and PGF to enhance the interpretability of ML-based surrogates for STL.

SA assesses the importance of each input parameter to determine the output variability of a system [256] and, thus, it is widely used to improve the interpretability of models and phenomena. Local SA, commonly used in the SD&V domain [257–259], assesses the effects of individually varying the inputs at a given point in the design space. On the other hand, GSA methods, such as FAST [260] and Sobol's method [261], provide sensitivities indices that can capture interaction effects and nonlinearities and are valid for a broad input variation range [256]. The drawback of GSA is that it usually demands a specific sampling method and numerous evaluations [262]. In vibroacoustic studies, GSA has been applied to investigate panel sound absorption levels [263], vehicles' interior noise [264], and STL [242, 265].

In the context of ML model interpretability, both classic SA and ML-specific techniques are utilized [266, 267]. Mean Decrease in Accuracy (MDA), also known as permutation feature importance, is an ML-specific technique that measures global first-order sensitivities of ML models by assessing the impact of randomly permuting feature values on model accuracy. Additionally, specific ML algorithms incorporate sensitivity indices as a post-processing step during training [268, 269]. An example is MDI, a by-product of RF that provides sensitivity indices similar to Sobol's total sensitivity indices [270, 86].

This chapter contributes to developing interpretable ML models by applying classical variance-based GSA and MDI together with PGF. The methodology is evaluated using various STL cases, and the results are compared with the STL sensitivity analyses in [242, 271]. The presented methodology applies to other engineering problems and experimental setups, enabling system identification and scientific discoveries. In summary, the main contributions of this chapter are:

1. Comparative analysis of MDI indices with Sobol and FAST indices regarding their formulation, accuracy, flexibility, and time elapsed.
2. Demonstration of how MDI and Sobol, combined with surrogates, enhance interpretability and understanding of physical phenomena with negligible additional computational cost.
3. Demonstration of the methodology's effectiveness through a comprehensive analysis of various model approaches, incident fields, and material properties in the context of the STL problem.
4. Showcase the effectiveness of PGF in enhancing surrogate accuracy and interpretability.

5. Utilization of GSA for screening, reducing problem dimension, and improving surrogate accuracy.

This chapter is structured as follows: In Section 3.1, we introduce variance- and ML-based GSA theory, including Sobol's, FAST methods (Section 3.1.1), and MDI (Section 3.1.2). Section 3.2 presents GSA results for STL analyses. A comparison between FAST and MDI is in Section 3.2.1, while MDI sensitivity results and PGF influence are discussed in Sections 3.2.2 and 3.2.4, respectively, for STL with different materials and modelling approaches. The effects of dimension reduction via GSA are demonstrated in Section 3.2.4. Conclusions appear in Section 3.3.

3.1 Global Sensitivity Analysis Methodology

3.1.1 Variance-Based Sensitivity Analysis

In the variance-based SA, the importance of each input and input interaction is assessed by their contribution to the output variance. Sobol's decomposition of variance states that the total variance of the output of a model $Y = f(\mathbf{x}) = f(x_1, x_2, \ldots, x_D)$ can be represented as the sum of the individual effects of each input x_i, denoted as V_i, and the higher-order effects from the interactions between a set of inputs as long as some conditions are met [261, 272]. That is:

$$Var[Y] = V = \sum_{i=1}^{D} V_i + \sum_{i=1}^{D} \sum_{j>i}^{D} V_{ij} + \cdots + V_{1,2,\ldots,D}. \tag{3.1}$$

Following [261, 272], the first-order Sobol sensitivity indices or main effect indices of the input variable x_i can be expressed in terms of conditional expected values as

$$S_i = \frac{V_i}{V} = \frac{E_{\mathbf{x}_{\sim i}}\left(Y \mid x_i\right)}{V}, \tag{3.2}$$

where $\mathbf{x}_{\sim i}$ denotes the set of all variables but x_i and the numerator of S_i is equivalent to the expected average reduction of the output variance when x_i is fixed [273]. The total contribution of the variable x_i to the variance output, including the effects caused by its interactions with other variables, is measured by the total sensitivity index [256]:

$$S_{T_i} = 1 - \frac{Var_{\mathbf{x}_{\sim i}}(E_{x_i}(Y|\mathbf{x}_{\sim i}))}{V}. \tag{3.3}$$

The total sensitivity indices can be interpreted as the remaining variance when all terms but x_i are fixed [274]. The sum of the total sensitivity indices is equal to or greater than one. When equal to one, the function is additive, i.e., only the first-order terms contribute to the variance, and there is no effect of the interaction between any parameter, what read as $S_{T_i} = S_i$.

Evaluating variance-based sensitivity indices involves sampling and uncertainty quantification stages usually based on Monte Carlo or quasi-Monte Carlo integral methods, which are computationally intensive. A popular and less costly technique

to evaluate variance-based sensitivity indices is the FAST, developed by Cukier et al. [260] to evaluate the main effect indices. The FAST method was extended by Saltelli et al. [275] to include the computation of the total effects and is now integrated with Sobol's indices. FAST uses a particular period sampling approach to define the multiple Fourier series expansion approximating the function $f(\mathbf{x})$. As the Fourier series expansion can be decomposed as an Analysis Of Variance (ANOVA)-like decomposition, the components of the model variance can be modeled in terms of the Fourier coefficients [274]. The FAST and Sobol methods assume uncorrelated and independent variables. Otherwise, they lead to biased results if specific methodologies for correlated and dependent variables are not applied [267].

3.1.2 Mean Decrease in Impurity

Besides being efficient and able to approximate complex models, RFs are relatively interpretable ML models. They provide different ways to access feature importances that relate to sensitivity indices. One is the MDI approach, a by-product of RF training. The principle of MDI is that the contribution of each input feature to reduce the impurity error is a measure of its importance. Given an RF with Φ_κ trees, $\kappa = 1, \ldots, T$ and following the notations at Section 1.2.2.4, the MDI importance can be defined as [79, 86]:

$$\mathrm{MDI}(q_i) = \frac{1}{T} \sum_{\kappa=1}^{T} \sum_{n \in \Phi_\kappa} \mathbb{1}(q_{i_n} = q_i) \frac{L}{L_{tot}} \Delta G_n(q_{i_n}, t_n), \tag{3.4}$$

where $\mathbb{1}(q_{i_n} = q_i)$ is one if $q_{i_n} = q_i$ and zero otherwise. The impurity $G_n(q_{i_n}, t_n)$ is defined according to Equation (1.3). The term $\Delta G_n(q_{i_n}, t_n)$ denotes the change in impurity at node $n \in \Phi_\kappa$. This change is weighted based on the fraction of samples at this node in relation to the total training size, denoted as $\frac{L}{L_{tot}}$. Therefore, similarly to how the total Sobol indices can be interpreted as the effect of one input variable in the output variance decrease, the MDI indices are the effect of the input feature in the RF impurity decrease. Indeed, Scornet [276] proved that, for the case of additive functions with independent variables, the MDI of a decision tree provides a decomposition of the output variance. In the presence of interactions, consistent MDI can be obtained via RF, i.e., by averaging the MDI of several decision trees [276]. Scornet also showed that the MDI is biased towards positively correlated variables.

MDA is also often applied to evaluate the SA of RF models. Jaxa-Rozen and Kwakkel [270] compared MDI and MDA indices with Sobol and Moris metrics for different study cases and highlighted that both MDI and MDA meet the criteria of an *"ideal"* sensitivity metric, i.e., *"suitable for global sampling projects, independent of model structure, relatively easy to implement numerically, stable across all sample sizes and bootstrap resamples and applicable with generic input sampling and heterogeneous input types"*. Nonetheless, MDI and MDA do not provide the direct effects of inputs on output variance, but their relative importances. The direct ef-

fects can be assessed by applying traditional SA approaches to the fitted RF-based surrogate.

3.2 Results and Discussions

3.2.1 Comparison of Variance-Based and Mean Decrease in Impurity Sensitivity Analyses

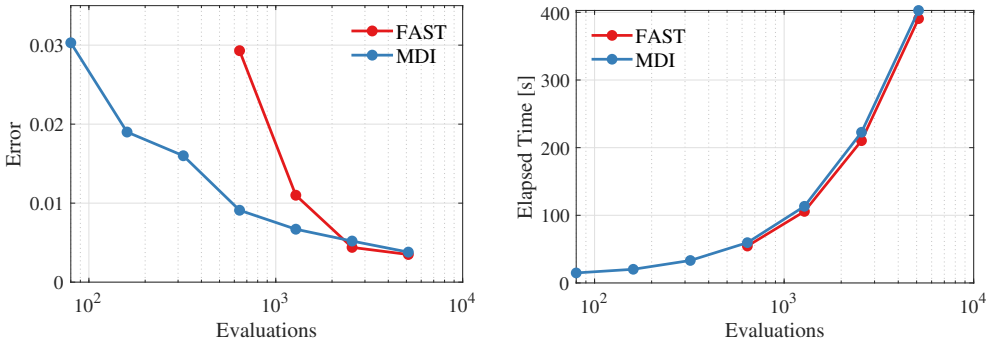
In this section, the total importances obtained with the variance-based SA using Sobol and FAST are compared with the MDI results regarding accuracy and computing time. The *Salib* library [277] was used to obtain the sensitivity indices with Sobol and FAST approaches, while the RF models were instantiated and trained using *sklearn*. One RF with 100 trees is trained for each of the N_ω outputs and 15% of the database is used to assess the RFs accuracy. The same RF architecture is used in the remaining sections of this chapter. The following comparison was made for the STL of isotropic infinite thin plates impinged by a diffuse field. The five-dimensional design space considered is presented in Table 2.1.

The MDI indices are comparable with the variance-based total effect indices S_T but on different scales, as the MDI indices sum up to one, while $\sum_i S_{T_i} >= 1$. Thus, to enable a prompt comparison between MDI indices and S_T, the importances will be scaled as $\bar{S}_{T_i} = \frac{S_{T_i}}{\sum_i S_{T_i}}$. MDI and FAST accuracies were evaluated having as a reference model R the Sobol analyses with 24576 evaluations, which was determined after a convergence test. The accuracy error of the sensitivity indices from analysis A in relation to the reference R is evaluated by

$$\varepsilon(A,R) = \frac{1}{N_\omega D}\sum_{\omega}^{N_\omega}\sum_{i}^{D} |\bar{S}^A_{T_i} - \bar{S}^R_{T_i}|. \tag{3.5}$$

Figure 3.1 shows the comparison between the performance of FAST and RF. The MDI indices are more accurate with few evaluations, while the FAST method is slightly more accurate for large databases. The elapsed time for the same number of samples is similar for both methods since sampling is the most time-consuming stage of both approaches. Recently Chai et al. [271] showed good agreement between FAST and MDA importances obtained with RF-based surrogates for an STL analysis. However, the approach did not prove advantageous due to the extra cost of MDA and its lack of mathematical foundation.

The scaled total sensitivity importances calculated with the three approaches are shown in Figure 3.2 for visual comparison. MDI and FAST importances were computed using 1280 evaluations, being that the RMSE of the surrogate evaluated in the test dataset was 0.40 dB. Figure 3.1a and Figure 3.2 indicate that there is a good agreement among the sensitivity indices calculated by all approaches and that the MDI indices obtained with RF-based surrogates are suitable indicators of input importance when few samples are available. Besides that, the RF can be trained based on a generic sampling strategy, while FAST requires a particular sampling scheme and can not reuse previous model evaluations.



(a) Sensitivity indice error ε.

(b) Elapsed time.

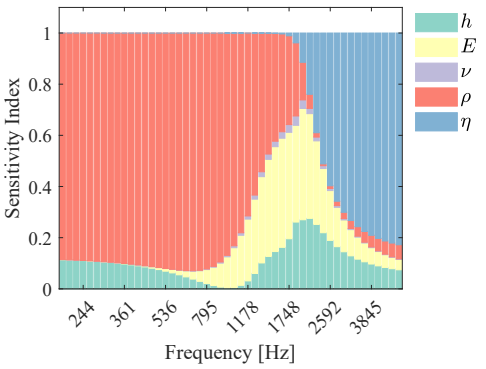
Figure 3.1: Comparison of Sensitivity Analyses approaches.

Furthermore, once a surrogate model is trained, it can quickly perform many forwarding analyses and be applied for classical sensitivity analysis. To illustrate this application, the NN-based surrogate trained in Section 2.3.1 with 500 supporting points was used to replace the true STL model in the calculation of Sobol's sensitivity indices with 24576 evaluations. While the elapsed time using the true model was $1867s$, the analysis using the NN-based surrogate took $16s$, i.e., around 116 times faster. At a previous stage, the sampling and training of the NN-based surrogate took $140s$.Figure 3.3 presents the first-order Sobol's sensitivity indices calculated with both the true model and the NN surrogate model, from which the similarity of the results can be observed.

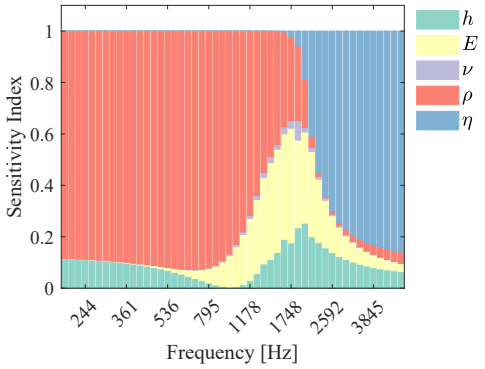
The advantage of the Sobol approach compared to MDI is that it allows for separately evaluating first and total order indices, therefore, facilitating the identification of input interactions or additive behavior. Figure 3.4a shows the total order of Sobol's indices, indicating that the interactions between inputs only occur around $1800Hz$, i.e., at the critical frequency range. Moreover, the higher order sensitivity indices indicated in Figure 3.4b represent only the importance due to these interactions, clarifying which inputs are interacting.

3.2.2 Sensitivity Analysis of STL Surrogate Models with Different Materials

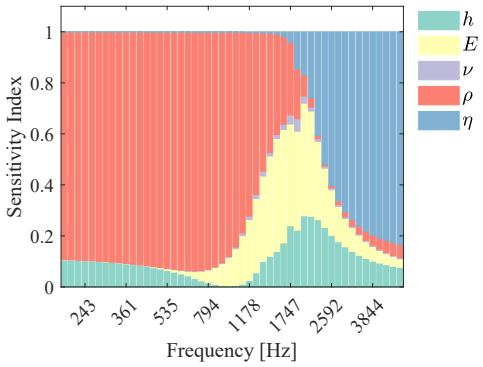
Surrogate-based sensitivity analysis can approximate expensive GSA and bring interpretability to the surrogate model, a constant demand from the scientific community regarding black-box models. Interpretability is also a prerequisite for the "explainable AI" field, which is critical for legal purposes as societies leverage AI. This section analyzes the STL for different materials and incident acoustic fields through surrogate models and the MDI sensitivity analysis. It should be noted that the MDI results depend on the selected design space, as is the case for every GSA, and thus, the results cannot be generalized to other input ranges. The analytical STL models and designed spaces used in this section are the same from Section 2.3.1. Each surrogate model was trained with 500 supporting points.



(a) Sobol total sensitivity indices.



(b) FAST total sensitivity indices.



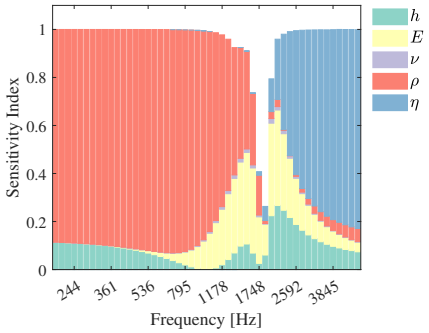
(c) MDI importance indices.

Figure 3.2: Total sensitivity indexes of the STL analyses for isotropic plates.

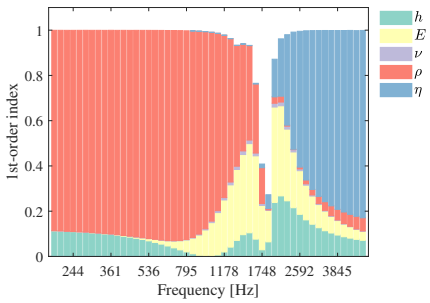
3.2.2.1 Isotropic Plates

According to the theory of STL for infinite thin isotropic plates [239], two response regions are expected: the inertia-controlled region, in which the mass law can approximate the STL, and the stiffness-controlled region. These regions correspond to the region where the first and second terms of the structural impedance of the plate in Equation (2.23) predominate, respectively.

The MDI sensitivity analysis performed for a plane wave with $\theta = 60°$ depicted in Figure 3.5 clearly shows both regions, the inertia controlled, where the variance in ρ and h defines all the output variance and goes up to around 2000 Hz, and the stiffness controlled region where E and h play major roles. The transition frequency range between these regions matches the range of the coincidence frequency for this space design, which ranges from approximately 1680 Hz to 2680 Hz. According to STL physics, the amplitude of the STL valley at the coincidence frequency is controlled by the plate damping η, which importance slightly rises at this frequency range in Figure 3.5. As noted by [242], when a set of interest rather than a specific plate design is considered, the location of the coincidence frequency dip is more relevant for the STL result than its amplitude, explaining why the damping importance is

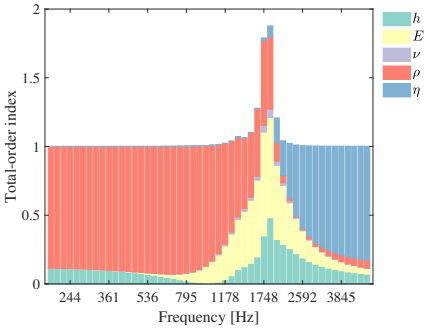


(a) True STL model is used to evaluate the outputs.

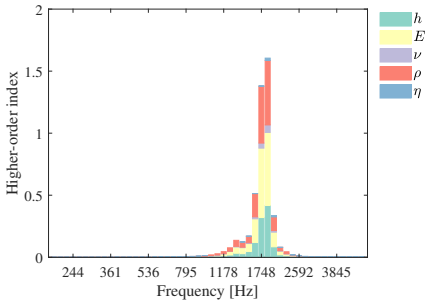


(b) NN-based surrogate is used to predict outputs.

Figure 3.3: First-order sensitivity indices with Sobol algorithm.



(a) Total order indices.



(b) Higher-order indices.

Figure 3.4: Sobol's sensitivity indices obtained with the NN-based surrogate for the isotropic thin infinite plate.

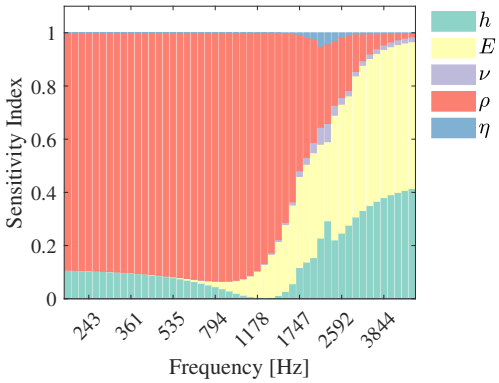


Figure 3.5: MDI indices of isotropic plates STL for a plane wave with $\theta = 60°$. Surrogate RMSE: $0.47 \pm 0.11 dB$.

smaller than expected.

In the case of an incident diffuse field (Figure 3.2), the coincidence phenomenon

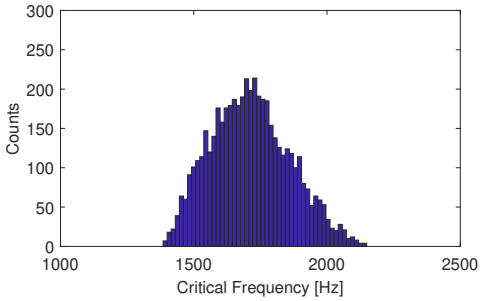


Figure 3.6: Histogram of the critical frequencies from the STL designs of isotropic plates in the set of interest from Table 2.1.

Table 3.1: Main MDI importances at different regions of STL of isotropic plates.

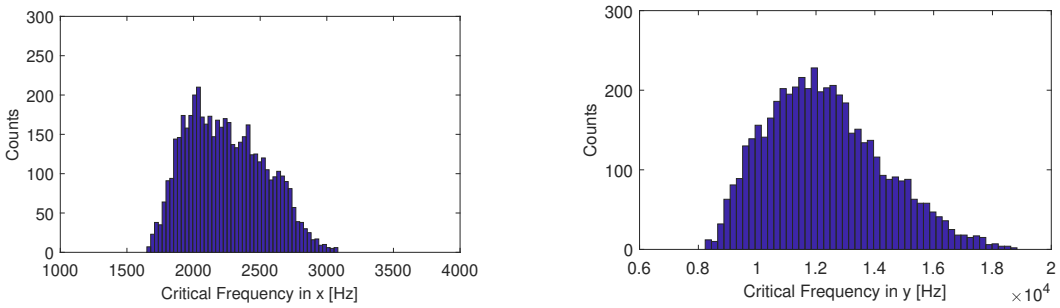
	$\omega < \omega_c$	At $\omega \approx \omega_c$	$\omega > \omega_c$
Plane Wave	ρ, h	ρ, h, E, η	E, h
Diffuse Field	ρ, h	E,h	η

occurs at the frequencies above the critical frequency, which varies approximately between 1400 and 2000 Hz. This range coincides with the range of critical frequencies calculated according to Equation (2.4) and shown in Figure 3.6 for the design space considered. In this frequency range, the importance of the inputs E and h, related to the stiffness, predominates. At higher frequencies in the coincidence region, the damping importance increases because for each frequency $\omega > \omega_c$ there will be a component of the diffuse field with an angle θ such that $\omega_{coinc}(\theta) = \omega$, and the amplitude is again the significant aspect of the STL. Thus, predicting the dip location is essential to minimize the error at the critical frequency range, while the STL amplitude is more important in the coincidence region. These results are summarized in Table 3.1 and corroborate that the RF-based surrogate is physically consistent and interpretable.

3.2.2.2 Orthotropic Plates

In the case of orthotropic plates, the coincidence phenomena occur in multiple directions, leading to multiple coincidence frequencies for each incident angle. To these phenomena, the STL's sensitivity indices of orthotropic plates were evaluated for incident plane waves with $\theta = 80°$, close to grazing angle, and with azimuth angle ϕ equals 0°, 45°, and 90°. The results are depicted in Figures 3.8a to 3.8c. It is noted that, regardless of the direction of the incident wave, the low-frequency region is controlled by plate inertia. Nonetheless, the behavior of the MDI indices in the high-frequency region differs for each wave direction.

The results for $\phi = 0°$ are similar to the ones for the isotropic plate (Figure 3.5), where h and E_x are the most important variables. Besides that, the small peaks of damping and stiffness importances coincide with the beginning of the theoretical



(a) Range of critical frequencies in x direction.

(b) Range of critical frequencies in y direction.

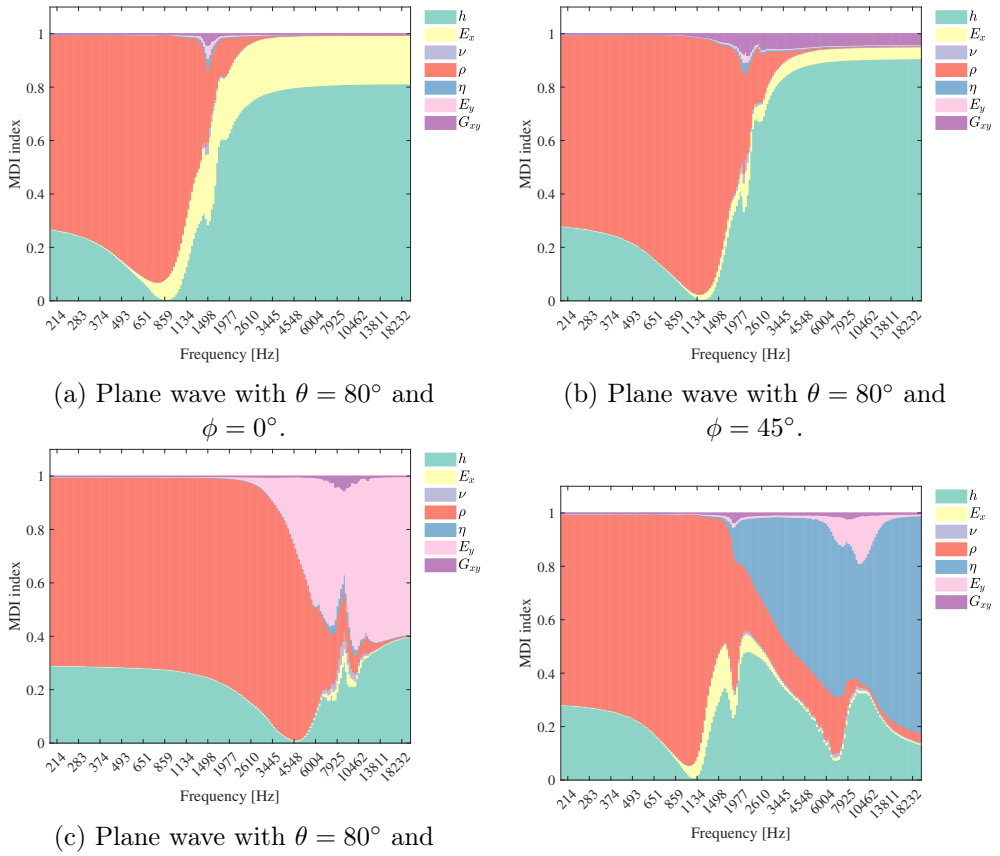
Figure 3.7: Histogram of the critical frequencies from the STL orthotropic plate designs in the set of interest from Table 2.2.

range of $\omega_{c,x}$ (Equation (2.27)), which ranges from approximately 1500 to 3000Hz, as depicted in Figure 3.7a. Similar peaks are identified for the results with $\phi = 45°$ at higher frequencies, indicating where the coincidence phenomenon starts. The in-plane shear modulus G_{xy} has considerably more influence when $\phi = 45°$, which aligns with the expected physical behavior for a highly orthotropic material. Finally, for $\phi = 90°$, the bending stiffness in the y direction becomes the most important variable at high frequencies, and the coincidence peak also coincides with the start of $\omega_{c,y}$ range, which goes from around 8 to 16 kHz (Figure 3.7b).

Figure 3.8d presents the MDI indices for the diffuse field analyses, where there is a remarkable increase in E_x and E_y importances in the regions of $\omega_{c,x}$ and $\omega_{c,y}$ ranges, respectively. In both regions, h also had increased importance. In the entire coincidence region, the damping importance is primordial. The findings for the MDI indices of the orthotropic plates are summarized in Table 3.2. Generally, for STL with an incident plane wave, the STL at $\omega > \omega_{coinc}$ is controlled by the corresponding stiffness. For STL with the diffuse field, the importance of variables related to stiffness increases at the critical frequencies, while the coincidence region is damping-controlled.

Table 3.2: Main MDI importances at different regions of STL for orthotropic plates. Results were obtained for plane wave (PW) cases with incident angle $\theta = 80°$ and variable azimuth angle ϕ, and for the diffuse field case.

	$\omega < \omega_{c_x}$	$\omega \approx \omega_{c_x}$	$\omega_{c_x} < \omega < \omega_{c_y}$	$\omega \approx \omega_{c_y}$	$\omega > \omega_{c_y}$
PW: $\phi = 0°$	ρ, h	ρ, h,E_x,η	E_x, h	E_x,h	E_x,h
PW: $\phi = 45°$	ρ, h	ρ, h	ρ, h,G_{xy},η	E_x,G_{xy}, h	E_x,G_{xy}, h
PW: $\phi = 90°$	ρ, h	ρ, h	ρ, h	ρ, h, E_y,η	E_y, h
Diffuse Field	ρ, h	E_x, h	η	E_y, η	η



(a) Plane wave with $\theta = 80°$ and $\phi = 0°$.

(b) Plane wave with $\theta = 80°$ and $\phi = 45°$.

(c) Plane wave with $\theta = 80°$ and $\phi = 90°$.

(d) Diffuse Field.

Figure 3.8: MDI indices of the STL analyses of orthotropic plates.

3.2.2.3 Sandwich Panels

The MDI indices were assessed for the STL using an incident plane wave with an angle of $\theta = 80°$ (see Figure 3.10a) and the STL with a diffuse field (see Figure 3.10b). Notably, similar to the damping coefficient η, the out-of-plane core shear modulus G does not significantly affect the STL response to a plane wave. However, its importance becomes significantly higher at high frequencies in the case of a diffuse field. Additionally, the core thickness h_c plays a crucial role in determining the coincidence region, as it influences the plate's bending stiffness. Once again, the change in behavior of the MDI indices in the diffuse field aligns with the range of critical frequencies calculated using Equation (2.33), as depicted in Figure 3.9

Table 3.3 summarizes the main MDI importances obtained for the sandwich plate analyses.

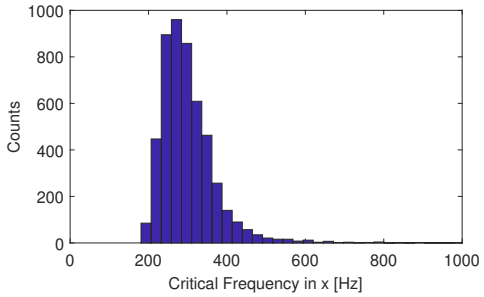
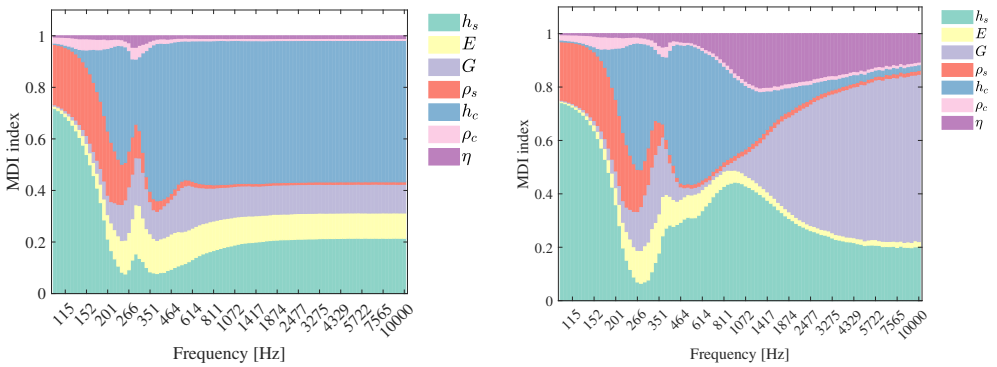


Figure 3.9: Histogram of the critical frequencies from the STL designs in the set of interest from Table 2.3.



(a) Plane wave with $\theta = 80°$.

(b) Diffuse Field.

Figure 3.10: MDI indices of the STL analyses of sandwich plates evaluated with RF-based surrogate.

Table 3.3: Main MDI importances at different regions of STL of sandwich plates.

	$\omega < \omega_c$	At $\omega \approx \omega_c$	$\omega > \omega_c$
Plane Wave	ρ_s, h_s	$E, h_{s,c}$, peak in η and G	$E_s, h_{s,c}, G$
Diffuse Field	ρ_s, h_s	$E, h_{s,c}$, peak in η and G	η, G, h_s

3.2.3 The Impact of Physics-Guided Features and Feature Screening on ML Interpretability and Accuracy

Chapter 2 showed that including PGF during surrogate training can enhance prediction accuracy. This section evaluates the impact of PGFs on the surrogate's interpretability and sensitivity indices. Furthermore, the information from sensitivity indices allows us to identify the most relevant features and perform feature screening, i.e., remove unimportant features to reduce the problem dimensionality.

The investigation is based on the MDI analyses for isotropic, orthotropic, and sandwich plates with an incident diffuse field. The PGF are the same used in Chapter 2 and presented in Table 3.5. The RFs are retrained with the same databases used in previous sections, and therefore, the time elapsed to perform the following

analyses is negligible. The feature screening procedure excluded features with average importance smaller than 5% and maximum importance for individual outputs smaller than 20%. Table 3.4 shows the PGF added for each analysis and underlines the selected relevant features.

Table 3.4: Physics-guided features included and important variables selected underlined.

	STL Variables	**Physics-guided features**
Isotropic Plates	$h, \underline{E}, \nu, \underline{\rho}, \underline{\eta}$	$\underline{m}, \underline{D}$
Orthotropic Plates	$h, E_x, E_y, G_{xy}, \nu, \underline{\rho}, \underline{\eta}$	$\underline{m}, \underline{D_x}, \underline{D_y}, D_{xy}$
Sandwich Plates	$h_s, E, \underline{G}, \underline{\rho_s}, \underline{h_c}, \rho_c, \underline{\eta}$	$\underline{m}, \underline{D}, \underline{N}, \underline{Y}$

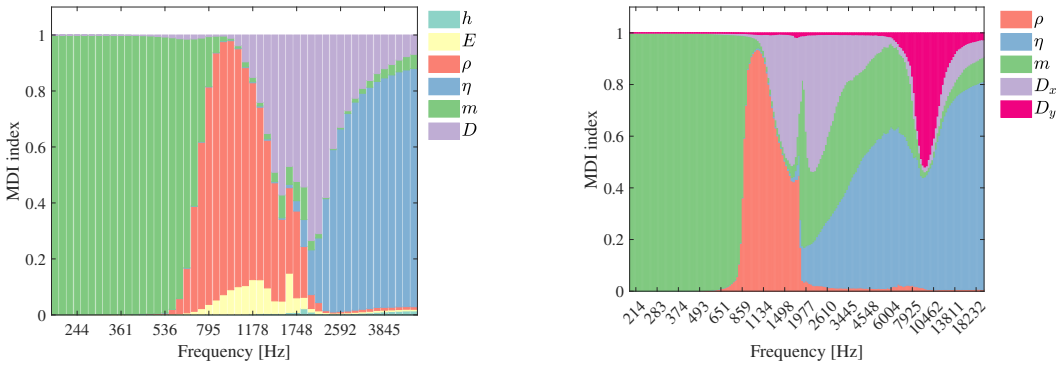
Figure 3.11 shows the MDI indices for the plate materials considered. Four general aspects are noted:

- In the mass-controlled region, the variance is entirely defined by m.
- The regions of critical frequencies are preceded by an increase in the importance of densities ρ with a subsequent increase in the stiffness related to the vibration mode of the coincidence phenomenon.
- In the coincidence regions, the STL variability is composed mainly of the effects of damping η and, for the sandwich case, core shear modulus G.
- Input interactions occur mainly at the critical frequency range.

The results show that the variable effects can be better represented within the added physical-guided terms. For example, the original analysis of the isotropic case (Figure 3.2c) showed a contribution of h both at low and high frequencies, while in Figure 3.11a, these contributions are encompassed by m and D at low and high frequencies, respectively. However, it should be stressed that the PGFs depend on or correlate with other input variables, and the MDI indices may be biased towards them [276]. Therefore, the MDI indices of Figure 3.11a can help identify the contributions of complex physical terms in the system output, but they should not quantitatively determine their importance. In addition to assisting the interpretation of MDI analysis results, the addition of the engineered features guides the RF model to follow the known physics. In this way, the PGFs increase the physical consistency of the black box and improve its accuracy, as seen in Table 3.5.

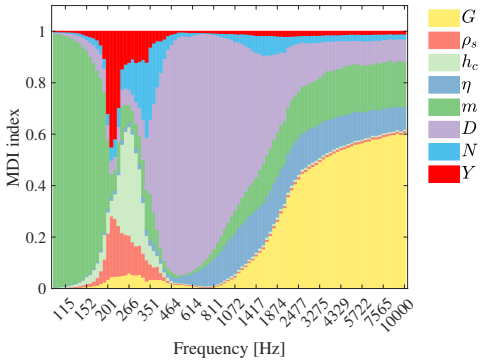
3.2.4 Sensitivity Analysis of STL Surrogate Models with Different Complexities

This section investigates the interpretability and consistency of the ML-based surrogates with different physics-driven models of STL presented in Section 2.3.2. MDI-based sensitivity analysis reveals the importance of each parameter in the system response, rendering the surrogate more interpretable. Additionally, the addition of



(a) Isotropic Plate.

(b) Orthotropic Plate.



(c) Sandwich Plate.

Figure 3.11: MDI indices of STL analyses with addition of PGF and carrying out feature screening.

PGF to the RF models, namely mass density m, the real part of the bending stiffness D_R, and the resonance coefficient term R, is evaluated together with reasoning on the physical insights from the GSA analyses.

For all cases, the critical frequencies for this design space range between approximately 1000 and 2500 Hz, as shown in Figure 3.12. One RF regressor with 200 decision trees was trained for each frequency of interest to obtain the features' importance of the outputs.

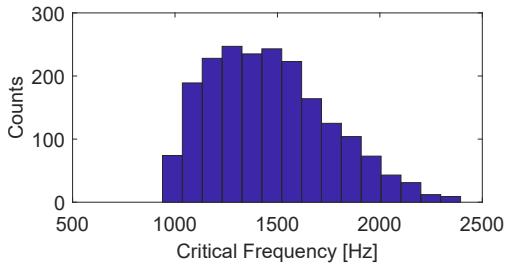
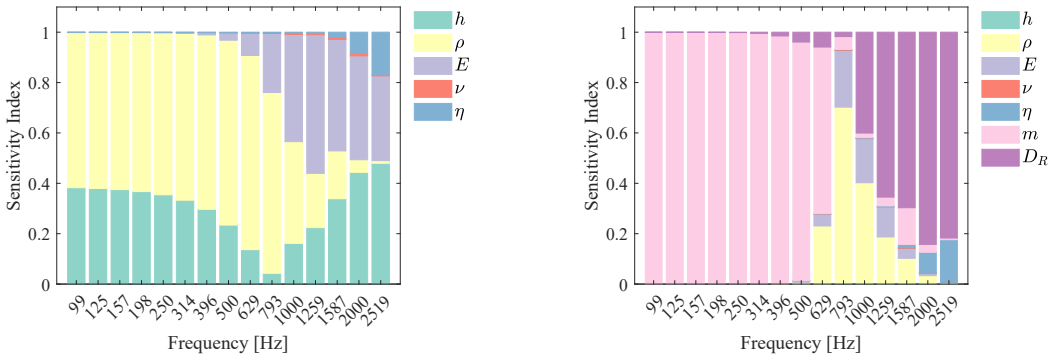


Figure 3.12: Histogram of the critical frequencies from the STL designs in the set of interest from Table 2.4.

Table 3.5: Accuracy of the RF-based surrogate with and without Physics-Guided Features (PGF) in terms of the RMSE.

	Accuracy without PGF and Feature Screening	Accuracy with PGF and Feature Screening
Isotropic Plates	0.50 dB	0.35 dB
Orthotropic Plates	0.27 dB	0.23 dB
Sandwich Plates	1.10 dB	0.88 dB

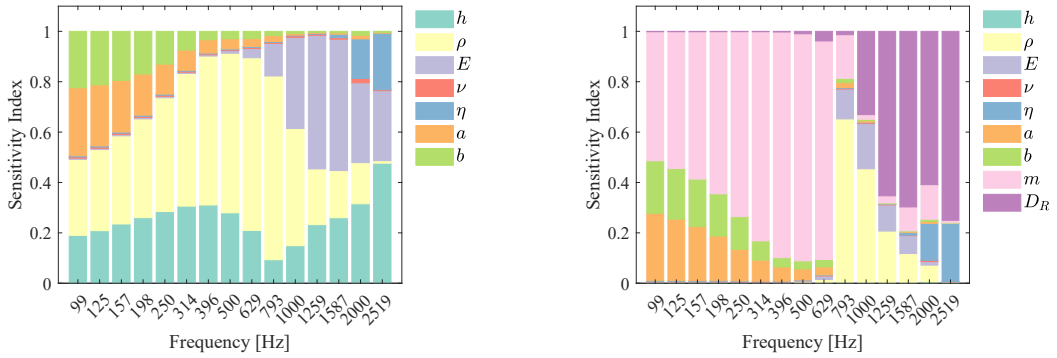
Figure 3.13a,b show the MDI-based sensitivity index obtained with the infinite plate models with and without using PGFs, respectively. The same observations made in Section 3.2.2 for the isotropic case stand here, i.e., distinctly mass-controlled and stiffness-controlled regions and the increasing importance of damping at higher frequencies. Furthermore, including the PGFs improves the surrogate accuracy from an RMSE of 0.21 ± 0.00 dB to 0.12 ± 0.01 dB.



(a) (b)

Figure 3.13: The mean decrease in impurity (MDI)-based sensitivity indices of the STL with the analytical infinite plate model: (**a**) the inputs of the surrogate only consist of the STL variables ($RMSE = 0.21 \pm 0.00$ dB); (**b**) m and D_R are included as PGFs for the surrogate ($RMSE = 0.12 \pm 0.01$ dB).

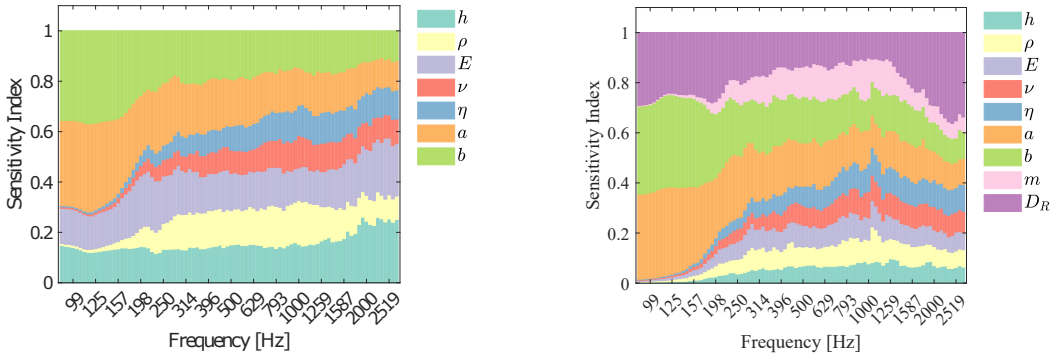
The MDI-based sensitivity indices change considerably when the model considers the structural dynamics of the finite plate and its resonant modes, as for the STL results with MS and FEM approaches in Figures 3.15 and 3.16, respectively. Overall, the sensitivity indices of the MS and FEM approaches are similar, indicating that they model the vibroacoustic phenomena with analogous considerations. The bending stiffness and the plate dimensions have a considerably high sensitivity index throughout the frequency range. Therefore, the behavior of the finite plates under analysis diverges from the one of a limp mass, an assumption held by the mass law and considered in the mass-controlled region of both the analytical approach for infinite plates and the correction factor approach for finite plates. Indeed, the mass density importance is negligible in the resonant region, and its influence in the



(a) (b)

Figure 3.14: MDI-based sensitivity of the STL of finite plates evaluated with the correction factor: (**a**) the inputs of the surrogate only consist of the STL variables ($RMSE = 0.36 \pm 0.01$ dB); (**b**) m and D_R are included as PGFs for the surrogate ($RMSE = 0.25 \pm 0.01$ dB).

middle-frequency range is not predominant.

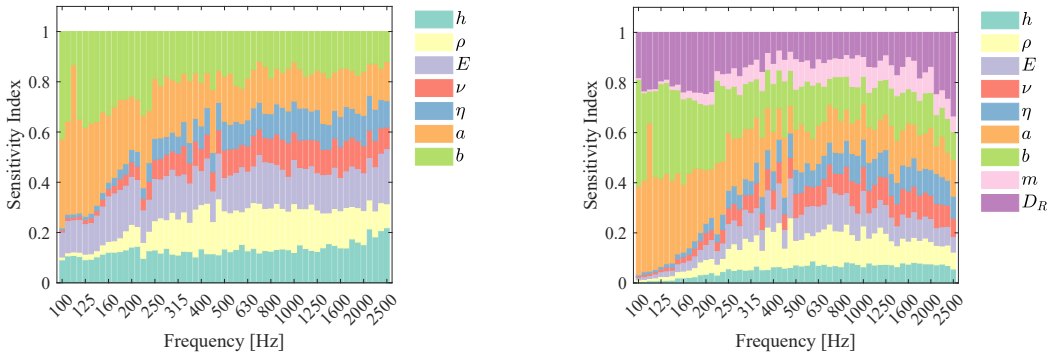


(a) (b)

Figure 3.15: MDI-based sensitivity of the STL of finite plates evaluated with modal summation: (**a**) the inputs of the surrogate only consist of the STL variables ($RMSE = 3.38 \pm 0.03$ dB); (**b**) m and D_R are included as PGFs for the surrogate ($RMSE = 3.24 \pm 0.02$ dB).

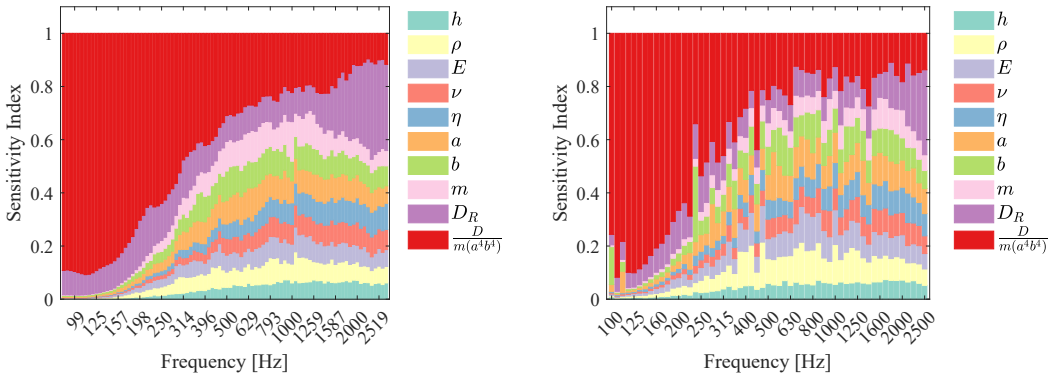
The sensitivity indices of STL of finite plates based on the correction factor approach (Figure 3.14) differ from the infinite plate indices only in the low-frequency region. This result is coherent with the methodology, which applies a correction factor in the low-frequency range to account for the plate dimensions and comprehend its resonant behavior. Although the plate stiffness would theoretically also impact the resonant modes of the structure, the STL evaluated by the correction factor is insensitive to this feature in the resonant region of Figure 3.14. Once more, the RMSE decreases from 0.36 dB to 0.25 dB with the inclusion of PGFs.

The achieved surrogate accuracy with PGFs is 3.24 ± 0.02 dB for the MS-based



(a) (b)

Figure 3.16: MDI-based sensitivity of the STL of finite plates evaluated with FEM: (**a**) the inputs of the surrogate only consist of the STL variables ($RMSE = 5.76 \pm 0.06$ dB); (**b**) m and D_R are included as PGFs for the surrogate ($RMSE = 5.60 \pm 0.06$ dB).



(a) (b)

Figure 3.17: MDI-based sensitivity of the STL of finite plates including mass density m, the real part of bending stiffness D_R and the resonance coefficient term R as PGF: (**a**) sensitivity indices from MS dataset ($RMSE = 2.67 \pm 0.02$ dB); (**b**) sensitivity indices from FEM dataset ($RMSE = 4.86 \pm 0.05$ dB).

results and 5.60 ± 0.06 dB for the FEM-based results. The high complexity and non-smooth behavior of the STL response with ML and FEM approaches lead to relatively poor surrogate accuracy even when m and D are used as PGFs, compromising the sensitivity analyses' accuracy. In particular, at high frequencies, the even distribution of the sensitivity indices among the features indicates higher complexity and possibly higher inaccuracies. As the modal behavior of the plate is posing difficulties for the surrogate model, the resonance coefficient term R (Equation (2.48)), which relates to the plate natural frequencies, is included as another PGF for the surrogates of the FEM and MS models.

Figure 3.17 shows that the resonance coefficient feature controls the STL in the

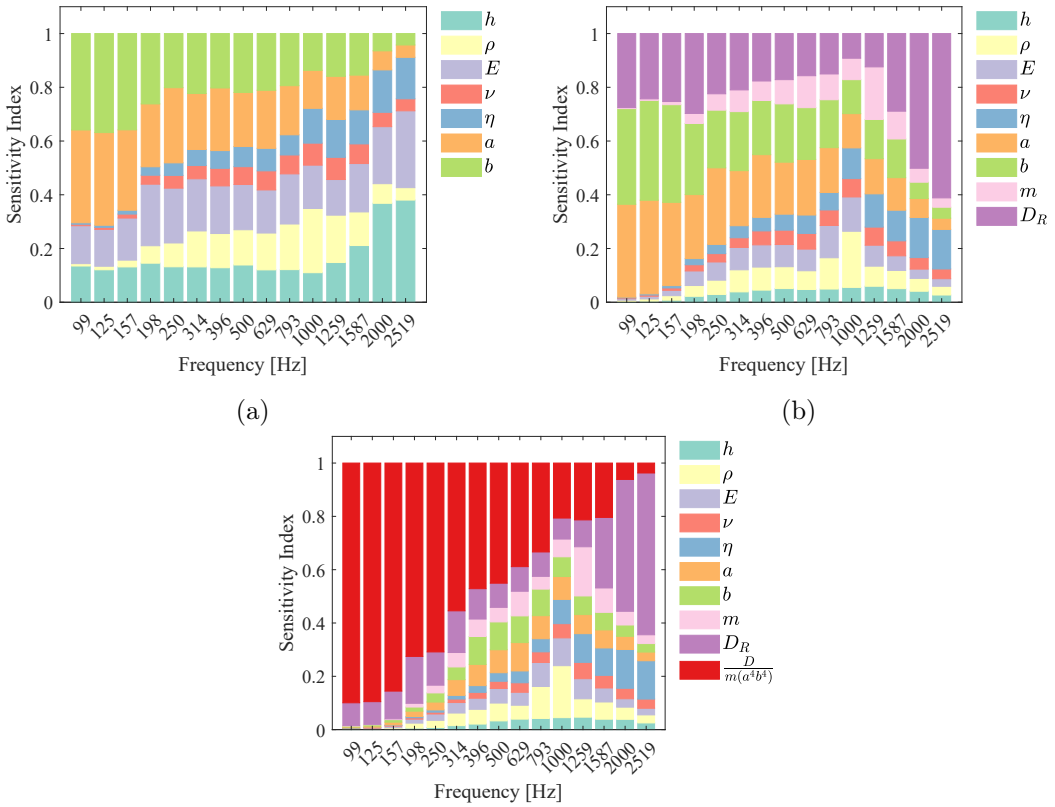


Figure 3.18: MDI-based sensitivity of the 1/3 octave band average STL of finite plates evaluated by modal summation (MS):(**a**) the inputs of the surrogate only consist of the STL variables ($RMSE = 2.19 \pm 0.05$ dB); (**b**) m and D_R are included as PGFs for the surrogate ($RMSE = 2.01 \pm 0.02$ dB);(**c**) m, D_R and R are included as PGFs for the surrogate ($RMSE = 1.49 \pm 0.02$ dB).

low-frequency range. Although the accuracy improved to 2.67 ± 0.02 dB for the MS-based results and to 4.86 ± 0.05 dB for the FEM-based results, it remains elevated. Additional improvements in the accuracy could be obtained by feature selection, as performed in the previous section.

Finally, the MDI-based sensitivity index is evaluated with one-third octave band average STL with MS and FEM, as shown in Figures 3.18 and 3.19, respectively. Once again, the stiffness and dimensions of the plate play a major role in all frequencies. The surrogate models that include the three PGFs have the best accuracy, which is $RMSE = 1.49 \pm 0.02$ dB for the MS model and $RMSE = 1.63 \pm 0.03$ dB for the FEM model. These surrogates have better accuracy as the band averaging smooths the STL curve while keeping the information valuable for practical purposes.

The results presented in this section show evidence of the physical consistency

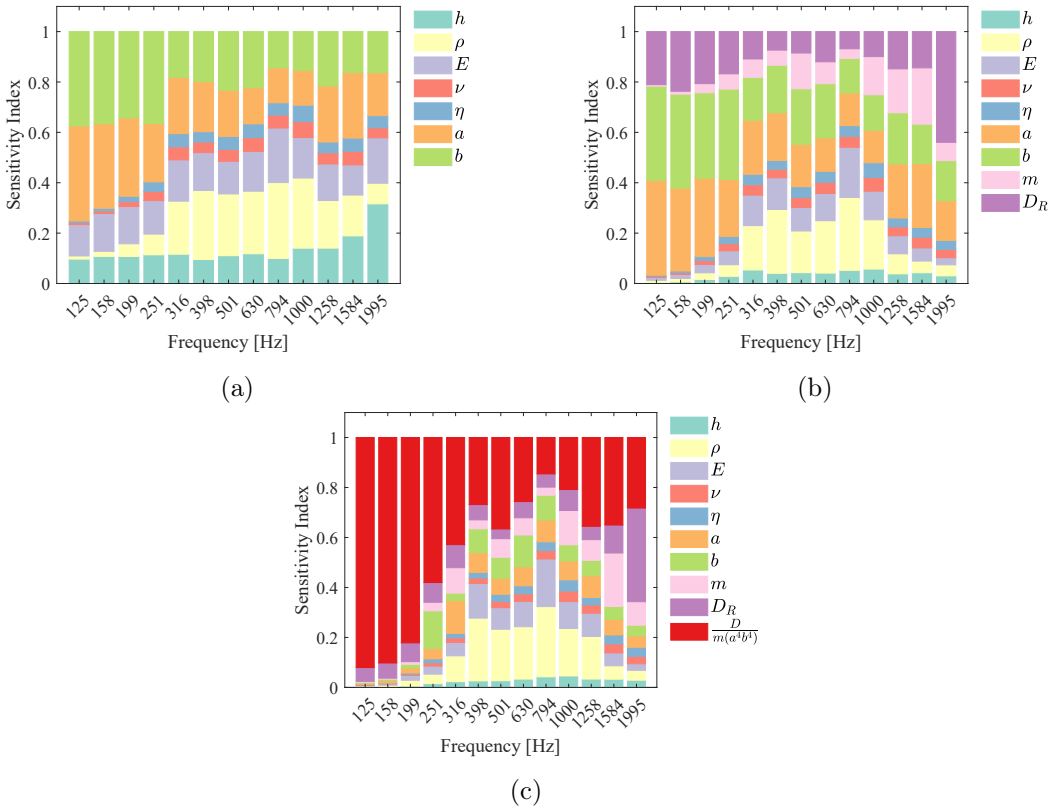


Figure 3.19: MDI-based sensitivity of the 1/3 octave band average STL of finite plate evaluated by FEM: (**a**) the inputs of the surrogate only consist of the STL variables ($RMSE = 2.23 \pm 0.02$ dB); (**b**) m and D_R are included as PGFs for the surrogate ($RMSE = 2.09 \pm 0.02$ dB); (**c**) m, D_R and R are included as PGFs for the surrogate ($RMSE = 1.63 \pm 0.03$ dB).

of the surrogate models, as the STL sensitivity indices have a behavior coherent with the literature of each STL physics-driven model. Furthermore, the sensitivity indices of the width and length of the plate are consistently equivalent in all analyses, as expected for isotropic materials. Physical inconsistencies could indicate regions where the ML model is not accurate. The investigation also demonstrates that PGFs can be readily included to improve surrogate accuracy and lead to a more apparent distinction of STL regions in the sensitivity analysis. However, interpreting the resulting feature's importance is not straightforward when PGFs are included, as the input features are no longer independent.

3.3 Conclusion

Interpretable and physics-guided RF-based surrogate models were implemented for the STL analysis through MDI sensitivity analyses and physics-guided features.

The MDI-based sensitivity indices showed good accuracy related to the reference Sobol's total sensitivity indices even for a small number of function evaluations. The MDI method proves to be potentially advantageous over FAST since it does not depend on a minimum number of evaluation points and specific sampling methods, furthermore, its performance was superior to or close to FAST performance for the same number of evaluations. However, it should be highlighted that the MDI accuracy depends on the RF's accuracy. The inclusion of physics-guided features was beneficial for surrogate accuracy. Furthermore, the combined use of MDI with feature engineering allows to explore the importance of user-defined terms, clarifying the physical relationships of the analyses.

These capabilities were illustrated for the STL analyses performed, which elucidated different aspects of the STL.Therefore, physics-based interpretations enabled by the methods implemented can increase the level of confidence in the surrogate and allow an in-depth comprehension of the physical phenomena with the potential to boost scientific discovery supported by data. In summary, the MDI-based sensitivity analyses presented in this chapter are a cheap-to-evaluate method that improves the surrogate interpretability and reliability and allows for the use of data-based information to deepen physical discussions on variable importance and interactions.

Chapter 4

Deep Transfer Learning for Inter-Generational Digital Twin Prototyping under Data Scarcity: A Viability Study

This chapter investigates the viability of using Transfer Learning (TL) for early Digital Twin (DT) construction under data scarcity, particularly in the context of surrogate models. Previous chapters have shown that Neural Network (NN) algorithms are suitable for constructing DT in vibroacoustic problems, albeit with a significant demand for data. In response to this data-demanding nature, we propose using TL techniques to harness information from previous product generations, thereby mitigating the need for extensive new data. Notably, there is a research gap in the well-established use of TL for applications with tabular data and conditional shifts, as is the case for surrogate models. To investigate the effectiveness of TL in this context, we evaluate various TL approaches, assessing their performance in transferring information between functions with controlled correlation. This chapter complements the previous chapters by addressing the issue of data scarcity in the development of accurate DT for vibroacoustic problems. It also sets the stage for applying DT in industrial settings where data availability is limited.

Contents

Introduction

DTs play a key role in supporting information management and data leverage for Product Lifecycle Management (PLM). In real-world PLM scenarios, the experience accumulated from past product generations significantly contributes to the design of new products, leading to more efficient and effective product development processes. However, knowledge maintenance heavily relies on the expertise of product designers and lacks systematic approaches to ensure information continuity across different product generations. This chapter aims to bridge this research gap by investigating the application of TL to automate and systematize the data leverage from previous product generations. By doing so, the chapter seeks to allow DTs to evolve across different product generations in a data-efficient manner, reducing the reliance on individual expertise and on the acquisition of new data at each new generation of products.

While traditional ML approaches learn a task from scratch, TL takes advantage of knowledge gained from previous ML problems $\langle \mathcal{D}_s, \mathcal{T}_s \rangle$, the source problems, to improve the performance of the predictive function $h_T(\cdot)$ in a related target ML problem $\langle \mathcal{D}_t, \mathcal{T}_t \rangle$, where either $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$, or both [278, 217] (Figure 4.1). When the source and the target problems are related, the leverage of latent knowledge from the larger source dataset can potentially minimize the requirement for abundant high-quality data in the target problem, thereby improving data efficiency and model accuracy [217]. The main challenge in TL lies in alleviating shifts in the input feature space (heterogeneous TL) or data distribution (homogeneous TL) between the source and target domains [279].

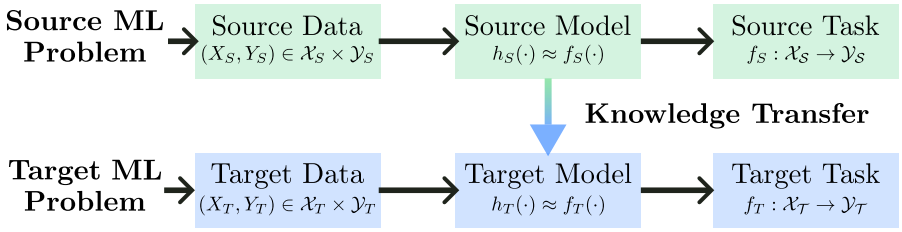


Figure 4.1: Transfer Learning representation.

The TL literature encompasses a wide range of techniques that offer significant benefits, especially for DL models, as TL reduces the model's reliance on large datasets [217, 218, 279]. TL's success has been remarkable in various fields, including autonomous driving systems, natural language processing, recommendation systems, and computer vision [218, 278].

In the field of SD&V, TL has gained significance, particularly in its application to Population-Based SHM (PBSHM) [42, 43, 219–222]. PBSHM focuses on transferring damage inference and damage-sensitive features within a population of structures or machines. This transfer can occur between nominally identical instances exhibiting variations in parameters, environmental conditions, and operational conditions (homogeneous PBSHM) or between distinct instances that share certain similarities

(heterogeneous PBSHM) [42]. In both homogeneous and heterogeneous PBSHM, feature-based TL approaches have mainly been used to address feature distribution discrepancies between the source and target domains [223, 70, 224, 225, 11]. Additionally, TL combined with graph representations has been employed to identify structural similarities between instances, facilitating the transfer of information in heterogeneous PBSHM [219, 221, 226].

Another relevant application of TL in SD&V is the transfer of knowledge from simulated to real domains. In this context, virtual models generate abundant numerical data in the source domain, which TL leverages to enhance predictions based on limited experimental data in the target domain. For instance, Trinh et al. [228] addressed scarce experimental datasets by applying TL to augment them with simulated data, enabling accurate prediction of acoustic responses parameterized by microstructural properties despite the scarcity of experimental data. In damage detection, Ai et al. [227] employed domain adaptation to transfer information from acoustic emission signals generated using FEM simulations to real-world scenarios, accurately locating the source of acoustic emission signals even in the absence of labeled experimental data. Furthermore, TL from simulated to real environments has been employed to reduce training costs in reinforcement learning for robotic control [280, 281].

Despite TL's promising outcomes and the increasing use of ML-based surrogate models, the combination of TL and ML-based surrogates remains relatively unexplored in the literature. Most existing studies focus on applying TL to construct multi-fidelity surrogate models. For example, Jiang and Durlofsky [229] applied TL techniques to develop a subsurface flow prediction surrogate model. They trained an NN with low-fidelity simulation data and fine-tuned it using a limited amount of high-fidelity simulation data, achieving a remarkable 90% reduction in training simulation costs while maintaining accurate predictions. Geng et al. [230] utilized TL with a deep reinforcement learning algorithm to construct a multi-fidelity surrogate model of a propeller blade and optimize its aerodynamic and aeroacoustic performance. They leveraged TL techniques to improve both the surrogate model and the optimization algorithm, leading to significant computational cost savings and noise reduction. Introducing an instance-based TL method, Kim and Lee [231] used a modified domain-adversarial NN to transfer information between the surrogate models of distinct designs. They augmented target-domain through domain adaptation techniques and subsequently selected helpful data, resulting in improved predictions for drone surrogate modeling with limited datasets.

To the best of the authors' knowledge, no prior research has explored the application of TL to transfer knowledge between different product generations. This study introduces a novel use of deep TL to aid in DT prototyping by leveraging data from previous product generations. We assume that since previous and current generations share similarities and are governed by the same physical principles, their domains are inherently related, enabling the transfer of valuable information between them.

However, this application involves applying TL to a conditional shift regression

problem with relatively low dimensionality, while most TL studies, especially in deep TL, focus on high-dimensional classification problems with covariate shift. Notably, Borisov et al. [232] highlighted the absence of well-established TL methodologies for tabular data, and Liu et al. [233] and Chen et al. [282] emphasized the lack of studies on conditional shift regression problems. Therefore, our first objective is to examine the viability of deep TL methods under conditional shift with tabular data and evaluate the best approaches considering the discrepancy between the source and target domains. We also evaluate the impact of TL under different degrees of non-smoothness to ensure applicability in the SD&V domain.

The main contributions of this chapter can be summarized as follows:

- Introducing TL as a valuable approach to leverage data from previous generations of DT prototypes, ensuring continuity of valuable knowledge and insights to enhance future DT prototypes and their corresponding physical counterparts.
- Developing a synthetic function to evaluate TL performance, considering controlled relatedness between functions and controlled degrees of smoothness.
- Investigating the performance of deep TL with low-dimensional tabular data, focusing on regression problems with conditional shifts between source and target domains.
- Providing insights into the viability of deep TL for inter-generational surrogate modeling with different relatedness between the source and target domains.
- Evaluating deep TL performance across different smoothness levels to assess its applicability for SD&V problems.
- Identifying suitable parameter-based deep TL approaches that can be integrated into the proposed DT framework to enable accurate DL surrogate models with scarce data.
- Bridging the research gap in the evolution of DT prototypes across different product generations.

This chapter is organized as follows: Section 4.1 presents the methodology and implementation details of the deep TL techniques. Section 4.2 describes the methodology for generating functions sampled from GPs with controlled smoothness and relatedness between source and target domains. This set of functions is instrumental in evaluating the performance of TL. Section 4.3 investigates the effectiveness of different deep TL approaches for knowledge transfer in surrogate modeling. The analyses also assess the effects of TL hyperparameters, target dataset size, and function smoothness on TL performance. Finally, Section 4.4 discusses and concludes the chapter.

4.1 Transfer Learning Methodology

This study proposes an approach to transfer information from a previous product to a new product generation using deep TL techniques. Although the previous and new products are related and share the same governing equations, changes in

the system's boundary conditions and constitutive relations will alter the system response function, therefore $f_S(\cdot) \neq f_T(\cdot)$. From a probabilistic perspective, this mismatch can be expressed as a conditional shift $P(Y_S|X_S) \neq P(Y_T|X_T)$ between the source and target problems. Additionally, while the design space $\mathcal{X}$ and the asymptotic marginal distribution $lim_{N\to\infty}P(X)$ remain the same for both product generations, the empirical marginal distribution between them will differ. This deviation is influenced by the accuracy and representativeness of the data collected, which depends on the sample size N. This difference introduces a covariate shift, meaning that the marginal distributions $P(X_S)$ and $P(X_T)$ are not exactly equal, even though they are closely related.

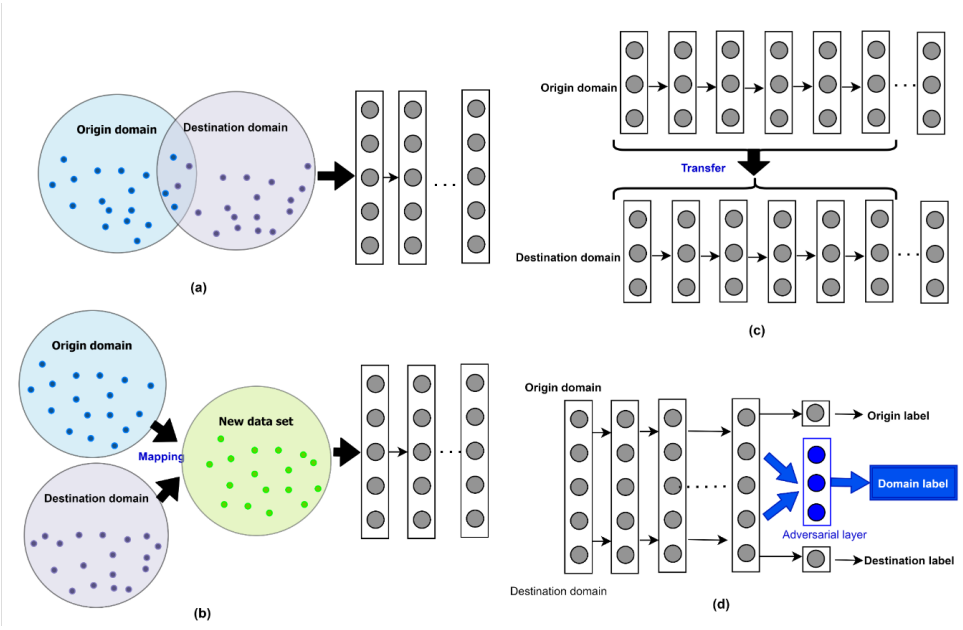


Figure 4.2: Categories of deep TL: (a) instance-based, (b) mapping-based, (c) parameter-based and (d) adversarial-based. Source: [278]

According to Tan et al. [278], deep TL methods can be categorized into four main approaches: instance-based, mapping-based, adversarial-based, and parameter-based, as depicted in Figure 4.2. Instance-based TL involves selecting and re-weighting instances from the source domain based on their proximity to the target domain, allowing their utilization in the target domain. Mapping-based TL, also known as feature-based TL [217], maps instances from both domains into a latent space encompassing shared features. Adversarial-based methods leverage adversarial techniques to extract domain-invariant features by training the network to learn indistinguishable features between the source and target domains. Finally, parameter-based or network-based TL involves reusing and fine-tuning pre-trained NN parameters from a source domain to a target domain.

However, as highlighted by Liu et al. [233], instance-based and adversarial-based methods may not be well-suited for addressing conditional shift problems. Addition-

ally, most metrics from mapping-based methods are primarily designed for adapting marginal distributions. Therefore, our study employs parameter-based approaches for TL, as they can effectively handle both conditional and covariate shifts when the source and target domains exhibit similarity and can be easily integrated into the DT framework.

Parameter-based TL methods are commonly used in deep TL applications, especially for high-dimensional classification problems. These approaches reuse and fine-tune the weights and biases from the source model in training the target model instead of training a new model from scratch. This methodology aims to expedite learning and enhance the model's performance on the target task by improving the starting point of optimizing NN parameters. The underlying assumption of parameter-based TL is that networks for similar tasks share common parameters [233].

Furthermore, parameter-based deep TL usually assumes that the initial layers of a deep NN model capture general low-level representations. In contrast, the top layers learn more domain-specific higher-level features. For instance, in a CNN, the initial layers identify low-level spatial features such as edges and basic shapes, while later layers learn more task-specific features such as the shape of specific objects. Therefore, the first layers should share more parameters between the source and target tasks. However, this assumption has mainly been studied for high-dimensional classification problems [283]. Further research is needed to determine if NN-based surrogates also learn transferable general-domain representations and if these representations are learned in the first layers of the NN.

There are two main methods of parameter-based deep TL: fine-tuning TL and feature extraction. In *fine-tuning TL*, the pre-trained source model is fine-tuned on the target dataset. This method considers the optimal weights and biases obtained from the pre-trained source model sub-optimal for the target domain. Therefore, they are used as starting points for optimizing NN parameters during training. This process enables the model to adapt to the specific characteristics of the target task while leveraging prior knowledge acquired from the source domain. On the other hand, *feature extraction TL* utilizes the representations learned from the pre-trained source model as inputs for a new model trained on the target task. In this approach, the top layers of the pre-trained model are discarded, and the output of the base layers, known as frozen layers or backbone layers, serves as input for a new model trained on target domain data. By reusing the pre-trained model's learned features, the target model can benefit from the underlying knowledge captured in the first layers from the source model.

In this study, we investigate the performance of four different deep TL configurations described below:

1. **Fine-tuning**: The entire pre-trained model is fine-tuned in the target domain with a small learning rate.
2. **Gradual**: Fine-tuning with a depth-dependent learning rate. The entire pre-trained model is fine-tuned with a learning rate that increases with the NN depth. This approach aims to preserve the low-level representations in the

initial layers while adapting the domain-specific higher-level features at the top layers more quickly.

3. **Frozen**: The weights and biases of the base layers from the pre-trained source model are kept frozen in the target domain. The parameters of the pre-trained model's top layers are discarded and restarted with random parameters, which are then trained in the target domain.
4. **Unfrozen**: This configuration follows the same procedure as the Frozen configuration. However, after training the top layers in the target domain, the backbone layers are unfrozen, and the entire model is trained in the target domain with a small learning rate.

To assess the impact of TL, we compare the results obtained with deep TL to a baseline model trained solely in the target domain without TL. The relative RMSE between the TL model and the baseline quantifies the improvement in prediction:

$$\frac{RMSE_{\text{with TL}}}{RMSE_{\text{without TL}}}. \tag{4.1}$$

A smaller value indicates a more significant reduction in prediction error due to TL.

4.2 Methodology for Generating Correlated Functions to Evaluate TL Performance

In order to investigate the feasibility and benefits of using deep TL in low-dimensional surrogate models in SD&V, it was essential to have a test function that could represent a wide range of functions while allowing for controlled smoothness and, most importantly, controlled relatedness between the source and target tasks. To address this need, we employed a synthetic data generation approach based on a Gaussian Process (GP) with a kernel prior $k(\boldsymbol{x}, \boldsymbol{x'})$ and zero mean $m(\boldsymbol{x})$. The definition of the kernel allows to set the desired prior information on the data distribution.

The methodology involved drawing samples from a GP according to the following:

$$Y = f(\boldsymbol{X}) \sim \mathcal{N}(m(\boldsymbol{X})\, k(\boldsymbol{X}, \boldsymbol{X})). \tag{4.2}$$

4.2.1 Functions with Controlled Smoothness

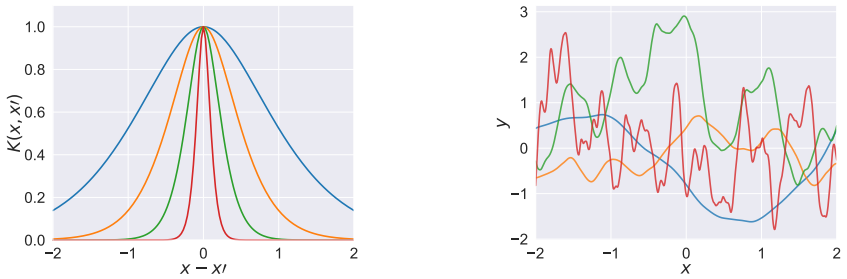
For generating input-output pairs in an n-dimensional input space $\mathcal{X}$, we employed the Matern kernel due to its ability to control the smoothness of the generated functions. These synthetic functions allowed us to assess the robustness of TL in SD&V scenarios. The Matern kernel prior can be defined as [252]:

$$k_\nu(\boldsymbol{x}, \boldsymbol{x'}) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\boldsymbol{x} - \boldsymbol{x'}\|}{\ell} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}\|\boldsymbol{x} - \boldsymbol{x'}\|}{\ell} \right), \tag{4.3}$$

where $\|\boldsymbol{x} - \boldsymbol{x'}\|$ is the Euclidean distance between two points $\boldsymbol{x}$ and $\boldsymbol{x'}$, K_ν is a modified Bessel function of the second kind, and Γ is the gamma function. The

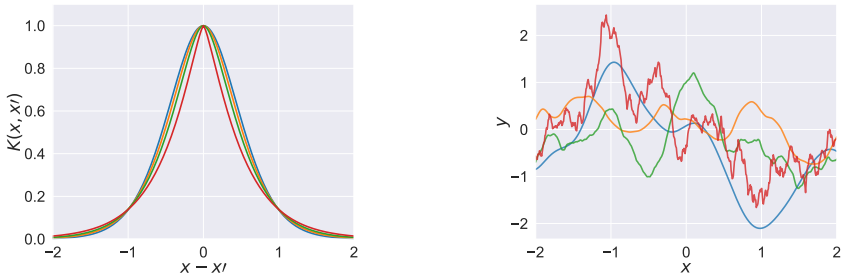
parameters ℓ, σ^2, and ν can be adjusted to fit the specific characteristics of the spatial process being modeled as follows:

1. The *variance parameter* σ^2 controls the overall magnitude of the function. A larger variance means that the function has a higher range of values.
2. The *length scale* ℓ controls the distance over which the function varies. A larger length scale means that the function varies more slowly, resulting in a smoother function.
3. The *smoothness parameter* ν controls the smoothness of the function. A smaller value of ν leads to a rougher, more wiggly function.



(a) Matern kernel. (b) Functions draw from the GP.

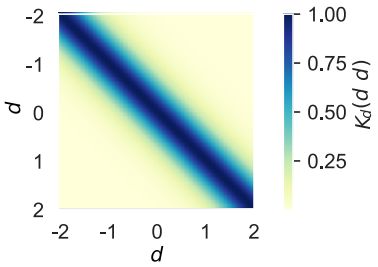
Figure 4.3: Effects of varying ℓ. Examples with constant $\nu = 2.5$ and — $\ell = 1.0$; — $\ell = 0.5$; — $\ell = 0.25$; or — $\ell = 0.1$.



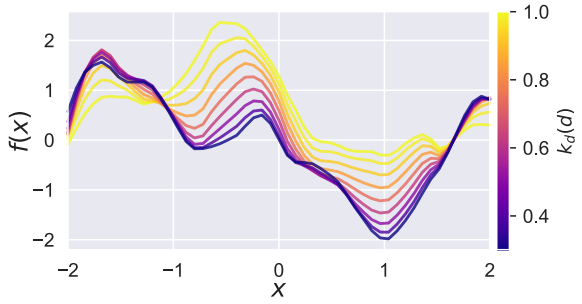
(a) Matern kernel. (b) Functions draw from the GP.

Figure 4.4: Effects of varying ν. Examples with constant $\ell = 0.5$ and — $\nu = 5.0$; — $\nu = 2.5$; — $\nu = 1.5$; or — $\nu = 0.75$.

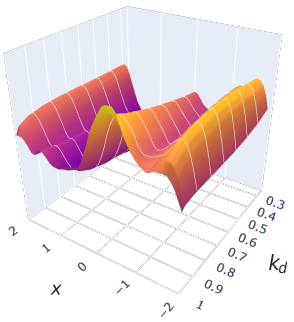
The tuning of the length scale ℓ and smoothness ν parameters allows to control the domain variability and smoothness, as depicted in Figures 4.3 and 4.4 for a one-dimensional example. Therefore, the proposed methodology can be used to generate domains with controlled different degrees of smoothness to evaluate TL's robustness for applications in the SD&V domain.



(a) Correlation matrix $k_d(d, d')$.



(b) 2D view of cross-sections of $f(x, d)$, with different values of constant d.



(c) 3D View of $f(x, d)$.

Figure 4.5: Illustrations of the synthetic functions generated with controlled relatedness for an example with $n = 1$, $\ell = 0.5$, and $\nu = 2.5$.

4.2.2 Controlling Relatedness between Functions

The previous approach allows sampling functions with controllable smoothness in n input dimensions. However, to introduce controlled relatedness between these functions, an additional dimension, denoted as d, represents the "distance" between them. The relationship between the functions in the d dimension is established using a RBF kernel. The RBF kernel is defined as follows:

$$k_d(d, d') = \sigma_d^2 \exp\left(-\frac{|d - d'|^2}{2\ell_d^2}\right), \tag{4.4}$$

where the parameters σ_d and ℓ_d are kept constant as $\sigma_d = 1$ and $\ell_d = 1$.

We define a kernel with $n + 1$ dimensions to sample different n-dimensional functions related to each other according to d. It can be formulated as follows [284]:

$$k_{n+1}(x_1, x_1'...x_n, x_n', d, d') = k_\nu(x_1, x_1'...x_n, x_n') \times k_d(d, d'). \tag{4.5}$$

Thus, one realization sampled from a GP with the k_{n+1} kernel results in a domain $f_{x,d} : \mathcal{X}, \mathcal{D} \to \mathcal{Y}$. Defining constant values in the d dimension creates the original domain space cross-sections. The cross-section with $d = 0$ is the source domain, and the cross-sections with $d = d_T$ are the target domains. Hence, the correlation

between a target domain at $d = d_T$ and the source domain at $d = 0$ is given by $k_d(0, d_T)$.

Figure 4.5 presents an example of one realization sampled from a GP with the k_{n+1} kernel, where $n = 1$, $\ell = 0.5$ and $\nu = 2.5$. In Figure 4.5b, seven sampled functions are displayed, including the source function represented by the yellow line. As the functions distance themselves from the source function, the line progressively transitions to a shade of blue. The 3D visualization in Figure 4.5c depicts the d dimension, visually representing how the relatedness between the source function ($d = 0$) and the target functions diminishes as d increases.

4.3 Evaluation of Deep Transfer Learning for Surrogate Models

This section presents the results of an investigation into the performance of deep TL approaches for knowledge transfer among related NN-based surrogate models. The surrogate models are constructed using synthetic datasets sampled from realizations of a GP with the k_{n+1} kernel. The input space for the domains has two dimensions ($n = 2$), defined as $\mathcal{X} = [-2, 2]^n$. The parameters for the Matern kernel are $\sigma^2 = 1$, $\ell = 0.5$, and $\nu = 2.5$ as standard, which are reasonable values considering the input range and the properties of the Matern kernel.

Ten different realizations were performed to conduct each analysis, resulting in ten distinct three-dimensional functions $f(x_1, x_2, d)$. Each of these functions were sliced at nine different d positions, yielding nine two-dimensional functions: one source and eight targets. The distances d between the functions range from 0 (source domain) to 1.55, corresponding to correlations between the source and target tasks ranging from 1.0 to 0.30 (Table 4.1). An example of one realization of the GP at four different cross-section domains is illustrated in Figure 4.6.

The chapter's analyses encompass multiple scenarios each trained with four distinct NN initializations and different training data splits for added robustness. Additionally, each analysis is conducted for the ten different realizations of the test functions. Consequently, the final results are derived from the mean of four training instances for each of the ten function realizations. This approach aims to enhance the statistical significance of the results by considering multiple training runs and

Table 4.1: Synthetic domains created for each realization of the GP.

		Source Domain	**Target Domains**						
Distance from Source	d	0	0.2	0.39	0.58	0.78	0.97	1.16	1.36	1.55
Correlation with Source	k_d	1	0.98	0.93	0.84	0.74	0.63	0.51	0.40	0.30

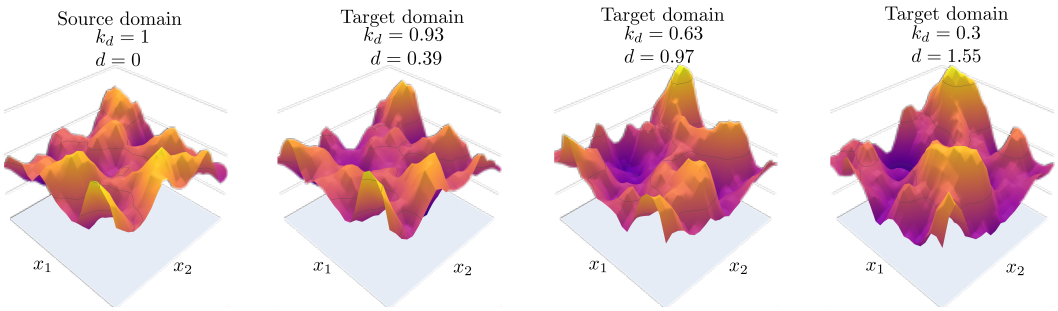


Figure 4.6: Bi-dimensional examples of the Source domain and related Target domains generated with $n = 2$, $\ell = 0.5$, and $\nu = 2.5$.

realizations for a more reliable assessment.

For each analysis, the source training dataset consists of $N_S = 1000$ samples, while the target training datasets are limited to $N_T = 50$ samples, both sampled using LHS. Twenty five percent of each training dataset is set aside as validation dataset. The NN models are implemented using the *PyTorch* library with a fully connected architecture. The Mean Square Error (MSE) is the evaluation function to assess model performance, and early stopping is applied to prevent overfitting based on the MSE computed on the validation set. A separate test dataset containing 100 samples is used to evaluate the performance of each approach. The final assessment of each NN model is derived by calculating the mean prediction error across the ten realizations.

A grid search was conducted to determine the best NN hyperparameters, maintaining a constant reference learning rate of 0.001 for NN training. The best-identified hyperparameters for the source model were 7 layers and 32 neurons per hidden layer, resulting in 6465 parameters. This configuration is used for all TL approaches. In contrast, the target model, trained without TL and with limited data, achieved optimal results with 3 layers and 32 neurons per hidden layer, totaling 2241 parameters. The discrepancy in dataset sizes led to a significant difference in NN complexity between the source and target models, highlighting that larger datasets allow for learning more parameters without overfitting.

4.3.1 Viability and Comparative Study of Deep TL Approaches

This section investigates the benefits of employing deep TL with surrogate models of related functions and evaluates the performance of different TL approaches. The accuracy of Finetuning, Gradual Finetuning, Frozen, and Unfrozen TL configurations in the target domain are compared with a baseline model without TL to assess the TL approaches' effectiveness. In the Finetuning and Unfrozen TL models, a factor of 0.1 is applied to the reference learning rate. In the Gradual Finetuning model, this factor increases incrementally from 0.1 to 1 as the NN depth increases. For Frozen and Unfrozen configurations, the three top layers of the source model are restarted with random parameters.

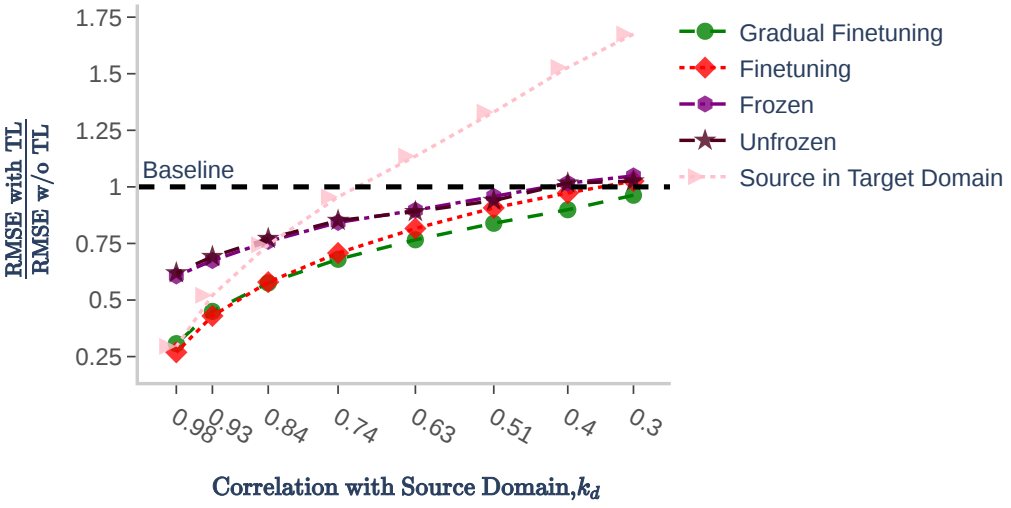


Figure 4.7: Relative reduction on prediction error due to different TL approaches.

The comparison outcomes are presented in Figure 4.7. All TL approaches reduce the prediction error compared to the baseline without TL. Both Finetuning and Gradual Finetuning TL approaches outperform the Frozen and Unfrozen methods. Among them, Gradual Finetuning exhibits the overall best performance. In scenarios where the source and target domains are closely related, such as $k_d = 0.98$, leveraging knowledge from the source domain reduces the error to only 25% of the baseline error, even when utilizing the Source model without finetuning.

As the correlation between the source and target tasks decreases, the magnitude of error reduction diminishes. However, even with a small correlation like $k_d = 0.3$, the Gradual Finetuning approach still achieves lower error than the baseline. Figure 4.8 depicts the results of Gradual Finetuning, including standard deviation, indicating the statistical significance of the improvement.

A comparison is performed using solely the target dataset for training both the source and target models to validate that the improvement is due to the transfer of knowledge and not merely the difference in training steps. The results in Figure 4.9 indicate no significant improvements, confirming that the improvement observed in Figure 4.7 are a result of knowledge transfer from the source domain. Therefore, this section establishes the effectiveness of deep TL approaches for knowledge transfer among related surrogate models with tabular data and conditional shift.

4.3.2 Impact of TL Hyperparameters on Transfer Performance

In this section, we explore the impact of TL hyperparameters on TL performance, aiming to offer practical insights for parameter selection and a more comprehensive

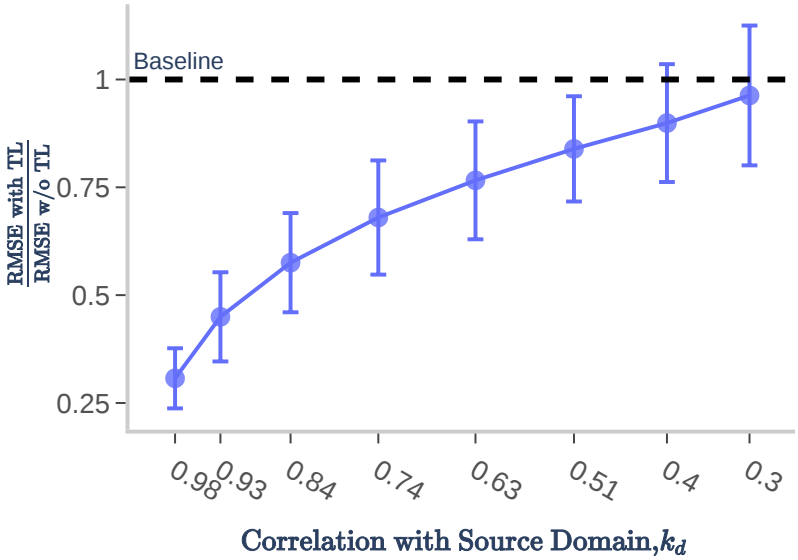


Figure 4.8: Relative error reduction, accompanied by standard deviation, achieved through the Gradual Finetuning approach.

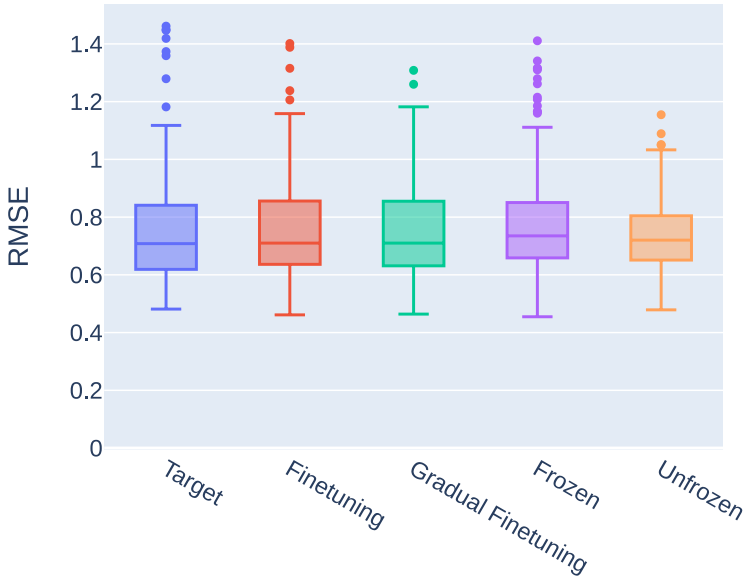
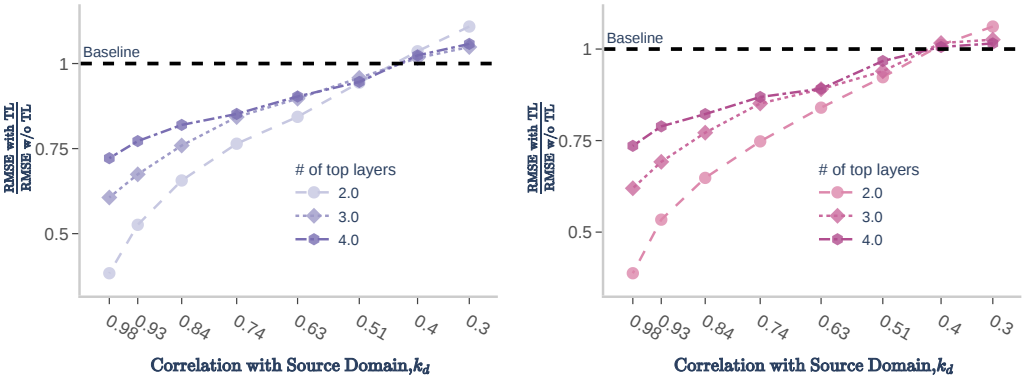


Figure 4.9: Validation results using TL approaches with the target dataset for both pre-trained and final model training.

understanding of TL mechanisms. Our investigation delves specifically into the learning rate factor within the contexts of Finetuning and Gradual Finetuning, as well as the influence of the number of top layers in Frozen and Unfrozen TL approaches. Throughout the analyses, we maintain a constant NN architecture to isolate the effects.

The effects of the number of top layers in the feature extraction TL approaches



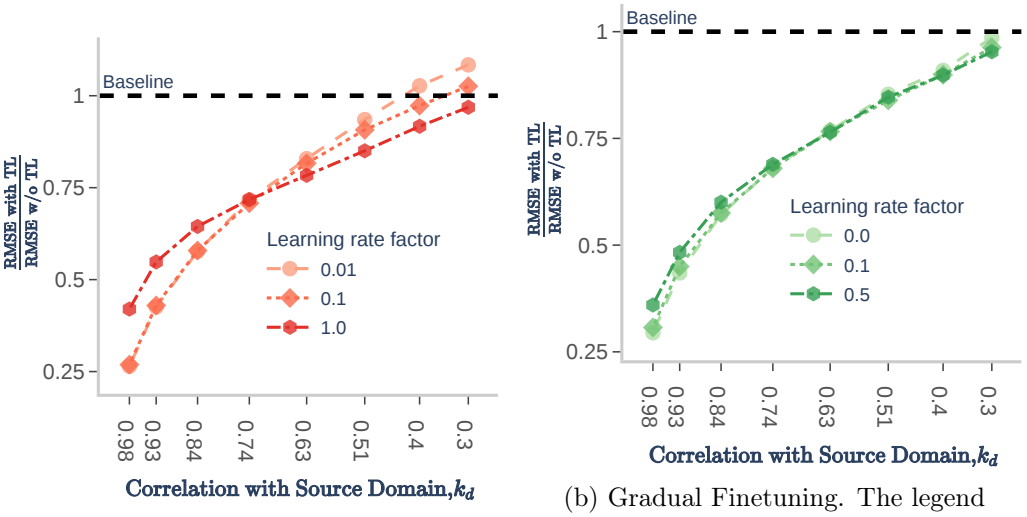
(a) Frozen TL.

(b) Unfrozen TL.

Figure 4.10: Influence of TL hyperparameters in the approaches based in feature extraction. The legend indicates the number of top layers restarted randomly.

are depicted in Figure 4.10. The outcomes suggest that restarting fewer layers is advantageous when the domains exhibit close relatedness. Nevertheless, the effectiveness of the finetuning methods continues to be superior.

Figure 4.11 displays the outcomes concerning finetuning approaches, revealing that the optimal learning rate factor increases as the correlation between target and source domains decreases. This observation is justified by the necessity for the NN to undergo more substantial adaptations when source and target domains are weakly correlated. Additionally, the results reveal the reduced sensitivity of Gradual Finetuning to the choice of the learning rate factor compared to standard Finetuning.



(a) Finetuning TL. The legend indicates the learning rate factor for the entire NN.

(b) Gradual Finetuning. The legend shows the learning rate factor of the first layer.

Figure 4.11: Influence of TL hyperparameters.

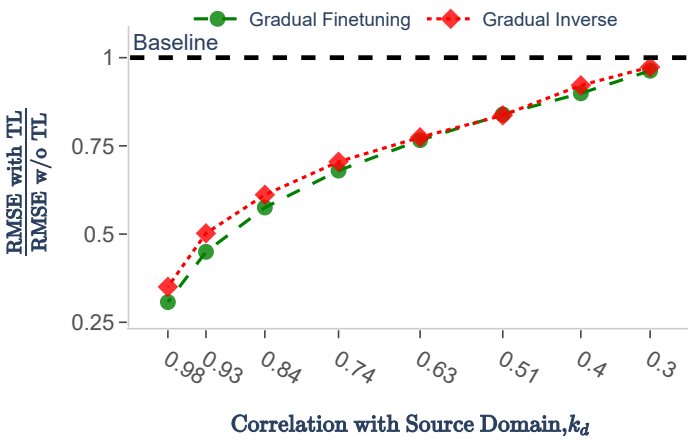


Figure 4.12: Comparison of Gradual Inverted TL and Gradual Finetuning with decreasing and increasing learning rate factors across NN depth, respectively.

An additional aspect of our investigation aims to explore the potential link between the superior performance of Gradual Finetuning and the depth-dependence of the learning rate factor. This exploration is grounded in the belief that domain-general representations primarily emerge in the bottom layers of the NN.To address this, we evaluated the performance of a Gradual Inverted TL, wherein the learning rate factor progressively decreases from 1 to 0.1 with increasing NN depth, in contrast to the incremental increase observed in Gradual Finetuning. The results, as depicted in Figure 4.12, indicate comparable performance between the two approaches. This outcome suggests that domain-general representations may not predominantly manifest in the bottom layers of NN-based surrogates, contrary to prior findings in the context of high-dimensional classification problems [283].

Consequently, the superior performance of Gradual Finetuning over conventional Finetuning appears tied to a good choice of learning rate factors but without exhibiting a dependence on NN depth. The gradual change in the learning rate factors seems to facilitate a balanced compromise between retaining previous knowledge and adapting to new information across different correlation degrees. Therefore, Gradual Finetuning is the preferred choice for the subsequent TL analyses.

4.3.3 Effectiveness of Transfer Learning across Varied Target Dataset Sizes

To evaluate the effectiveness of TL under diverse target dataset sizes, we varied the number of training points in the target domain from 25 to 200 while maintaining a constant 1000 training points in the source domain.

The results shown in Figure 4.13 reveal an expected reduction in absolute prediction error as the number of training points in the target domain increases, a trend evident in both models with and without TL. Notably, the influence of the target dataset size on NN prediction accuracy is more pronounced in the model without

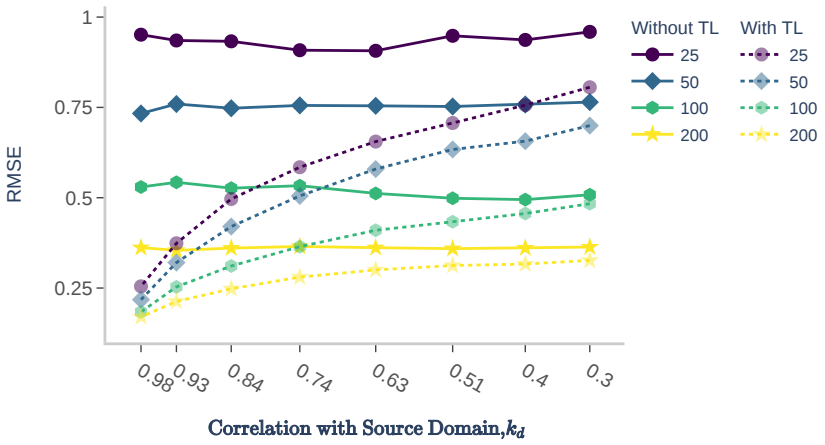


Figure 4.13: Absolute prediction error with increasing target dataset size for the models with and without TL.

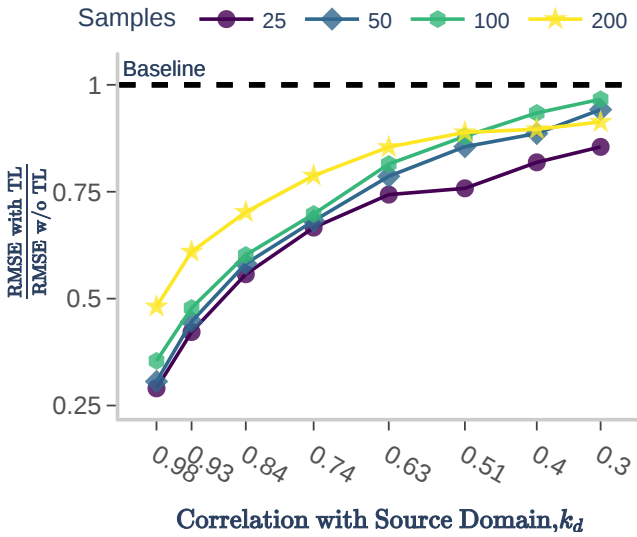


Figure 4.14: Relative improvement in prediction accuracy due to TL with different target dataset sizes.

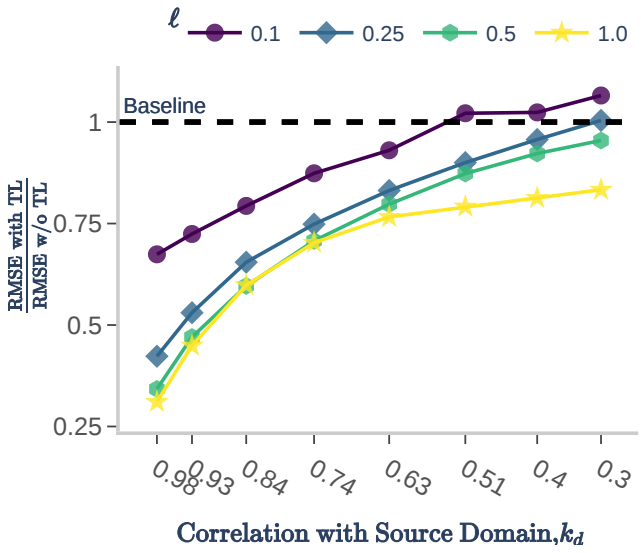
TL, as it relies solely on the information from the target dataset. This difference is particularly evident when compared with TL models in highly correlated target and source domains.

This disparity leads to a diminishing relative improvement in prediction accuracy with TL as the number of training points in the target domain increases, as illustrated in Figure 4.14. Nevertheless, TL consistently proves effective in enhancing accuracy across all tested scenarios, even when a reasonable amount of data is

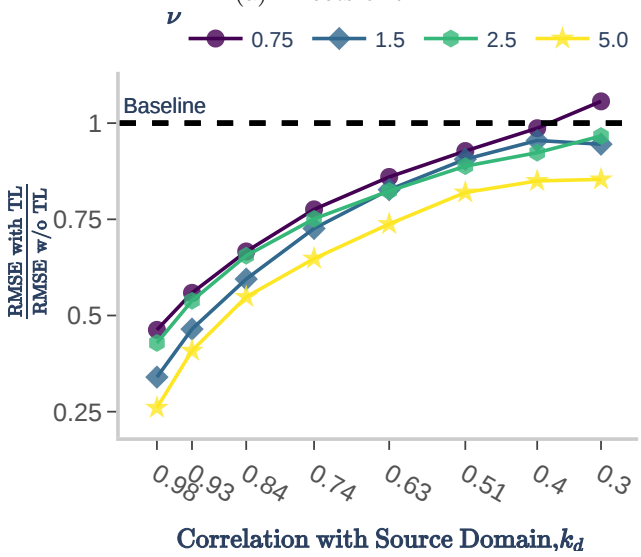
available in the target domain.

4.3.4 Assessment of the Robustness of TL to Domain Smoothness

In the previous sections, we demonstrated that TL effectively leverages knowledge from related surrogate models, improving data efficiency. This section extends our investigation to evaluate TL's robustness across varying degrees of function smoothness—an essential consideration for its applicability in SD&V.



(a) Effects of ℓ.



(b) Effects of ν.

Figure 4.15: Effect of function smoothness on TL error reduction.

To explore the impact of function smoothness, we varied the parameters ℓ and ν of the Matern kernel used to generate the domains. Specifically, we considered

four distinct values for ℓ: 1, 0.5, 0.25, and 0.1, while maintaining a fixed smoothness parameter of $\nu = 2.5$, allowing us to observe how TL responds to changes in the spatial characteristics of the function. Figure 4.15a shows that TL exhibits greater improvement for functions with slower variations (small ℓ).

Furthermore, we investigated the influence of the smoothness parameter ν, varying it among values of 5, 2.5, 1.5, and 0.75, while keeping the length scale parameter fixed at $\ell = 0.5$. The results of this analysis are illustrated in Figure 4.4, from which we can conclude that TL performs better when the tasks exhibit smoother behavior (larger ν). These findings suggest that TL benefits from spatial coherence in smoother functions, which may lead to improved source domain performance and, thus, more effective knowledge transfer.

While TL performed better for smoother functions, it also demonstrated noteworthy reductions in prediction errors for highly non-smooth functions. This conclusion underscores TL's potential benefits in scenarios characterized by erratic or complex behavior, as is often encountered in the context of SD&V surrogate tasks.

4.4 Conclusion and Discussion

This chapter highlights the effectiveness of deep TL for knowledge transfer among related NN-based surrogate models in DT prototypes. The analysis encompassed different deep TL approaches, including Finetuning, Gradual Finetuning, Frozen, and Unfrozen configurations. Gradual Finetuning consistently outperformed other approaches, demonstrating its adaptability across scenarios with varying correlation degrees between the source and target domain. The analysis extended to TL hyperparameters, emphasizing Gradual Finetuning's robustness to the learning rate factor.

Our investigation also covered TL's performance across diverse target dataset sizes, highlighting a decrease in the reliance of new data in the target domain due to TL. Furthermore, TL consistently enhanced accuracy, even when relatively more data was available in the target domain. Additionally, TL exhibited robustness across varying degrees of function smoothness, making it applicable to non-smooth SD&V problems.

In summary, the leverage of source-domain data through TL effectively improves NN-based surrogate accuracy, addressing a research gap in the under-explored TL scenario involving tabular data and conditional shift. This study positions TL as a promising tool to enhance DT prototype continuity and evolution across product generations.

Chapter 5

Digital Twin Framework: Development and Industrial Application

In this chapter, we build upon the data-based methods explored in previous chapters to develop a comprehensive DT framework for product design in SD&V. Our primary objective is to seamlessly integrate these techniques, making the DT framework readily applicable in industrial settings. Furthermore, we explore the interpretability of the framework's models for guiding decision-making processes and explore its capacity to harness historical data, reducing the need for new data. The proposed framework's effectiveness is demonstrated through an industrial application: the design of a lightweight gear transmission with a focus on NVH performance, and the transfer of knowledge to the next-generation hybrid transmission. Our ultimate goal is to provide a versatile tool that empowers early product design, streamlining the decision-making process with fast and informed insights.

Contents

Introduction

Careful design of SD&V systems is paramount to ensuring safety, energy efficiency, and comfort across various industrial applications. Consequently, the accurate and efficient modeling of SD&V systems has become the subject of extensive and long-standing research. Recently, virtual representations of physical systems or products merging physics- and data-based models, known as DT prototypes, have emerged as promising tools to facilitate more efficient and reliable system evaluations and product design. These DT prototypes offer numerous benefits, including reduced development cycles and improved decision-making processes. However, implementing such DT prototypes effectively and efficiently requires a consolidated methodology yet to be established. We now want to challenge the hypotheses formulated in this thesis and leverage the concepts, methods, metrics, and frameworks developed in the past chapters by tackling a complex and realistic design problem in SD&V. The proposed framework tackles ML-based DT's challenges in SD&V related to non-smooth behavior, interpretability, and data scarcity.

To demonstrate the capabilities of our DT framework, we apply it to a significant problem in the SD&V domain: the multi-objective optimization of a gear transmission under design uncertainties. Gear transmissions play a critical role in various machines across diverse sectors, and their design is complex due to numerous variables and conflicting objectives, such as efficiency, weight reduction, noise and vibration attenuation, and structural reliability. Moreover, the performance of gear transmissions depends on complex phenomena involving multi-scale, multi-physics, coupled, and non-linear effects.

The complexity and significance of the gear transmission design challenge have led to numerous studies addressing it [285–290]. However, most of these studies have been limited by computational limitations, preventing the consideration of all relevant aspects simultaneously. For instance, although many studies optimized gear transmissions concerning their macro and micro parameters, most of them have done so separately, disregarding potential multi-scale interactions. Moreover, the effects of uncertainties are often neglected during optimization. Garambois et al. [291] addressed these aspects, but their approach relied on a simplified model of the gear tooth and disregarded the housing flexibility effects.

In this chapter, we demonstrate how the proposed DT framework can be applied to perform well-informed and inexpensive virtual prototyping to address these limitations and enhance the state-of-the-art SD&V design. To illustrate its real-world applicability, we apply the DT framework to an optimization study of the 7DCT300, a 7-speed dual clutch transmission developed by Magna International for use in combustion vehicles. Furthermore, the DT transfers the insights gained from the 7DCT300 to improve the design of the 7HDT300, a derivative transmission designed for hybrid vehicles, illustrating the applicability of the inter-generational DT in industrial scenarios.

The key contributions of this chapter are two-fold:

1. Integration of design, ML, and TL techniques to propose an efficient DT prototype framework, which includes:
   - Incorporating domain knowledge to enhance NN models using PGF.
   - Implementing two levels of surrogate model fidelity for exploratory and final analyses in the design process.
   - Utilizing surrogate-based GSA for feature selection, informed optimization strategy, and improved NN model interpretability.
   - Applying innovative deep TL to leverage past product data for data-efficient surrogate training.
2. Demonstration of the DT prototype construction in gear transmission design, leading to:
   - A streamlined optimization process.
   - Original multi-objective chance constrained gear transmission optimization using a fully coupled model and considering micro and macro parameters with manufacturing uncertainties.
   - Investigation of micro and macro geometry interactions.

This chapter is structured as follows: Section 5.1 provides a concise overview of the proposed DT framework and its key steps. Section 5.2 presents the gear transmission design methodology, assumptions, and design parameters. In Section 5.3, we implement a NN-based low-fidelity surrogate model using physics-guided features, followed by a GSA to define important variables and optimization strategy. Section 5.4 constructs a high-fidelity DT model with reduced dimensionality, which is employed in the multi-objective optimization of the gear transmission design, using chance constrained optimization and Genetic Algorithms. In Section 5.5, the DT prototype is extended to the hybrid transmission with limited data. Finally, Section 5.6 summarizes the chapter's key contributions and highlights advancements in virtual product design in the field of SD&V.

5.1 Methodology Overview of the DT Framework

The construction of the proposed DT Prototype involves several interconnected steps that culminate in a flexible, interpretable, and data-efficient model capable of replicating the behavior of a physical system. The workflow of the DT framework consists of the six steps summarized below:

Step 0: Construction of the physical problem's true model $f(\cdot)$.

The initial step involves creating the ground 'true' model, denoted as $f(\cdot)$, by implementing an accurate physics-based numerical simulation. This simulation serves as the foundation for the DT and is used to sample supporting points used for the training of the ML algorithm.

Step 1: Construction of the low-fidelity NN-based surrogate $h_L(\cdot)$.

In this step, a low-fidelity surrogate, denoted as $h_L(\cdot)$, is constructed to explore the system behaviour in the design space $\mathcal{X}_L \in \mathbb{R}^{n_L}$, where n_L is the number of

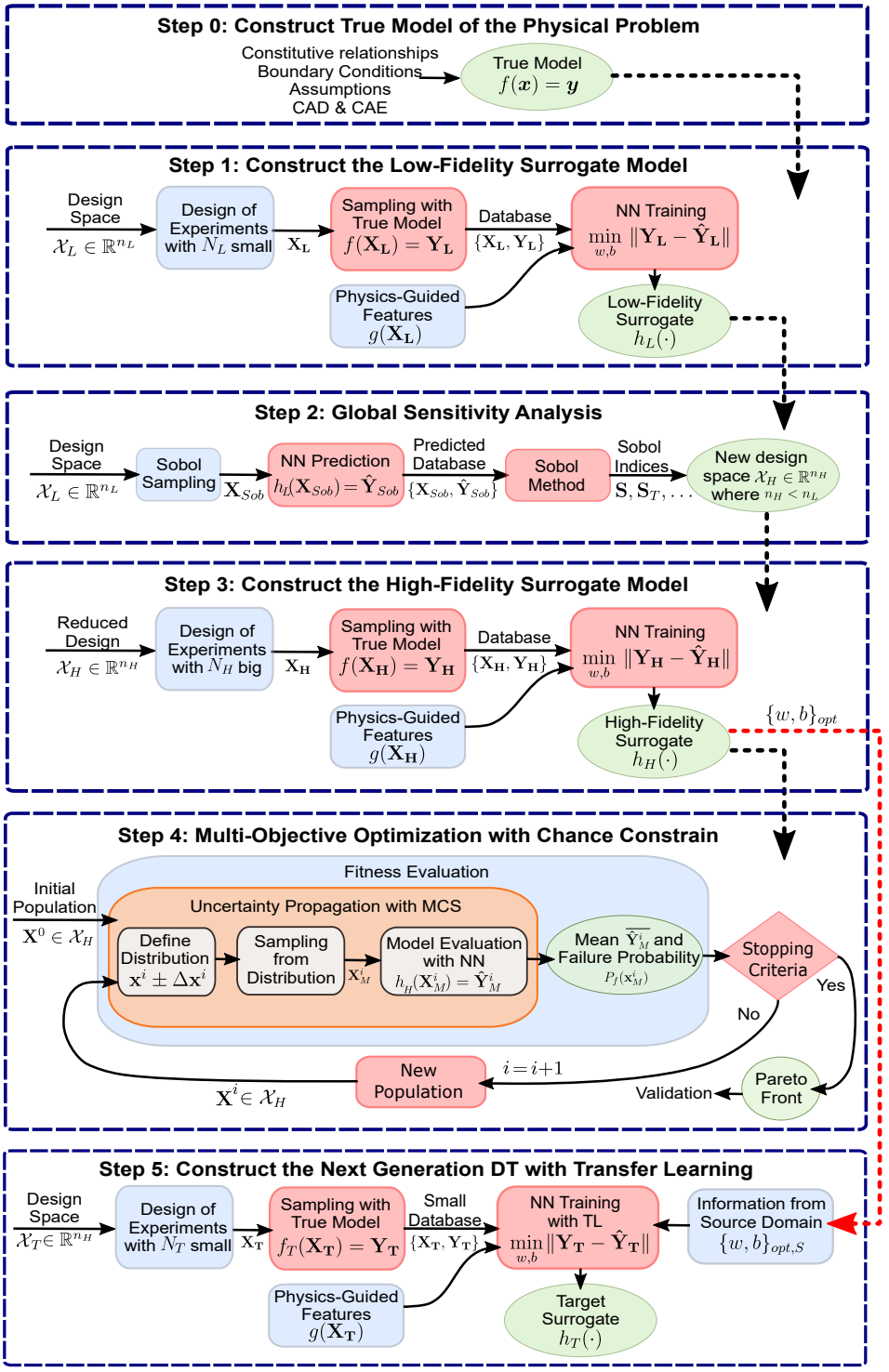


Figure 5.1: Workflow for constructing the surrogate model of the gear transmission and the multi-objective optimization with reliability constraint.

parameters considered. The NN-based model $h_L(\cdot)$ is trained using a small dataset, $\{X_L, Y_L\}$, consisting of N_L supporting points sampled from the true model $f(\cdot)$. To enhance the accuracy of the low-fidelity surrogate, physics-guided features $g(X_L)$ are incorporated, improving the model's data efficiency and physical consistency.

Step 2: Global sensitivity analysis and reduced design space $\mathcal{X}_H$.

The low-fidelity surrogate model $h_L(\cdot)$ is leveraged in this step to perform a fast GSA with the Sobol method. This analysis reveals the most critical variables in the system. Additionally, Sobol's indices assess the interactions among the input variables, allowing for the definition of independent variables that can be optimized separately. Therefore, this step aids in comprehending and simplifying the problem, thereby reducing its dimensionality and guiding the selection of an appropriate optimization strategy. A reduced design space, $\mathcal{X}_H \in \mathbb{R}^{n_H}$ with dimension $n_H < n_L$ can, thus, be defined.

Step 3: Construction of the high-fidelity surrogate model $h_H(\cdot)$.

In Step 3, a high-fidelity NN-based surrogate model $h_H(\cdot)$ is trained with a more extensive dataset X_H containing $N_H > N_L$ supporting points sampled from the reduced design space $\mathcal{X}_H$ using the true model $f(\cdot)$. Similar to Step 1, physics-guided features $g(X_H)$ are incorporated to enhance the accuracy of the high-fidelity surrogate.

Step 4: Multi-objective optimization with chance constraint.

Step 4 involves utilizing the high-fidelity surrogate model $h_H(\cdot)$ to perform a fast multi-objective optimization under uncertainties. For each design considered in the optimization process, uncertainty propagation with MCS is performed to account for the input uncertainties. Statistical moments, as the response mean, and reliability criteria, as the probability of failure, can then be used in the objective functions and constraints to ensure robustness for the optimization process. The optimization iterates until convergence, resulting in the robust Pareto Front. The multi-objective optimization with chance constraint is a relevant use of the developed model, as it is an important procedure to achieve an optimal and robust design, but that involves several evaluations of the system. Therefore, the DT is critical to enable this procedure with reasonable computational costs. The optimization presented is only one of the relevant possible applications, as the DT is flexible and applicable in other procedures and scenarios.

Step 5: Construction of the next generation DT with TL.

To enable the construction of the DT prototype for the next product generation under low data availability, Step 5 employs TL, leveraging the knowledge gained from the current DT prototype. For that, the optimal NN parameters $\{w, b\}_{opt,Source}$ from the current high-fidelity surrogate model $h_H(\cdot)$ serve as a starting point for the training process of the new-generation DT $h_T(\cdot)$. The new-generation DT $h_T(\cdot)$, or target DT, is then trained with a scarce dataset $X_T \in \mathbb{R}^{n_H}$ with N_T supporting points, which are sampled from the true model of the new system $f_T(\cdot)$. This

approach enables the construction of the new-generation DT using deep learning despite data scarcity, thereby facilitating the development of an efficient DT for future iterations and giving continuity to the insights gained during the current product design.

Figure 5.1 schematizes the steps in the workflow of the DT framework. The following sections describe each step and its results for the study case of the gear transmission design. Section 5.2 describes the numeric model used for the NVH analysis of the gear transmission (Step 0). Section 5.3 shows the methodology and results of Steps 1 and 2. Section 5.4 presents the optimization procedure methodology and the results of Steps 3 and 4. Finally, Section 5.5 applies TL to improve the DT of the new hybrid gear transmission using the information acquired from the combustion gear transmission DT.

5.2 Industrial Application: Gear Transmission NVH Analysis

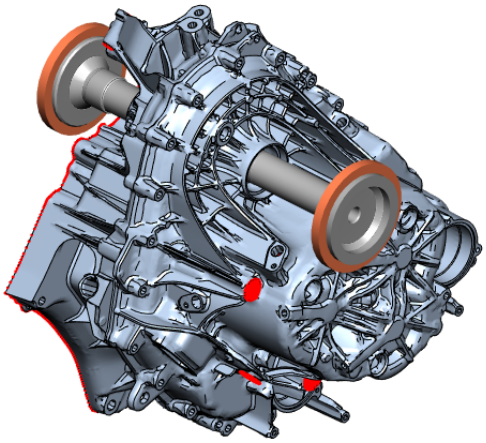
This section provides an overview of the numerical model implemented in Masta [292], aimed at simulating the NVH performance of a gear transmission. The methodology is applied to construct the numeric model for two related automotive gear transmissions: the 7DCT300 and the 7HDT300 designs, both developed by Magna International for combustion and hybrid vehicles, respectively.

Figure 5.2 shows the Masta model for both designs. Comparing the two, the 7HDT300 model incorporates an electric motor and an additional gear pair, distinguishing it from the 7DCT300. Consequently, the housing undergoes modifications to accommodate these new components, thereby altering the overall system stiffness and dynamic response.

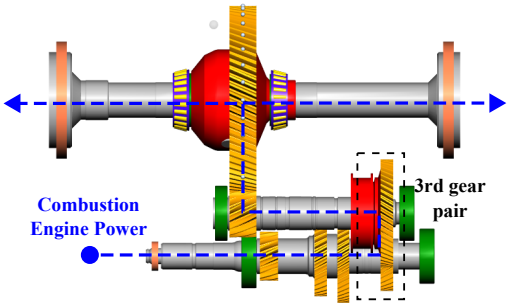
For our analysis, we focus on the third gear pair, highlighted in Figure 5.2, to assess its NVH performance under various load conditions, specifically at input torques of $20, 40, 60, 80, 100$, and $120 Nm$, while considering its macro and micro-geometric parameters. In addition to the numerical model details, this section also encompasses the design parameters and the designated design space under consideration.

5.2.1 Numerical Model

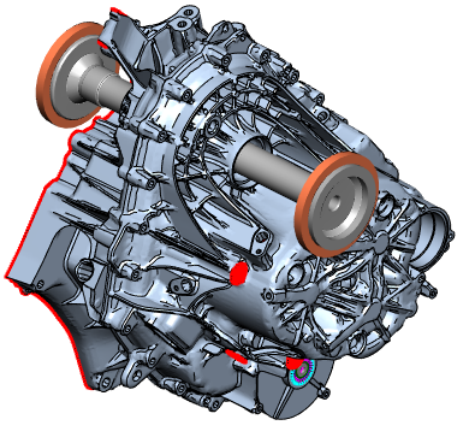
Gear whine is a significant NVH issue in gear transmissions, characterized by tonal and annoying noise originating from the gear meshing process. The main cause of gear whine is the Static Transmission Error (STE), defined as the difference between the actual contact position of the output teeth and its theoretical position in a perfectly conjugated contact [293]. As the gear rotates, the STE fluctuations act as excitations at the meshing level. These vibrations propagate through gears, shafts, and bearings until reaching the housing, where the vibrations lead to the undesirable airborne noise, known as gear whine. To mitigate this NVH issue during



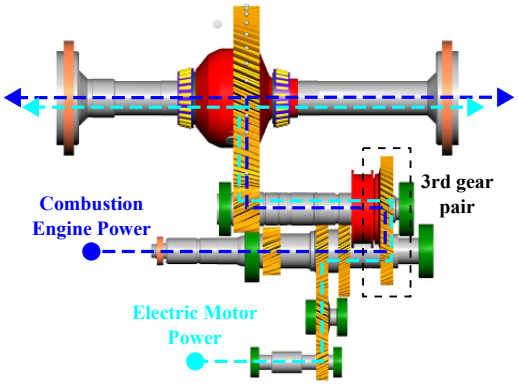
(a) Combustion gear transmission model, 7DCT300.



(b) Power paths of the 7DCT300 and gear pair under analysis.



(c) Hybrid gear transmission model, 7HDT300, successor of 7DCT300



(d) Power paths of the 7DCT300 and gear pair under analysis.

Figure 5.2: Industrial gear transmissions under analysis.

gearbox design, we focus on minimizing the amplitude of the source excitation, the Peak-to-peak Transmission Error (PPTE).

Accurate calculation of PPTE requires a fully coupled model of the gear transmission, considering the system's elastic deflection that influences mesh misalignments and impacts the contact analysis [294]. Therefore, the Masta model incorporates gears, shafts, bearings, and the housing, modeled with FEM (Figure 5.2). Specifically, the gears and housing are modeled as 3D FE meshes for accuracy, and dynamic reduction is applied at the connecting points, such as the gear flanks and bearings, to reduce computational costs while accounting for dynamic behavior. The shafts are modeled using Timoshenko beam elements. The bearings are represented with a fully populated [5x5] stiffness matrix that considers coupling among

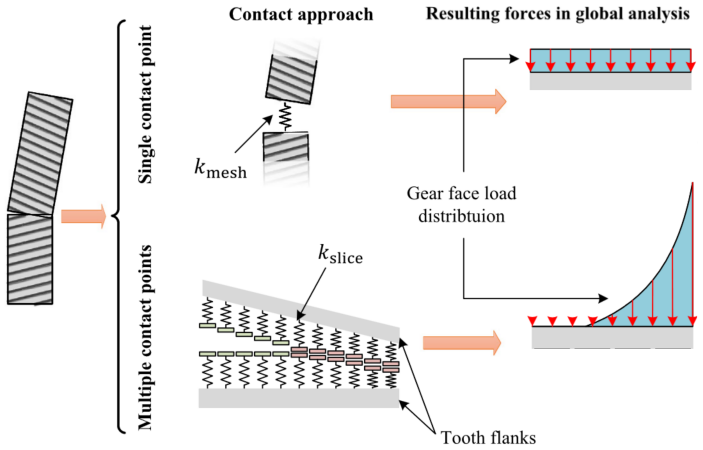


Figure 5.3: Illustration of the tooth contact model with single and multiple contact points and the resulting load distributions. Adapted from [295].

translational and rotational displacements [294]. In view of that, the non-linear behavior in rolling bearings arising from factors like non-linear stiffness, Hertzian contact stresses, and clearance is approximated by a linearized behaviour near the operational condition.

The STE evaluation initially involves a static analysis to assess the fully coupled system's deflection. However, this analysis depends on the mesh stiffness k_m, which can be derived from the gear set macro-geometry, according to ISO 6336, or can consider the specified gear micro-geometry, as adopted for the model implemented in this chapter. Given the mesh stiffness, the system deflection analysis provides critical information, including mesh misalignment.

Next, the STE Δ as a function of the input angular position ψ is evaluated using the Advanced Loaded Tooth Contact Analysis (LTCA) module in Masta. The LTCA applies multiple contact points by distributing the static force transmitted F among strips i in the contact line. The multiple contact points approach is largely used for gear contact analysis as it allows to evaluate the load distribution along the tooth face and account for the actual microgeometry of the gear profile [293, 295]. Figure 5.3 illustrated the difference between single and multiple contact point analyses. In the Advanced LTCA, a full 3D FE model of the gears is combined with Hertzian contact equations, to provide accurate and variable mesh stiffness, deformation, and stress along the tooth face.

Then, at each angular position ψ, the static load distribution vector $\boldsymbol{p}$ is determined simultaneously with the STE in an iterative process. The resulting system of equations is represented as:

$$\begin{cases} \mathbf{C}(\psi)\mathbf{p}(\psi) = \Delta(\psi) - \mathbf{e}(\psi) \\ \sum_i \boldsymbol{p_i}(\psi) = \mathbf{F} \end{cases}, \tag{5.1}$$

subject to the constraint that the contact is closed:

$$\begin{cases} \mathbf{C}(\psi)\mathbf{p}(\psi) + \Delta(\psi) >= \mathbf{e}(\psi) \\ \mathbf{p_i} >= 0 \end{cases} . \tag{5.2}$$

Here, $\boldsymbol{e}$ is the vector containing the initial gap between the teeth at each discrete point, including the effects of the micro-geometric modifications, manufacturing errors, and gear mesh misalignment resulting from the system deflection. $\mathbf{C}$ is the tooth compliance matrix that considers tooth bending and root rotation, Hertz deformation, gear and shaft In its formulation, the Advanced LTCA model assumes constant mesh misalignment, neglects friction in the contact zone, disregards lubricant effects, and assumes contact lines remain in the plane of action [292]. bending, and bearing deflection. The resulting problem is non-linear since it involves engaging and disengaging contact, with variable number of teeth and contact position, besides the non-linear Hertzian deformation.

The fluctuations in STE lead to a mesh stiffness fluctuation $k(\psi)$, which can now be more accurately evaluated as the derivative of the transmitted load relative to the STE [291]:

$$k_m(\psi) = \frac{\partial F}{\partial \Delta(\psi)}. \tag{5.3}$$

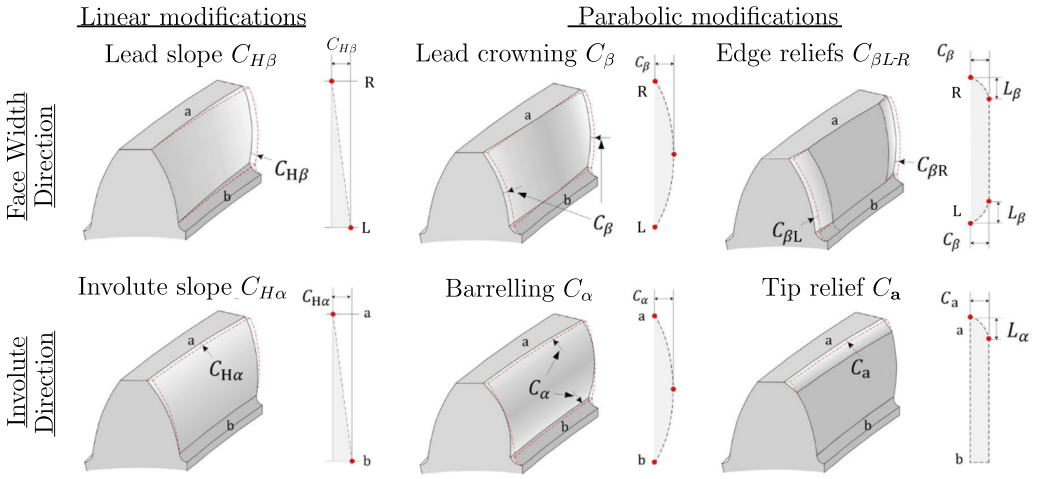
Once the numerical model was used to sample the database used for the ML training, a *Python* code was implemented to load, modify and run the Masta simulation to ease the sampling procedure. Although the focus of the DT under development is minimized PPTE, the model implemented allows analyzing other performance indicators, such as teeth root stress, pressure on the teeth edges, shaft bending, among many others, that can be included in the DT construction. Finally, as a subsequent step, one can perform harmonic analysis to evaluate the housing vibration level and the resulting sound pressure.

5.2.2 Design Parameters and the Design Space

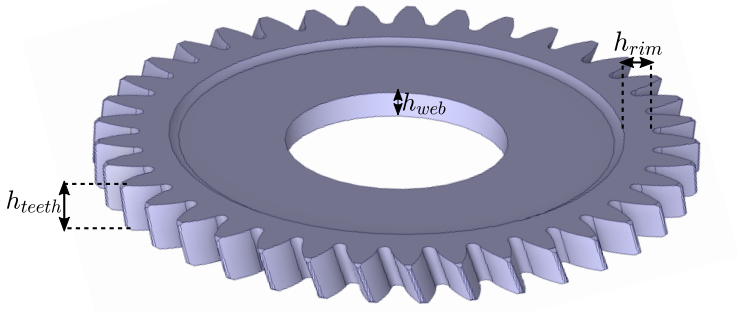
The performance of the third gear pair of 7DCT300 and 7HDT300 is analyzed concerning its macro and micro-geometries as design variables. We consider three adjustable macro-parameters (Figure 5.4b): the pinion web thickness h_{web}, the pinion rim thickness h_{rim}, and the teeth face width h_{tooth}, which is kept equal for pinion and wheel. These parameters influence the gear stiffness and, consequently, the PPTE, but also impact the gear pair mass, and should be adjusted to present a trade-off. The remaining macro parameters are kept fixed: number of teeth are $Z_P = 2$ and $Z_G = 2$; module equals $2mm$; pressure angle is 20°; helix angle is 20°; addendum coefficient is 1; dedendum coefficient is 1.25; gears material is steel. The pinion and gear values are the same.

Furthermore, we consider a comprehensive set of micro-geometry parameters for both the pinion and wheel gears. These parameters include modifications to the gear tooth profile in the face width axis, such as lead crowning C_β, lead slope $C_{H\beta}$,

left edge relief $C_{\beta L}$, and right edge relief $C_{\beta R}$. Additionally, we modify the involute profile of the teeth with the barrelling C_{α}, involute slope $C_{H\alpha}$, and tip relief C_{a}. Figure 5.4a illustrates these modifications and their geometric definitions can be found in ISO 21771 and ISO 1328-1.



(a) Micro-geometry modifications of gear tooth profile. Adapted from [295].



(b) Macro-geometry parameters.

Figure 5.4: Gear Parameters.

Lead crowning and barrelling serve essential purposes, ensuring a continuous contact pattern with improved load distribution, and reducing edge contact stress [296]. Moreover, crowning and barrelling enhance the gear's robustness against manufacturing errors and misalignments [295]. Slope modifications impact gear engagement, load distribution, and effectively compensate for mesh misalignment [295]. Tip and edge reliefs promote smooth gear engagement, minimizing the risk of tooth-to-tooth interference, and also increase the gear's tolerance to misalignments [296].

Therefore, in total, seventeen design variables are considered ($n_L = 17$), being three macro-geometry parameters, and seven micro-geometry parameters for each gear of the gear pair. The initial design space $\mathcal{X}_L$ is shown in Table 5.1 and was chosen according to typical micro- and macro-geometry values in automotive indus-

try.

Table 5.1: Design space $\mathcal{X}_L$ of the gear teeth micro and macro-geometries. The micro-geometry ranges apply to both pinion and gear.

	Micro-geometry for Pinion and Wheel							**Macro-geometry**		
	C_β $[\mu m]$	$C_{H\beta}$ $[\mu m]$	$C_{\beta L}$ $[\mu m]$	$C_{\beta R}$ $[\mu m]$	C_α $[\mu m]$	$C_{H\alpha}$ $[\mu m]$	C_a $[\mu m]$	h_{tooth} $[mm]$	h_{web} $[mm]$	h_{rim} $[mm]$
Min	0	-20	0	0	0	-20	0	12	7	3
Max	15	20	25	25	15	20	40	18	12	6

5.3 Exploratory Low-Fidelity Surrogate with Physics-Guidance and Interpretability

This section carries out the construction and employment of physics-guided NN-based low-fidelity surrogate $h_L(\cdot)$ of the combustion gearbox model for exploratory purposes. Section 5.3.1 presents the effects of embedding basic domain knowledge during training in the surrogate accuracy. Section 5.3.2 employs the low-fidelity model to perform GSA with Sobol method and provide an deeper understanding of the system response. The sensitivity results guide the choice of the reduced design space $\mathcal{X}_H \in \mathbb{R}^{n_H}$ with dimension $n_H < n_L$ and of the optimization strategy.

5.3.1 Low-fidelity surrogate training and effects of embedding physical guidance

The training of the surrogate model follows the methodology presented in Section 2.1. Therefore, the first step uses LHS to sample N_L input points $X_L \in \mathcal{X}_L$ with good representativeness of the design space variability. The gearbox physical model is used to obtain the corresponding output pairs, $Y_L = f(X_L)$. The PPTE at six input torques are the outputs under analysis, thus, $Y_L \in \mathbb{R}^6_+$. Subsequently, the training data is standardized and the NN model is trained to minimize the MSE between its prediction and the true output values in this training dataset.

At this exploratory stage, high-precision is not mandatory and a small number of supporting points $N_L = 50$ is used to construct the low-fidelity surrogate model. The elapsed time to sample the training dataset is of $12203s$ and to train the NN is of $20.6s$.

Knowledge about the physics of the gearbox problem can be used to improve training performance and data efficiency via PGF (see Section 2.3). Even in the absence of a known algebraic equation for the PPTE problem, the analysis of numerical solutions reveals valuable insights. Notably, certain profile modifications from the meshing teeth combine in the mesh plane, resulting in a sum of both the pinion and wheel modifications. These modifications, which include the lead slope,

crowning, left, and right edge reliefs, specifically refer to modifications in face width direction. As a result of this observation, we can sum the contributions from both the pinion and wheel sides for these four modifications during a data pre-processing stage, effectively reducing the problem's dimensionality from seventeen to thirteen dimensions. Furthermore, since the PPTE will always be positive, we utilize a $ReLU$ layer at the end of the NN to ensure that the model's output is strictly positive.

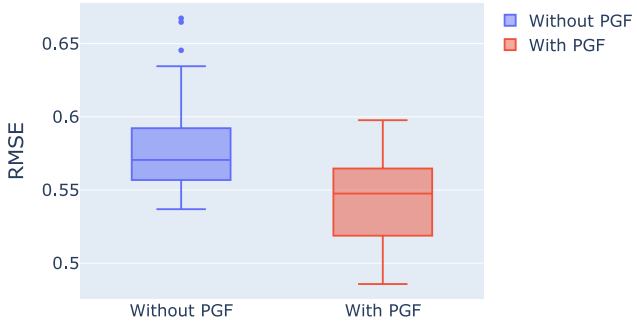


Figure 5.5: Comparison of low-fidelity surrogate prediction RMSE with and without physical guidance.

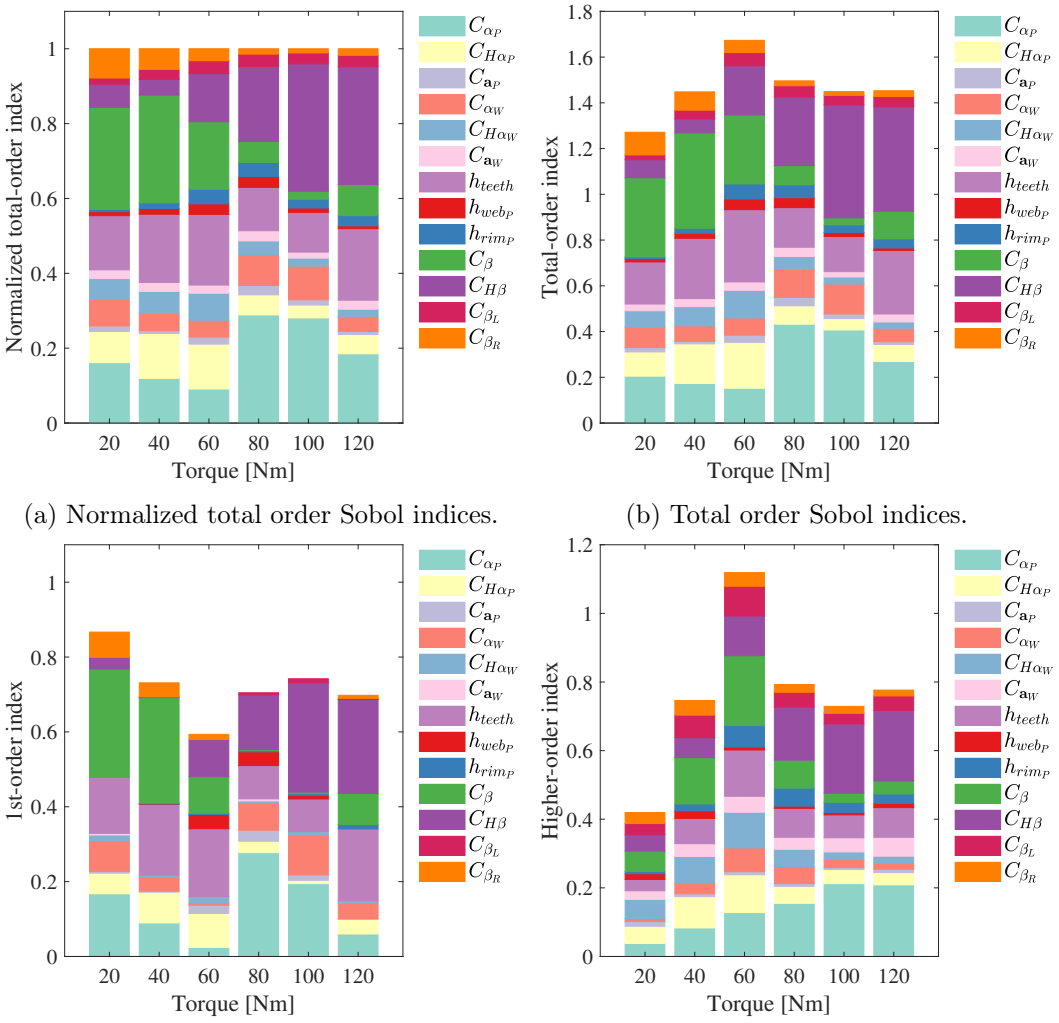
Figure 5.5 shows the comparison between NN prediction errors with and without incorporating this physical guidance. The comparison was conducted across 40 different random initializations and dataset splits to ensure statistical significance. The results reveals that the model with physical guidance exhibits slightly improved predictions with no addition of computational cost.

5.3.2 Global Sensitivity Analysis with Sobol Method

This section presents the GSA with Sobol of the gear transmission PPTE obtained, employing the NN-based low-fidelity surrogate to speed up the analysis (refer to Section 3.1.1 for Sobol methodology). The GSA improves the model interpretability and guides the DT construction by revealing critical variables and their interactions.

While the same design space as outlined in Table 5.1 is used for computing Sobol indices, the lower and upper bounds for face width modifications are doubled once the modifications from both pinion and wheel are combined together. The computational benefit of performing GSA with surrogate model is evident, with only $45s$ required to compute 28672 evaluations from Sobol analysis, whereas performing the same analysis without the surrogate would demand approximately 1943 hours.

The resulting Sobol indices are illustrated in Figure 5.6, including first-order indices $S^q_{1_i}$, higher order indices $S^q_{H_i}$, and total order indices $S^q_{T_i} = S^q_{1_i} + S^q_{H_i}$, where the subscript i indicates the input variable and superscript q indicates the load case. Figure 5.6a depicts the normalized total order Sobol sensitivity indices, and Table 5.2 presents their average values for the most important features. The results suggest that the most influential parameters, ranked in order of significance, are the pinion barreling C_{α_P}, the resulting lead slope $C_{H\beta}$, the face teeth width h_{teeth}, and the resulting crowning C_β.



(a) Normalized total order Sobol indices.

(b) Total order Sobol indices.

(c) First-order Sobol indices.

(d) Higher-order Sobol indices.

Figure 5.6: Sobol indices for the PPTE of the 7DCT300.

Exploiting the capability of Sobol analysis to discern first and higher order sensitivity indices, we investigate input interactions across diverse load conditions. As elucidated in Section 3.1.1, when total order sensitivity indices sum up to one, it implies that output variability is driven solely by individual input factors, with input interactions having negligible influence. Functions with this attribute are referred to as additive functions. For instance, an additive relation between variables x_1 and x_2 can be expressed as:

$$f(x_1, x_2) = f_1(x_1) + f_2(x_2). \tag{5.4}$$

Consequently, the optimization of functions with additive relations can be partitioned into several distinct, independent optimization sub-problems, each with reduced dimensionality.

Table 5.2: Normalized total Sobol sensitivity indices averaged across all load cases.

C_{α_P}	$C_{H\beta}$	h_{teeth}	C_{β}	$C_{H\alpha_P}$	C_{α_W}	$C_{H\alpha_W}$
0.19	0.18	0.15	0.15	0.08	0.06	0.04

Figure 5.6b illustrates total order Sobol indices across different load conditions, each with sum bigger than unity, indicating significant input interactions. Hence, an appropriate optimization strategy is selecting relevant variables for joint optimization, i.e., variables with significant interaction effects, which can be estimated by Sobol's higher-order indices. Thus, our selection criterion designates input variables whose higher-order indices contribution for output non-additive variability exceeds 7.5% in at least one load condition, that is:

$$\text{if} \quad max_q \frac{S^q_{H_i}}{\sum_i S^q_{H_i}} > 0.075, \quad \text{variable } i \text{ is selected.} \tag{5.5}$$

The selected variables and their associated higher-order indices contribution are tabulated in descending order in Table 5.3, and approximately align with the overall most important variables in Table 5.2. Thus the new reduced design space $\mathcal{X}_H \in \mathcal{X}_L$ has $n_H = 8$ dimensions. The variables that did not match the selection criterion were $C_{\mathbf{a}_P}$, $C_{\mathbf{a}_W}$, h_{web_P}, $C_{\beta R}$, and C_{α_W}.

Table 5.3: Selected variables and their respective maximum contribution for the higher-order Sobol indices.

C_{α_P}	$C_{H\beta}$	C_{β}	$C_{H\alpha_W}$	$C_{H\alpha_P}$	h_{teeth}	C_{β_L}	h_{rim_P}
0.29	0.28	0.18	0.14	0.12	0.12	0.09	0.08

Figures 5.6b, 5.6c, and 5.6d show, respectively, the total, first- and higher-order Sobol indices for different load conditions. An increase in input interactions can be noted with the increase of the input torque, likely due to increased misalignment under heavier loads. However, at the input torque of $60Nm$, a peak in higher sensitivity indices emerges together with a decrease in the first-order effects, diverging from the overall trend and indicating significant interactions at this specific load condition. The analysis of Figure 5.6c underscores that several variables have negligible individual effects, with their importance arising from interactions with other variables.

In essence, the GSA employing low-fidelity surrogate model effectively elucidates the outcomes of the PPTE analysis, providing valuable insights to simplify the design problem while retaining essential information. These insights guide the reduction of the design space to important variables that, due to their significant interactions, cannot be independently optimized. Therefore, this strategic step not only aids in comprehending and simplifying the DT, but also guides informed decision-making on the design strategy.

5.4 High-Fidelity Digital Twin for Efficient and Robust Product Design Optimization

This section introduces an efficient methodology for effective and robust product design optimization utilizing a DT prototype. Initially, the high-fidelity data-base model $h_H(\cdot)$ of the combustion transmission is trained in the reduced design space $\mathcal{X}_H$, as detailed in Section 5.4.1. This DT is then employed in a multi-objective optimization targeting the third gear pair parameters of combustion transmission. The optimization seeks to minimize the gear pair mass and PPTE under varying operating conditions, while accounting for geometric tolerances with chance constrains. Section 5.4.2 presents the gear transmission optimization formulation and outcomes. Refer to Appendix B for the methodology of multi-objective optimization with chance constrain employed.

5.4.1 High-Fidelity NN model

The high-fidelity NN-based model employs a more extensive dataset denoted as X, Y_H, comprising $N_H = 250$ data points. Input data is obtained through LHS from a reduced design space $\mathcal{X}_H$, while output values are generated by evaluating the Masta model. Dimensional reduction, using PGF and feature selection from the previous section, transforms the original $\mathcal{X}_\mathcal{L}$ with $n_L = 17$ dimensions into the reduced $\mathcal{X}_\mathcal{H}$ with $n_H = 8$ dimensions, which still presents a relatively high-dimensional challenge.

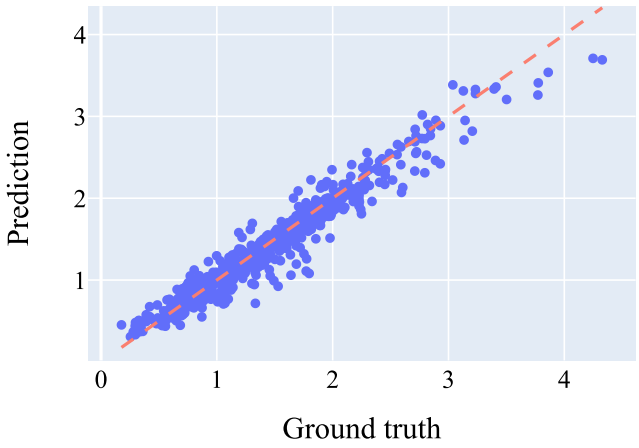


Figure 5.7: Scatter plot of the NN high-fidelity model of the combustion transmission with $RMSE = 0.17\mu m$ and $MAE = 0.12\mu m$ in the test dataset.

Implemented using *PyTorch*, a single fully connected NN predicts PPTE values for $q = 6$ load conditions. Hyperparameters for the model are optimized through grid search employing k-fold cross-validation with five folds. The hyperparameters search process yields the following optimal specifications: five hidden layers, each containing 128 neurons activated by $ReLU$, L2 regularization with strength 1×10^{-3},

and learning rate of 0.005.

The training procedure involves 32-sample batches conducted over 2500 epochs, completed in a time of $57s$. The achieved accuracy, averaging $0.12\mu m$ across all load conditions, is satisfactory for the PPTE problem. The model's efficacy is visually depicted in Figure 5.7, illustrating a scatter plot comparing the true values of the test dataset against the corresponding NN-generated predictions.

5.4.2 Gear Transmission Optimization

5.4.2.1 Problem Formulation

The NVH gear transmission multi-objective optimization with chance constrain is carried out in this chapter can be defined by its:

Decision Variables: The micro- and macro parameters of the gear pair, represented by $\boldsymbol{x} = [x_1, x_2, \ldots, x_{n_H}] \in \mathcal{X}_{\mathcal{H}} \in \mathbb{R}^n_H$.

Objective Functions: Minimize the mean of the PPTE, $\overline{\delta}(\boldsymbol{x}) = \sum_{i=1}^{M} \delta(\boldsymbol{x})/M$, at $q = 6$ load conditions, and minimize the mass of the gear pair, m_{gears}.

Constraints: The probability of failure should not exceed 5% for all load conditions analyzed, where failure is defined by $\delta >= 2\mu m$.

Formally:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}_H} \quad & f(\boldsymbol{x}) = \left[\overline{\delta_1}(\boldsymbol{x}), \overline{\delta_2}(\boldsymbol{x}), \ldots, \overline{\delta_q}(\boldsymbol{x}), m_{gears}\right] \\ \text{s.t.} \quad & P(\delta_j(\boldsymbol{x}) \leq 2\mu m) \leq 0.05, \quad \text{for } j = 1, 2, \ldots, q \end{aligned} \tag{5.6}$$

where the mean and probability of failure are evaluated with a MCS uncertainty propagation with M samples. The mass of the gear pair m_{gears} is calculated analytically and do not include the mass of the wheel web.

5.4.2.2 Uncertainty Modeling

The input uncertainties are addressed through MCS with M samples drawn from the input distributions to assess the system response in terms of both mean and the probability of failure. The input distributions are characterized by the design's nominal value, represented as the mean $\mu = x_i$, and a standard deviation σ derived from one-third of the manufacturing tolerance Δ. The input parameters with positive values are modeled using a gamma distribution, while those with both positive and negative values follow a normal distribution.

The specific input tolerances employed are documented in Table 5.4. These tolerance values are derived from the guidelines of ISO 1328 [297] with tolerance class 7, as common in automotive applications. For parameters not specified by the standard, we have assumed reasonable tolerance values consistent with the automotive sector. It's important to note that the tolerances for $C_{H\beta}$, C_{β} and $C_{\beta L}$ are duplicated in Table 5.4, as they encompass both pinion and wheel tolerances.

Table 5.4: Manufacturing tolerances for gear parameters Δ.

h_{teeth} $[mm]$	C_{β} $[\mu m]$	$C_{H\beta}$ $[\mu m]$	C_{α_W} $[\mu m]$	C_{α_P} $[\mu m]$	$C_{H\alpha_P}$ $[\mu m]$	$C_{\beta L}$ $[\mu m]$	$C_{H\alpha_W}$ $[\mu m]$	h_{web_P} $[mm]$
± 0.1	± 5	± 24	± 2	± 2	± 10	± 6	± 10	± 0.1

5.4.2.3 Optimization Results

The NSGA-II optimisation was carried out in Python using the pymoo library [298] and the high-fidelity DT prototype of the combustion transmission. The optimization was performed with a population size of $R_t = 30$ and was terminated after a maximum of 500 generations. At each generation, $R_t/2$ offspring individuals are created. The initial population was generated using random solutions. The crossover and mutation distribution index, which affect the spread and diversity of the offspring generated, was set to 25.

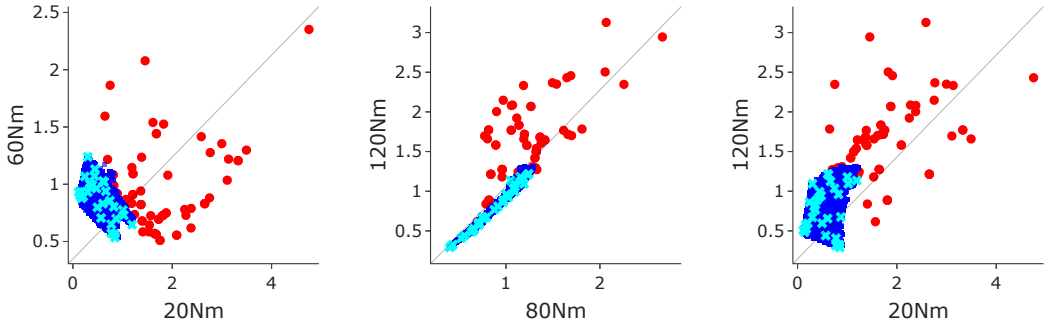
To incorporate uncertainty propagation and apply chance constraints, we employed MCS with $M = 1000$ samples for each design. The system responses underwent normalization to facilitate equitable comparisons during dominance checks, with PPTE responses being scaled by $1.5\mu m$ and the mass of the gear pair by $0.5kg$, representing reasonable expected output values.

The entire optimization process, including uncertainty propagation, took only 128.8 seconds. In contrast, the estimated time of the optimization without a surrogate model would have been approximately 37.8 million seconds or 10.5 thousand hours. In other words, the use of surrogate modeling enabled the complex multi-objective optimization of complete large transmissions under uncertainties by reducing the elapsed time by around 1 million times.

The optimization results are visually represented in Figure 5.8 for select pairs of objective functions. Red points indicate infeasible solutions, where the probability of failure exceeded 5%, blue squares represent feasible solutions, while cyan X's indicate the Pareto set in the last generation. The Pareto front solutions may not always align with the 2D frontier of the objective functions plotted, but they are non-dominated in at least one other objective from the seven-dimensional solution space.

Figure 5.8 reveals a trade-off between certain objective functions. For instance, individuals that yield reduced PPTE at 20 Nm tend to result in increased PPTE at 60 Nm and 120 Nm, as well as a higher overall mass. Furthermore, a noticeable trade-off exists between PPTE at 120 Nm and the mass of the gear pair. In contrast, the PPTE at higher torque conditions, such as 120 Nm and 80 Nm, does not exhibit conflicting trends.

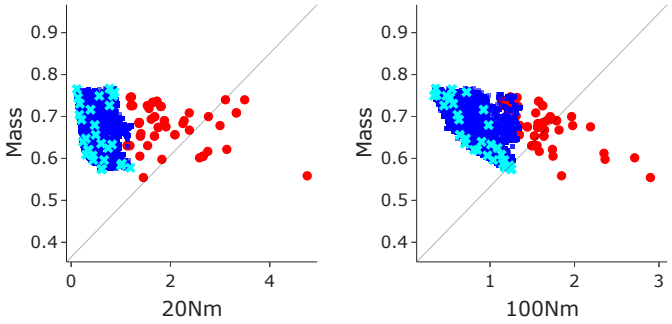
Figure 5.9 displays the nominal parameter values of individuals from the Pareto front in the last generation along with their corresponding objective functions. For comparison purposes, the results of a baseline design with nominal parameter values described in Table 5.5 are also plotted in red.



(a) PPTE at 20 Nm vs. 60 Nm.

(b) PPTE at 80 Nm vs. 120 Nm.

(c) PPTE at 20 Nm vs. 120 Nm.



(d) PPTE at 20 Nm vs. mass.

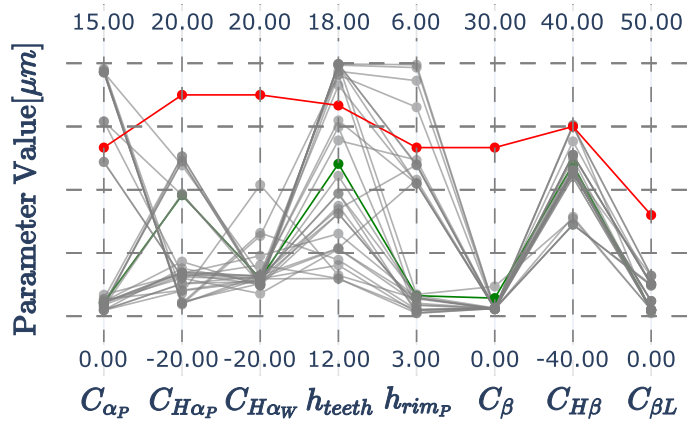
(e) PPTE at 100 Nm vs. mass.

Figure 5.8: Results of all generations of the NSGA-II optimization at different 2D views of the objective functions. The red points are infeasible, the blue squares are feasible, and the cyan crosses represent the solutions on the Pareto front.

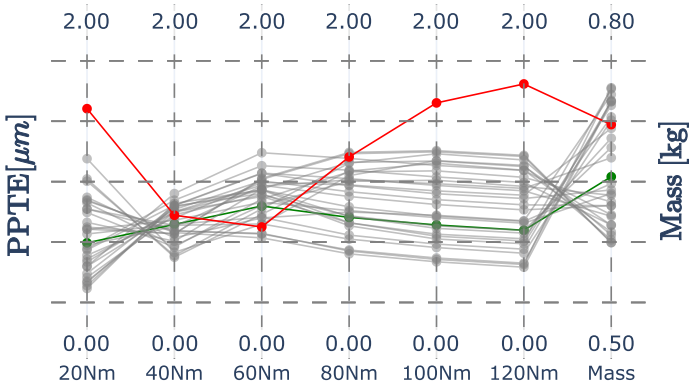
In Figure 5.9a, it can be observed that the optimal values of the lead crowning C_β and involute slope in the wheel $C_{H\alpha_W}$ are well-defined. The other parameters offer a wide range of potential solutions. Consequently, the final design decision should be guided by the trade-offs observed across different objectives, as evident from Figure 5.9b.

In this chapter, our selection of the best design from the Pareto Front is based on minimizing the weighted sum of all normalized objectives. A weight of one is assigned to the PPTE responses at each load condition, while a weight of six is assigned to the gear pair mass. Figure 5.9 displays the selected design based on this criteria by a green line. Notably, the selected design offers a good compromise between mass and PPTE across various load conditions, outperforming the baseline design in all objective measures.

To visualize the impact of uncertainties in our analysis, Figure 5.10 provides two 2D examples of the Pareto front including the min-max response range obtained with the MCS. These illustrations reveal that PPTE at low torque conditions is more sensitive to manufacturing tolerances than at higher torques. Moreover, it's notable that the robustness of PPTE to micro- and macro-geometry uncertainties



(a) Nominal values of the parameters.



(b) Objective values.

Figure 5.9: Design parameters and objective values from the baseline design (—), the individuals in the Pareto set (—), and the selected design from the Pareto set (—).

varies among individual designs. This variability underscores the importance of accounting for uncertainties in the optimization process.

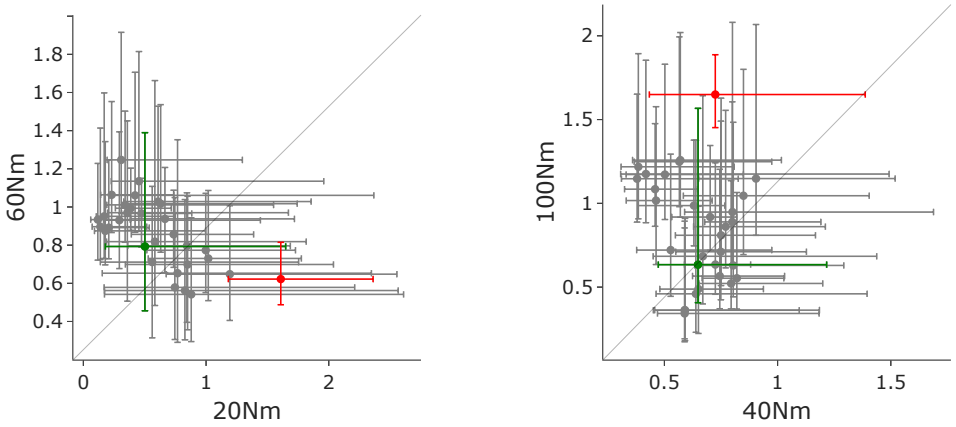
Finally, we conducted validation of the selected optimal design obtained through the optimization process using the true simulation model. As presented in Table 5.6, the deterministic outputs from the DT surrogate exhibit a satisfactory level of agreement with the ground true simulation. Specifically, the mean and maximum PPTE discrepancies across all load conditions are $0.098\mu m$ and $0.185\mu m$. The validation confirms that the proposed solution achieves the desired design criteria.

In summary, this section utilized the DT prototype of the combustion transmission for an original multi-objective optimization of gear micro- and macro-geometry under chance constraints. Surrogate modeling drastically reduced computational time, enabling this resource-intensive optimization.

The process efficiently produced a set of feasible non-dominant solutions, forming the Pareto set. These solutions offer a trade-off between different objective functions, facilitating informed decision-making. Notably, they achieved weight reduction and

Table 5.5: Nominal values of gear micro-geometry for the baseline design.

C_β [μm]	$C_{H\beta}$ [μm]	$C_{\beta,L}$ [μm]	$C_{\alpha,P}$ [μm]	$C_{H\alpha,P}$ [μm]	$C_{H\alpha,W}$ [μm]	h_{teeth} [mm]	h_{rim} [mm]
20	20	20	10	15	15	17	5



(a) PPTE at 20 Nm vs. 60 Nm.

(b) PPTE at 80 Nm vs. 120 Nm.

Figure 5.10: Different 2D views of the Pareto front with min-max response intervals including: Pareto set (—); selected design (—); and baseline design (—).

improved NVH performance for all load conditions compared to the baseline design. In essence, this approach demonstrates the effectiveness of leveraging the DT to enhance gear designs' performance and robustness amid manufacturing uncertainties.

5.5 Knowledge Continuity: Transfer Learning for Successive Digital Twins

In this section, we demonstrate the practical application of the deep TL methodology introduced in Chapter 4 in a real-world example of inter-generational DT prototyping: the transfer of information from the combustion engine-driven gear transmission, 7DCT300, to its hybridized successor, the 7HDT300. In this context,

Table 5.6: Validation of the PPTE prediction of the DT with the ground true simulation for the selected optimal design.

Load Condition	**20 Nm** [μm]	**40 Nm** [μm]	**60 Nm** [μm]	**80Nm** [μm]	**100 Nm** [μm]	**120 Nm** [μm]
DT prediction	0.49	0.65	0.80	0.70	0.64	0.60
Simulation	0.39	0.81	0.71	0.60	0.62	0.68

the combustion transmission is the initial product generation with relatively abundant dataset, serving as source domain for the TL. The hybridized transmission represents the next product generation, characterized by limited data availability, and is the target domain in our TL analyses.

The combustion and hybrid gear transmissions under analysis share many similarities, including common-parts, similar assemblies, and analogous power paths (refer to Figure 5.2). Despite their similarities, the hybrid version introduces an electric traction motor, a three-gear train connecting the electric motor to the intermediate shaft, the corresponding new bearings, and modifications into the housing to accommodate the new components, including new mounting points. These changes alter the system's behavior, affecting the stiffness and, consequently, system deflection, mesh misalignments, PPTE, and noise radiated.

Therefore, although the similarities suggest shared behavioral traits between the product generations, the information learned from the initial generation should adapt to accommodate the modifications introduced in the new generation. Assessing how much of the source information remains pertinent and how to adjust it to the target domain is a complex task that usually relies on the expertise of the product designers. The use of deep TL within the DT framework allows to automatize this task, adapting the information from the source to the target domain.

Our evaluation of TL performance in this problem involves two distinct configurations of the target domain:

1. when the hybrid transmission is driven by the combustion engine.
2. when the hybrid transmission is driven by the electric motor.

It is expected that TL will prove more beneficial in the first scenario, as the power path remains consistent between the source and target domains, resulting in a higher degree of correlation. Conversely, the second scenario introduces a shift in the power path between source and target tasks, exacerbating the disparities between the domains and, consequently, increasing the complexity of TL adaptation. In both scenarios the source model is the high-fidelity DT presented in the previous section.

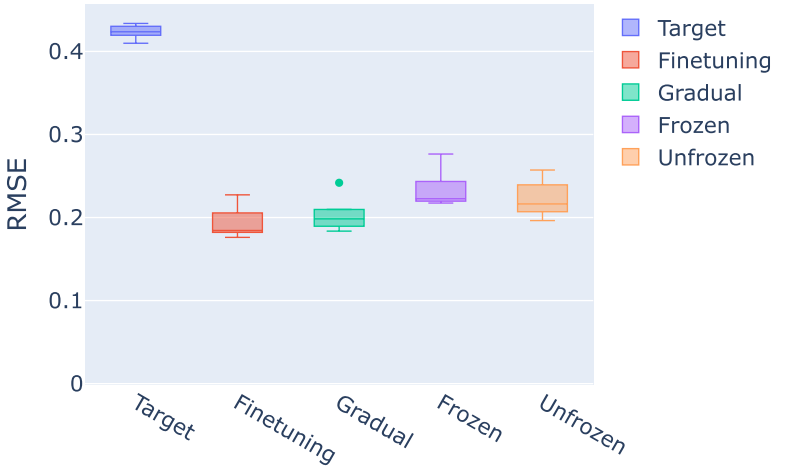
5.5.1 Transfer Learning Results

This section presents the outcomes of the deep TL between the combustion transmission (Source) and hybrid transmission (Target) domains. We assume that combustion and hybrid tasks are related and that the most influential and interacting variables identified for the source domain (refer to Section 5.3) remain consistent in the target domain.

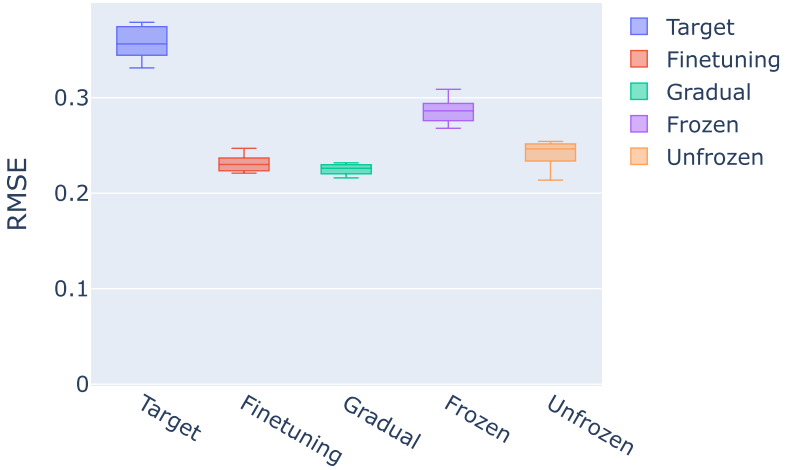
The training dataset X, Y_T comprises a mere 50 samples from the target domain, with $X_T \in \mathcal{X}_H$. We implement deep TL by using the optimal parameters $\omega, b_{opt,S}$ obtained in the training of the high-fidelity NN model $h_H(\cdot)$ for the combustion transmission as the starting point for constructing the DT for the hybrid transmission. Four distinct deep TL methods, as presented in Chapter 4, are evaluated: Finetuning, Gradual Finetuning, Frozen, and Unfrozen. To validate the accuracy improvement achieved by deep TL, we compare the performance with and without

TL.

The NN model's hyperparameters without TL are determined through a five-fold grid search, leading to a configuration of five hidden layers, each containing 16 neurons, and learning rate of 0.005. The NN models with TL retain the same architecture as the Source model, as defined in Section 5.4.1. In the case of the Finetuning and Gradual TL methodologies, a learning rate coefficient of 0.1 is adopted. This choice has proven effective across a wide spectrum of distances between the source and target domains in the study of Section 4.3.2. Within the Frozen and Unfrozen TL strategies, the top three layers of the Source model are replaced by three similar layers with random parameter initialization, which are subsequently trained from scratch.



(a) Input torque originated from the combustion engine.



(b) Input torque originated from the electric motor.

Figure 5.11: Comparing accuracy performance of the hybrid gear transmission's DT trained with scarce data: without TL and employing various TL strategies.

Figure 5.11a presents the TL results for the scenario where the input torque

originates from the combustion engine. Remarkably, all four deep TL methods outperform the Target model without TL. The Finetuning approach achieved the best performance, with errors 2.2 times smaller in average than those observed without TL.

Results for TL in the scenario where the transmission is driven by the electric motor are presented in Figure 5.11b. Once again, models with TL consistently yields lower errors than the model without TL. For this scenario, the Gradual approach leads to the most substantial improvement, with prediction errors reduced in average 1.6 times compared to the errors without TL. As expected, the extent of improvement from TL is somewhat diminished compared to the first scenario due to decreased correlation between the domains. However, these improvements remain significant even in this scenario where, besides the changes in geometry and boundary conditions, the power input has changed.

In summary, the proposed deep TL approach for inter-generational DT prototyping demonstrates successful results in the context of real-world industrial product design, enhancing data efficiency and accuracy and bridging the knowledge gap between product generations. By leveraging information from legacy designs to evolving ones, deep TL for surrogate modeling empowers industries to deal with the challenges of data scarcity during early product development. Furthermore, this approach transforms traditional reliance on expert knowledge for information continuity into a more automated, data-driven process.

5.6 Conclusions and Discussion

In this chapter, our objective was to seamlessly integrate the techniques outlined throughout this thesis into an inter-generational DT framework aimed at enhancing early product design in the field of SD&V. To demonstrate the real-world efficacy of this framework, we applied it to the construction of a DT for a combustion gear transmission system that also adapts for the next generation of transmissions.

The DT proposed is based on NN surrogate models embed with domain expertise through PGF, resulting in improved accuracy. Furthermore, the DT framework employs surrogate-based GSA for feature selection, optimization strategy refinement, and enhanced NN model interpretability. We introduce a two-level surrogate model fidelity approach, enabling informed decision-making during the design process by combining extensive exploration of design variables with high accuracy in the final model. Remarkably, the DT prototype implemented has negligible computational cost, making it a cost-effective replacement for resource-intensive simulations. Additionally, the framework employs deep TL techniques to transfer and adapt the knowledge from the current designs to future product generations, promoting information continuity and data efficiency.

In practice, we successfully implemented the DT framework for the gear transmission study. This application demonstrates the DT power when applied in the multi-objective optimization of the transmission weight and NVH performance. Due

to the minor cost of DT evaluations, we were able to explore both micro- and macro-geometric aspects of gear transmission design while accounting for uncertainties and using the coupled transmission model at various load conditions—an achievement not previously reported in the literature. Moreover, the DT, initially developed for a combustion gear transmission, has shown its adaptability by accommodating the modifications from the successor hybrid transmission through deep TL with limited data in the new task.

In sum, the proposed DT framework effectively addresses the challenges of accuracy, interpretability, and data scarcity that are often encountered in the virtual modeling and design of SD&V systems. Furthermore, its versatility makes it readily applicable in different industrial contexts. Ultimately, the DT framework presented in this chapter emerges as a powerful design tool, offering a promising avenue for enhancing product design, reducing development cycles, and improving decision-making processes in industrial applications.

Chapter 6

Conclusion and Outlook

This thesis aimed to introduce a Digital Twin (DT) framework integrating physical- and data-based models for early-stage Structural Dynamics and Vibroacoustics (SD&V) product design. The proposed DT framework uses Machine Learning (ML)-based surrogate models to enable cost-effective predictions of the system responses to support informed and fast decision-making.

The literature review identified gaps in the integration of ML techniques for SD&V product design, emphasizing challenges such as the non-smooth nature of problems, interpretability issues, and the demand for substantial data. Furthermore, it reveals a lack of research on models that explore knowledge continuity through different product generations. Therefore, this research investigates how to develop ML-based inter-generational DT that overcomes these challenges.

Benchmarking ML algorithms in Sound Transmission Loss analyses with increasing complexity levels indicate that Neural Network (NN) can accurately predict non-smooth behaviors, consistently outperforming Random Forest (RF), Gradient Boosted Decision Trees (GBT), and Gaussian Process Regressor (GPR). Additionally, integrating domain knowledge through Physics-Guided Featuress (PGFs) improved model accuracy and data efficiency. Consequently, NN with PGFs was chosen as the core algorithm for the DT.

To enhance interpretability, global sensitivity analyses, including by-products of ML methods like Mean Decrease in Impurity (MDI) and classical variance-based methods like Sobol, were integrated into the DT framework. The MDI analysis demonstrated advantages over variance-based methods regarding flexibility and accuracy when limited system evaluations are available. In contrast, Sobol analysis becomes affordable when surrogate models are available, offering further insights into input interactions. The integration of these sensitivity analyses allowed for an investigation into the physical consistency of ML models and a deeper understanding of the physical phenomena. Moreover, the selection of important features revealed by the sensitivity analysis reduces the problem dimension, contributing to improved model accuracy.

This study also promotes the adoption of inter-generational DTs, harnessing existing data from prior product generations through deep Transfer Learning (TL) techniques. An exploration into the novel application of deep TL for surrogate models with tabular data, incorporating a conditional shift between source and target domains, was conducted, underscoring the efficacy of this approach in the proposed context. Among the deep TL methods examined, Gradual Finetuning exhibited superior overall performance with more robustness to the degree of relat-

edness between the domains and to hyperparameter choice. The findings highlight that inter-generational DTs reduce the reliance on new data, fostering knowledge continuity in the product development process.

The proposed DT framework's effectiveness was demonstrated in a real-world NVH analysis, specifically in the lightweight design of silent gear transmissions. The DT exhibited predictions with improved accuracy and interpretability and significantly expedited the design process. Notably, successful knowledge transfer between generations of gear transmissions showcased the data-efficient DT modeling and adaptation capabilities of the methodology proposed.

In conclusion, the developed methodology emerges as a versatile tool with immediate applicability in industrial settings, empowering early product design and streamlining the decision-making process in SD&V.

Despite the progress made, further advancements are needed to implement a comprehensive DT, which ideally should evolve throughout the entire product life cycle and incorporate all available data.

Within the scope of this thesis, our focus has been on the initial design phases, where only simulated data is available. The next challenge in DT implementation is the integration of experimental data from the final stages of product design. At these stages, the DT should self-calibrate to correspond with measured data, ensuring it accurately represents the real system. Additionally, while the current DT effectively addresses epistemic errors, the DT enriched with measured data should be able to address also aleatory uncertainty effects.

Furthermore, the evolution of the DT should extend into the product's usage phase. The development of DT for the usage phase is a domain of ongoing and intensive research, with applications such as failure monitoring, risk assessment, mission planning, and real-time optimization and control. However, the development of DTs that span the product's entire life cycle and evolve across product generations, integrating models and assimilating information from conception to retirement, is rare. Therefore, this gap represents a promising avenue for future research.

In summary, the current research contributes to bridging the gap in ML application within SD&V product design and introducing inter-generational DTs, providing a foundation for future advancements in the field. Still, further research is required to achieve a comprehensive and integrated DT.

Publications During the Thesis

International Journal

[11] Cunha, B., Droz, C., Zine, A., Foulard, S., and Ichchou, M. A review of machine learning methods applied to structural dynamics and vibroacoustic. *Mechanical Systems and Signal Processing* 200 (2023), 11053

[107] Cunha, B. Z., Zine, A.-M., Ichchou, M., Droz, C., and Foulard, S. On Machine-Learning-Driven Surrogates for Sound Transmission Loss Simulations. *Applied Sciences* 12, 21 (2022), 10727.

International Conference

[234] Cunha, B. Z., Ichchou, M., Droz, C., Zine, A.-M., and Foulard, S. Interpretable and Physics-Supported Machine Learning Model for Sound Transmission Loss Analysis. In *International Conference on Noise and Vibration Engineering, ISMA2022*.

[299] Cunha, B., Zine, A., Ichchou, M., Droz, C., and Foulard, S. Neural Network-based Surrogates of Gear Whine Noise for Uncertainty Propagation. In *7th European Conference on Structural Control*, Vol. 1. 192–200.

[300] Cunha, B. Z., Zine, A.-M., Ichchou, M., Droz, C., and Foulard, S. On machine learning-driven surrogates for sound transmission loss simulations. The *Journal of the Acoustical Society of America* 152, 4-Supplement(102022), A135-A135. (**As invited speaker.**)

To be submitted

- Cunha, B. Z., Ichchou, M., Droz, C., Zine, A.-M., and Foulard, S. Transfer Learning for Inter-Generational Surrogate Modelling in Data-Scarce and Low-dimensional Scenarios.
- Cunha, B., Zine, A., Ichchou, M., Droz, C., and Foulard, S. Streamlining Product Design with Inter-Generational DT Framework Enabled by Transfer Learning: A Gear Transmission Case Study.

Appendices

Appendix A

Machine Learning in SD&V for Structural Health Monitoring and Active Control

This appendix presents a comprehensive literature review focused on the application of ML techniques in the realm of SD&V, with a specific emphasis on Structural Health Monitoring (SHM) and Active Control. While the primary focus of the thesis is on ML applications in SD&V for product design, a significant body of research within the broader SD&V domain has been concentrated in the areas of SHM and Active Control. Section A.1 of this appendix provides an in-depth exploration of SHM, recognized as one of the most developed applications, while Section A.2 delves into the application of ML in Active Control scenarios. This literature review offers valuable insights into the advancements, challenges, and trends in ML applications within these critical areas of SD&V research.

Part of this chapter was published in the authors' comprehensive review paper in reference [11] from the journal Mechanical Systems and Signal Processing.

A.1 Structural Health Monitoring (SHM)

SHM is an engineering area that covers detecting and diagnosing recipient failures and predicting the remaining useful life (RUL) of engineering structures based on measurements. The benefits of SHM are manifold and well known for structural reliability and integrity management, as the employment of SHM can help avoid catastrophic failures and define a maintenance schedule to optimize service time. The widespread deployment of low-cost connected sensors favors using data-driven methods over physical-based models in SHM applications [301]. In addition, physical-based models usually struggle to replicate the operating conditions of complex dynamic systems, and their costly computations are prohibitive for online monitoring [302]. On the other hand, data-driven methods can extract damage-related knowledge from data while handling its intrinsic uncertainties [301].

Due to ML capabilities of inferring knowledge from data, ML-based SHM has established itself as a major research topic, decreasing the dependency on expert judgment and increasing the accuracy and degree of automation on damage detection and assessment [70, 303]. Furthermore, most SHM methods rely on measurements of the system's dynamic response, such as vibration signals and acoustic emissions,

which can be easily monitored online during operation and are sensitive to damage on a global level, without requiring knowledge of the exact damage location. Therefore, the joint field of ML and vibration- and acoustic-based SHM is broad [304, 305], being the most extensive and consolidated use of ML in SD&V.

Although traditional ML algorithms enabled some level of automation in damage detection compared to knowledge-based approaches, they still heavily rely on handcrafted methods to unveil damage-sensitive features [70, 301]. Moreover, the most informative set of features is often unknown in real-life scenarios, especially for complex cases with little domain knowledge. Because of that, DL has been increasingly applied to automatically perform high-level feature extraction [306, 65]. Furthermore, Lei et al. [70] states that traditional ML is unsuitable for large datasets, with which DL generalizes best. Although DL can further automate and improve damage diagnosis and prognosis, its application is still limited to cases where large datasets and training time are available. The different workflows of traditional ML and DL in SHM are illustrated in Figure A.1.

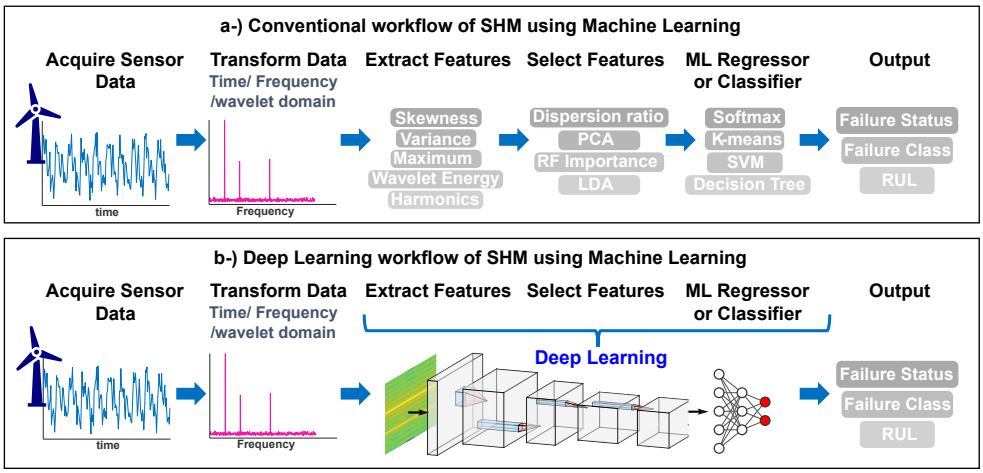


Figure A.1: Structural Health Monitoring workflow: in the traditional ML approach, feature extraction and selection are handcrafted and followed by an ML model (a); Deep learning models perform end-to-end predictions by automating feature extraction and selection (b).

ML-based SHM faces two major challenges. The first is the limited availability of labeled data since machines and structures usually operate in healthy conditions, and labeling abnormal data is costly. The second challenge is the Environmental and Operational Variability (EOV) that also changes the structural dynamic response, making it more difficult to identify damage effects on the data [307, 308]. The different levels of complexity in SHM for rotating machines and bridges illustrate these challenges. The data availability of rotating machines is usually higher than for bridges once they are often monitored and have similar counterparts. Moreover, the failure modes of rotating machines are well-defined and well-correlated with vibration signatures, making it easier to label damage [308]. The operating

and environmental conditions of rotating machines also are usually more controlled. Consequently, there are many successful industrial applications of SHM for rotating machinery, also known as condition monitoring [308, 305, 309]. On the other hand, bridges are commonly not entirely monitored, have a unique design, and their damaged response is unknown. Additionally, their excitation sources, such as traffic load and seismic activity, as well as environmental conditions, can vary considerably in an uncontrolled way.

The difficulty of the SHM problem also increases according to the prediction goal, as defined by Rytter's hierarchy [310]:

- *Level 1 - Damage Detection*: identify the presence of damage.
- *Level 2 - Damage Location*: locate the damage.
- *Level 3 - Damage Assessment*: estimate the severity and/or class of damage [1].
- *Level 4 - Damage Prognosis*: forecast health condition, such as RUL.

As the level in Rytter's hierarchy increases, it is likewise more costly to label the data, and thus, the datasets are more scarce. Furthermore, the performance of the higher levels usually depends on the previous levels [311], so the literature is uneven among the levels.

This section aims to overview ML-based approaches of SHM in SD&V, stating their merits in face of the challenges and complexity levels. Section A.1.1 defines the basics of data processing, feature extraction, and feature selection methods suitable for data of dynamic system response. Sections A.1.2 to A.1.5 review ML approaches applied to each level of Rytter's hierarchy. Section A.1.6 introduces current trends to overcome the scarcity of labeled data. Finally, Section A.1.7 summarises and discusses the strengths and limitations of the main ML algorithms used in SHM.

For further details on ML-based SHM, the reader is encouraged to consult the wealth of reviews dedicated to the topic [306, 312–314, 70, 311]. The reviews in [301, 305, 315] focused on DL-based SHM. An informative overview of SHM based on monitoring structural vibrations and waves is presented in [308]. The book by Farrar and Worden [303] hands over in-depth aspects of ML-based SHM, such as data acquisition, data processing, and ML algorithms. Reviews are also available for specific applications, such as for rotating machinery [305, 309], civil engineering [316–319], bridges [320], and earthquake engineering [321].

A.1.1 Data Processing and Features Extraction

The accuracy of the ML damage prediction strongly depends on the quality of the data provided, so proper data acquisition, signal processing, and feature extraction and selection are crucial. Even DL algorithms, which may handle raw data, can improve their accuracy and efficiency with data preprocessing. Many techniques are employed to improve vibration and acoustic data representation for SHM. The first step is proper data acquisition, including the definition of sensor type, number, and

[1] The original Rytter's hierarchy in [310] only accounts for damage severity at Level 3

location. Moreover, data should be acquired under the different expected environmental and operational conditions so that the impact of EOV can be statistically quantified [308]. Since this is often unfeasible, the ML must be designed to be robust to the EOV. Data normalization techniques also help to mitigate the EOV effects in the data to a certain degree [322, 307]. Sohn et al. [323] summarize many aspects of data acquisition and processing in SHM.

In SD&V applications, performing data domain transformation is often helpful once vibration and acoustic signals are commonly better represented in frequency or time-frequency domains. Representations on the frequency domain are suitable for stationary signals and can be obtained with fast Fourier transform, multiple signal classification, and bispectrum analysis [324–326]. Time-frequency or wavelet domain is convenient for non-stationary signals and can be obtained with discrete wavelet transform, wavelet packet transform for noise reduction and adaptive resolution [327, 328], Morlet wavelet [329], short term Fourier transform [330], Hilbert-Huang transform [330], empirical model decomposition [331], among others [332, 326].

Feature extraction can also be performed in the time domain (e.g., root mean square, skewness, kurtosis, and autoregressive coefficients), in the frequency domain (e.g., power bandwidth, harmonics, and spectral skewness), and in the time-frequency domain [326]. Other feature extraction methods used in SD&V problems include multi-domain statistical feature [333], compressed sensing techniques [327, 334] and histogram of oriented gradients for vibration images [335]. As irrelevant or redundant features and high-dimensional inputs might worsen the predictor performance, feature selection is usually performed along or after feature extraction [70]. ML algorithms for dimension reduction can also be employed to perform feature extraction and selection [305]. Varanis and Pederiva [336] compared them in an SHM context and concluded that linear discriminant analysis is suitable for non-stationary cases, PCA is convenient for stationary signals, and independent component analysis for problems with combined faults. Some ML algorithms, such as decision tree-based and LASSO algorithms, implicitly select relevant features [311].

As DL can automate these stages for large datasets, handcrafted feature extraction and selection are mainly used by traditional ML. Additionally, unsupervised DL algorithms can be used to automatically perform feature extraction and dimension reduction and then be stacked with traditional shallow ML to output the final prediction. This configuration has been increasingly explored in SHM, either for unlabeled datasets [337, 338] or labeled datasets [339–342, 328]. Autoencoders and their variants are the most commonly used algorithms in this framework.

A.1.2 Damage Detection (Level 1)

Damage detection is the most fundamental level of diagnosis for identifying whether the signal is healthy or unhealthy. In SD&V, the damage is usually assumed to change the system's dynamic response, allowing the presence of damage to be identified by monitoring deviations from normal conditions. This approach can be car-

ried out using unsupervised learning, making damage detection widely applicable to real-life problems. Given the data availability in level 1 and its importance as a foundation for subsequent Rytter's levels, there is rich literature on damage detection.

Anomaly detection algorithms identify outliers or abnormal conditions using unlabeled data and, therefore, are conveniently used for damage detection. However, the challenge in anomaly detection is to detect the damage while being robust to EOV and data noise, which can also be detected as an outlier, leading to false-positive predictions and unnecessary maintenance [343, 344]. The review in [345] classifies anomaly detection algorithms as domain-based, e.g. one-class SVM [346, 347]; distance-based, e.g. k-means and k-nn [348]; probabilistic-based, e.g. Gaussian mixture model (GMM) [349] and reconstruction-based, e.g. SOM [350], PCA [343, 344, 349], and autoencoders [337, 351]. Markou and Singh reviewed anomaly detection algorithms with a statistical approach [352] and an NN-based approach [353].

Vos et al. [346] performed anomaly detection based on only-healthy data using a one-class SVM and reported that the prediction accuracy improved by using features extracted by LSTM for consecutive time series and statistical features for non-consecutive time series. In [348], k-nn was trained with only-healthy data from a population of centrifugal fans while reducing the dataset size and computational time by selecting the most representative samples from various operational conditions. The approach led to an accurate detection of distinct anomalies with an indication of fault severity. SOM-based anomaly detection using statistical features from only-healthy data or mixed data was implemented in [350]. When the unlabeled data include healthy and non-healthy measurements, the problem of inclusive outliers should be considered [308]. To address this issue, Dervilis et al. [354] introduced a robust multivariate statistical method to reveal outliers and aid in selecting robust features.

Many anomaly detection algorithms have also delivered predictions robust to EOV. Reconstruction-based ML has been found to improve the accuracy of damage detection by isolating structural changes due to damage from EOV effects [343, 344, 349, 355]. PCA has been applied to consider the linear [343] and nonlinear [344] effects of environmental changes in the features extracted from a one-year-long vibration dataset of a bridge. Long-term bridge monitoring was also addressed in [356] using k-means clustering to identify structural and sensor damage. In [349], nonlinear PCA and GMM performed better than linear damage detection algorithms in the long-term monitoring of bridges under unknown sources of variability. According to [352], GMM performs well with limited training data and can be used in a probabilistic framework but suffers from the curse of dimensionality with high dimensional feature space. In [355], nonlinear factor analysis was used as an unsupervised NN-based method to learn the latent structure of damage features separated from EOV effects and to classify damage based on the reconstruction error. Santos et al. [347] proposed four kernel-based unsupervised algorithms to detect linear and nonlinear damage in a framed structure considering EOV. The kernel-based models

were fed in with features extracted from autoregressive models and performed better than benchmarked algorithms.

DL-based anomaly detection has also been implemented successfully in the literature. In [357], wind turbine anomalies were detected based on the reconstruction error of deep autoencoders. Similarly, aircraft fault was detected in [337] by deep autoencoders using multi-sensor raw time series as the dataset. After detecting the fault, a clustering algorithm was employed for fault disambiguation. Recently, Michau and Fink [351] used unsupervised transfer learning in anomaly detection problems to take advantage of information from other fleet instances whose data was sampled under distinct environmental and operational conditions. The article used an adversarial DL architecture to identify domain-independent features and integrate them with an extreme learning machine to perform anomaly detection.

Supervised algorithms are also applied for robust damage detection. For instance, Laory et al. [358] used supervised algorithms to study structural damage detection of a bridge with EOV based on the prediction of its natural frequencies. The study found that including temperature and traffic loads as input of the algorithm improves the accuracy of natural frequency prediction. Additionally, RF and SVM outperformed MLP, decision tree, and multiple linear regression in this task. In [359, 360], damage detection in bridges under EOV was performed using variational mode decomposition to remove seasonal patterns from frequency signals before applying RNN prediction. The publications in subsequent sections also perform damage detection, implicitly or explicitly, as the diagnosis of levels 2 to 4 depends on whether the damage was detected.

A.1.3 Damage Location (Level 2)

The second level of damage diagnosis is to locate the damage, enabling better inspection and maintenance routines. For this level of diagnosis, supervised learning is generally required.

Fuentes et al. [308] suggests that acoustic emissions are suitable for non-intrusive damage location as the difference between time-of-flight of the sensors can be used to localize the source of the unhealthy signal. This methodology was implemented in [361], where a GPR was trained to map artificial damage sources while automatically selecting active sensors. Janssen and Arteaga [324] used acoustic measurements to locate a plate's failure and investigated data processing and augmentation methods. However, this approach requires multiple sensors relatively near the structure.

In non-rotating structures, the damaged structural response is usually studied by simulations or experiments that consider stiffness reduction, losing connections, or added mass to represent the damage [317]. According to the review of SHM in civil engineering in [317], two approaches are more common for damage detection and location. In the first approach, known as parametric, the natural frequencies and mode shapes of the structure are used as features for classifiers such as MLP and neuro-fuzzy system [362, 363]. In the non-parametric approach, PCA or autoregressive models perform feature extraction, and a classifier predicts the damage

location [364, 365].

Papatheou et al. [366] added masses at different panels of an aircraft wing to simulate damage effects and accurately predicted the damage location with an MLP, even with test data from real saw-cut damage. Abdeljaber et al. [367] used compact and fast 1D CNN to enable real-time detection and location of failure at the joints of a framed structure. In [339], a sparse autoencoder extracted features from the vibration signals of an induction motor and was stacked with a dropout NN to locate the damaged component.

A.1.4 Damage Assessment (Level 3)

Damage assessment aims to define the damage severity or the damage mode. Effective diagnosis of multiple health state classifications is still challenging, especially as labeled data is rare as it usually requires expert judgment to label. The difficulty increases with system complexity, sensory data heterogeneity, strong ambient noise, and working condition fluctuations.

A great part of the literature on damage assessment is applied to rotating machines, as their vibration signatures are well-known and can correlate with damage mode and severity. Gecgel et al. [325] compared traditional ML and DL approaches to classify the severity of gear tooth crack based on simulated-based vibration signals with added noise. The accuracy of shallow SVM, RF, and decision tree algorithms using handcrafted features was inferior to the prediction accuracy of CNN and LSTM. Many techniques to encode vibration signals into images were tested to generate inputs for the CNN, but raw vibration signals reshaped as a 2D matrix led to the most accurate prediction. On the other hand, Jing et al. [333] trained a 1D CNN to classify gear fault modes and showed that the accuracy improved considerably when the data were in the frequency domain in comparison to raw input or uni-dimensional time-frequency signals. In addition, the 1D CNN reached higher accuracy than MLP, SVM, and RF. In [340], a stacked denoising autoencoder and a softmax layer were employed to classify bearing damage mode under variable operational speed and ambient noise, achieving more accurate and robust prediction than SVM, RF, and other autoencoder architectures, but requiring longer training time.

Tao et al. [341] used DBN with unsupervised pre-training and supervised fine-tuning to classify bearing fault modes. The DBN efficiently adapted multi-sensor data fusion and provided higher accuracy than SVM, k-NN, and MLP. Similarly, [368] used the weights of a pre-trained DBN to initialize and fine-tune an MLP, which outperformed SVM in identifying combined faults from bearings and gears in a gearbox. In both [341, 368], handcrafted extracted features are used as input for the DBN. Yu et al. [369] also reported improved accuracy of DBN over traditional ML when diagnosing sensor faults, actuator faults, and system faults of wind turbines. Li et al. [328] implemented a method to merge acoustic emission and vibration signals by extracting the signal features through deep Boltzmann machines and merging them with an RF, showing improved accuracy in classifying many gearbox damage

conditions in comparison to other shallow and deep ML algorithms. Booyse et al. [338] used only healthy vibration data to detect and classify damage and to predict the health index in rotating machines. Order tracking preprocessing was applied to normalize the data with respect to rotational speed and the time-synchronous average over the recording period. GAN and variational autoencoder were employed during the unsupervised learning, being that GAN presented the best performance.

According to the review in [317], damage assessment for civil structures usually relies on simulated data to work around the lack of labeled data on damage severity and mode. In [370], the severity of corrosion on bridges, characterized by the thickness reduction, is predicted with NN trained on data from finite element method (FEM) impact simulations. Hakim et al. [362] implemented an NN ensemble to accurately predict damage location and severity in an I-beam structure using the structure mode shapes as inputs.

A.1.5 Damage Prediction (Level 4)

Health prognosis analyses aim to forecast the degradation curve and the RUL and, therefore, can potentially enable maintenance schedule optimization with reduced downtime and safe operational conditions. Reviews on the approaches to evaluate RUL are provided in [371–373], mainly focusing on machinery condition monitoring. According to Lei et al. [371], health prognosis can be divided into four stages: data acquisition, health index construction, health stage division of healthy/unhealthy stages of the health index degradation trend, and RUL forecast, which is defined generally by a threshold applied to the degradation trend curve. The review also reveals that NNs and SVMs are the most used ML algorithms in health prognosis literature, followed by GPR and neuro-fuzzy systems [371].

A big challenge in health prognosis is that run-to-failure data are rare. When historical degradation data are available, ML algorithms can predict the RUL based on online-sensor data and operational conditions. Gugulothu et al. [374] proposed RNN-based algorithms for predicting RUL that do not rely on assumptions on the degradation trend. Additionally, these algorithms are robust to noisy and missing data and can capture multi-sensor temporal dependencies. The RNN generates embeddings for the multivariate sensor signals that are clean from noise and can be compared to health embeddings to estimate the RUL. For example, an LSTM was implemented in [375] to predict RUL curves with uncertainties using a turbofan benchmark dataset, outperforming CNN and other RNN-based algorithms. In their study, Zhao and Yuan [376] implemented a CNN that detects and classifies bearing damage and predicts RUL in real-time. To improve the accuracy of the predictions, the algorithm uses an online adaptive delay correction method. Other examples of NN-based RUL predictors include semi-supervised algorithms using variational autoencoder and RNN [377], as well as LSTM with dimension reduction methods for multi-sensor data [378].

Goebel et al. [379] compared the RUL prediction performance of relevance vector machine, GPR, and NN for RUL and reported that RUL predictions strongly rely

on health index predictions, which were significantly different among the algorithms. Benkedjouh et al. [380] employed nonlinear feature reduction and SVM to perform online RUL prediction of bearings based on the current health index estimation and on the degradation model fitted offline. In [381], a hybrid model with NN and GPR was applied to predict fatigue failure time with adaptive confidence interval. The adaptability potential of the GPR makes them appropriate for RUL prediction problems, whereas the dataset is small, as noted in [371]. Neuro-fuzzy systems, which use engineering knowledge and statistical information from data-driven methods, have also shown promise as RUL predictors. Chen et al. [382] used an adaptive Adaptive Neuro Fuzzy Inference System (ANFIS) integrated into a high-order state space model to predict the probability density function of RUL from a planetary gear carrier plate.

Stender et al. [383] approaches the acoustic brake squeal problem in two steps: brake NVH assessment and brake squeal prediction. First, short-time Fourier transform and data augmentation techniques are employed to create an augmented image database to train a CNN. The CNN accurately classifies brake noise and indicates when and at what frequency it occurred. In the second task, the problem parameters over time are the inputs of an LSTM that predicts when the squeal will occur. However, this methodology performed poorly when predicting a different brake configuration. TL algorithms for heterogeneous populations, outlined in Section A.1.6.3, could be employed to overcome this issue.

Similarity models can be used when run-to-failure data from similar structures are available [384, 342]. Liao et al. [342] used an enhanced Restricted Boltzmann Machine algorithm to extract and select features monotonically related to the degradation and a SOM algorithm to aggregate the features into a health index. The RUL prediction was based on the similarity with other degradation curve patterns. The authors claimed that the method is suitable for monitoring incipient damages that can lead to sudden failure. An unsupervised approach to RUL prediction based on similarity is the encoder-decoder LSTM implemented in [385]. This method predicts the health index curve based on the autoencoder reconstruction error and estimates the RUL based on the similarity of the curve predicted with train instances without making assumptions on the degradation trend.

A.1.6 Current Trends in ML-Based SHM

Recently, some approaches to tackle the lack of labeled data in SHM have gained prominence, as outlined below. Further discussion on the current and future trends of SHM is found in [70].

A.1.6.1 Hybrid Models

Hybrid models, also known as grey-box models, integrate physics-based and ML-based models to decrease the need for data, enable the inclusion of scenarios not available in the dataset, and increase interpretability while still learning from real

measurements. Discussion and examples of physics-informed ML for SHM are found in [302]. Cross et al. [187] recently reviewed physics-guided ML for SHM applications.

A.1.6.2 Semi-Supervised and Active Learning

In many SHM scenarios, unlabeled data are available, but labeled data are scarce. Semi-supervised learning is an appropriate approach in these cases, as it takes advantage of both labeled and unlabeled data. Bull et al. [386] implemented probabilistic damage classification using a GMM trained with labeled data and updated using the unlabeled data and Expectation Maximisation algorithm, showing improved accuracy compared to solely supervised learning. Yoon et al. [377] performed RUL prediction in a semi-supervised framework by using the latent variables generated by a variational autoencoder, which was unsupervised trained with all available data, as input to the supervised learning of an RNN.

AL is also an appropriate approach when it is possible to actively query and label samples that maximize the information learned, reducing the total number of labels needed. Bull et al. [387] initialized the damage diagnosis problem with a one-class GMM for anomaly detection, and, as new clusters were discovered, the model adapted to a multi-class algorithm, enabling damage assessment. The probabilistic output of the model guided the sampling strategy, and the model was updated with the new labeled data. Alternatively, Bull et al. [388] proposed a hierarchical sampling for AL through a cluster-adaptive algorithm and achieved performance comparable to supervised learning while using only a portion of the labeled data. The probabilistic AL framework implemented in [389] guided the sampling strategy to minimize the risk of a decision based rather than maximizing accuracy.

A.1.6.3 Population-Based SHM

Accurate and robust damage assessment and forecast depend on labeled data with a representative number of samples from different damaged states and environmental and operational conditions, which is often unfeasible in real-world applications. PBSHM offers a potential solution to this issue by transferring damage information and inference between similar instances of a population [42]. PBSHM assumes that the information learned in the source domain with labeled data can be reused in the target domain where labeled data are scarce or unavailable [70], similar to the concept of TL in Section 1.4.2. The foundations of PBSHM were recently reviewed in the series of articles in [43, 219–221]. According to the comprehensive PBSHM introduction by Worden et al. [42], the applications can be categorized between homogeneous population, in which the instances are nominally-identical, and heterogeneous population, in which the instances are different but share some similarities. Similarly, Lei et al. [70] categorizes TL in SHM as transfer in identical machines or across different machines.

In homogeneous PBSHM, the structures in the population are pair-wise structurally equivalent but are subject to a degree of variability in their parameters and

environmental and operational conditions [42, 43], e.g., a unit of a wind turbine fleet. According to [42], the entire homogeneous population can be generalized by a model that approximates the overall expected behavior of the population. GPRs are well-suited for this purpose because they provide mean predictions, expected population response, and variance predictions, including the response of the individual units of the population [42]. This approach was applied in [43] for similar structures that have different uncertain parameters and load conditions.

Alternatively, TL can share information from one instance to another, considering that their data belong to different distributions [70]. Lei et al. [70] presents approaches to transfer information on aspects shared by source and destination distributions for both homogeneous and heterogeneous populations. The feature-based approach, also known as domain adaptation, is the most used to reduce the discrepancies in the feature distributions between source and target domains. This is achieved by mapping the features into a latent space with shared features [223, 70]. Subsequently, the ML algorithm is trained with source-domain data in the adapted domain with minimized discrepancies where the ML can also be applied to the target domain data. The feature-based approaches include transfer component analysis, joint domain adaptation, deep TL, transfer factor, and subspace alignment, with recent applications in machinery monitoring reviewed in [70]. Lei et al. [70] also surveyed applications of GAN-based, instance-based, and parameter-based TL in PBSHM.

Heterogeneous PBSHM faces the challenge of transferring damaged knowledge between structures with substantial distribution discrepancies and, in view of this, has fewer implementations in the literature [70]. In [223], several feature-based approaches were used for heterogeneous PBSHM to detect and locate damage in building structures, outperforming a classic classifier. Gardner et al. [226] performed heterogeneous PBSHM to locate the damage on an aircraft wing using an unlabeled dataset and transferring damage-location information from another aircraft wing. They evaluated the structural similarities between the wings by creating graphical representations to identify common sub-graphs and applied balanced distribution adaptation to map the target and source domains to the common domain. A k-nn classifier was trained in the common domain with the source-domain data and achieved 100% accuracy when applied to the target dataset. Heterogeneous PBSHM developed from feature-based TL and graph representation that identifies structural similarities is also analyzed in [219, 221].

A.1.7 On the Merits of ML-Based SHM Algorithms

This section outlined the wealth of ML algorithms and approaches for feature extraction and damage prediction in SHM and SD&V. It is then pertinent to draw comments on the main algorithms used and their strength and drawbacks in SHM applications. Generally, the traditional ML approach is convenient for small datasets and cases where domain experts know an appropriate set of damage-related features. On the other hand, DL algorithms are more convenient for large-scale data and end-

to-end prediction with automated feature extraction, reducing the need for expert knowledge. The complexity of the ML model required typically increases with the complexity of the problem at hand. Therefore, DL models predominate in the upper levels of Rytter's hierarchy. Furthermore, according to this review, the most popular ML algorithms in SHM are SVM, MLP, CNN, and RNN for supervised learning, and PCA and autoencoders for unsupervised learning.

SVM is a popular algorithm in SHM due to its ability to excel in data-efficient classification and its applicability to unsupervised anomaly detection as a one-class SVM [347, 390]. As SVM maximizes the distance between the healthy and unhealthy classes, it tends to generalize well even for small datasets and feature space not thoroughly sampled. However, SVM may not be suitable for large datasets and is sensitive to the choice of kernel and hyperparameters. A review of SVM applied to SHM is provided in [390]. Shallow NNs are also widely used in SHM as they can model complex functions and support supervised and unsupervised training; However, they may not generalize well for small datasets and have a long training time. It is worth noting that both SVM and NN are not interpretable.

Conversely, decision trees, naive Bayes classifiers, and k-nn are interpretable and well-suited to investigate rules and physical meaning in damage diagnosis. These algorithms also deal better with discrete inputs than SVM, ANN, and DL [305]. However, k-nn requires much memory with large datasets and is sensitive to imbalanced distributions [371]. As decision trees usually lead to high-bias predictions, RF is a popular alternative that improves the generalization capability of decision trees while sacrificing some of their interpretability. The naive Bayes classifiers are prone to poor accuracy as they assume independent features. K-means clustering is employed mainly for unsupervised damage detection and fault disambiguation. GPR is commonly applied with small datasets and for problems that require an adaptive model or the prediction of the confidence interval.

Because of their well-known capabilities for extracting spatial features, CNNs are the most prominent algorithm in vision-based SHM [391, 312]. As time series also presents a spatial relation, i.e., the position of the data in the temporal dimension is relevant information, they are also appropriate inputs for a CNN. Many techniques to encode time series to images while preserving their spatial relationships are used to enable conventional 2D CNN. Examples of encoding techniques include the omnidirectional regeneration [335], Gramian angular field, Markov transition field [325], or wavelet transform to provide spectrograms [325, 334]. Moreover, using 1D CNNs allows time series to be directly processed to perform vibration-based SHM [391, 392, 333, 393–396]. Besides dismissing data transformation, 1D CNNs are simpler and smaller than 2D CNN, requiring fewer data and shorter training time, which enables real-time monitoring [397, 367, 398]. According to [70], ResNet architectures have similar advantages as CNN in SHM but possibly perform better for complex and variable operating conditions.

RNNs and their variants are proper candidates for SHM applications as they are designed to handle sequential data. Furthermore, RNNs were robust to EOV when applied along data normalization [359, 360]. As RNNs are computationally

expensive and require more expertise to implement and train, their use is usually justified for complex SHM problems, especially for damage forecast [375, 377, 378, 383, 385].

Other DL algorithms, such as Deep Boltzmann Machines and DBN, have also been applied in SHM [369, 399]. These algorithms allow pre-training with unsupervised learning of their layers, followed by supervised fine-tuning [70, 305, 306]. Fu et al. [399] applied DBN for cutting state monitoring and showed improved accuracy compared to traditional ML, as well as better data separability performance than PCA. DBN-based models also performed better than traditional ML and other DL models in [369, 341, 368]. The pre-training phase allows DBNs to avoid common problems from backpropagation, such as gradient vanishing and getting stuck with local minima, but they require large datasets and training time [70].

Unsupervised dimension reduction is applied in all SHM levels, usually as a data preprocessing step to automate feature extraction and selection. PCA is the most popular algorithm for linear dimension reduction and is often used to select handcrafted extracted features. On the other hand, autoencoders can extract more complex features from raw data and perform nonlinear dimension reduction, but are data-demanding and can learn unnecessary information [305]. Often the features extracted by autoencoders and PCA are the inputs for a supervised ML algorithm that performs damage diagnosis [400, 340]. Furthermore, the reconstruction errors of PCA and Autoencoders can be used for damage detection solely based on unsupervised data [357, 337]. It was also observed that nonlinear dimension reduction algorithms, such as autoencoders and kernel PCA, are more suitable for detecting damage under EOV, as EOC can have nonlinear effects on the system response.

A.2 Active Control of Noise and Vibration

Active control is the area of study that aims to model dynamic systems and design control mechanisms to guide the system behavior to the desired state. Active Vibration Control (AVC) of flexible structures is crucial for ensuring the safety, comfort, and precision of various structures, including vehicles, aircraft, machines, and buildings[401–403]. Active Noise Control (AVC) or noise-canceling is a subject of longstanding research that is based on destructive interference to reduce noise levels [404]. The growing importance of user comfort, ergonomics, and NVH performance during product development [405] increases the efforts to control vibration and noise. The need for AVC and ANC is higher in low-frequency ranges, where the application of passive control is limited [406]. As such, this section explores ML algorithms applied in the active control of noise and vibration, with a focus on system identification, reduced-order modeling, sensor and actuator placement, and adaptive control design.

Active control and ML are deeply correlated fields, both of which rely on data-driven approaches that have been accelerated by the popularization of sensors, IoT devices, and improvements in signal processing and computational power. In ad-

dition, ML algorithms can be applied in several stages of the control system, as analyzed in [407, 205]. An extensive but not exhaustive number of applications of ML in ANC and AVC is illustrated in Figure A.2. Brunton's series of videos, named *"Data-driven control with machine learning"* [408], covers overall aspects of ML applied in active control. The Least Mean Squares (LMS) filter is a basic linear ML algorithm widely used in active control to estimate the state and controller parameters. More advanced ML algorithms, especially NN, are also popular for modeling and controlling nonlinear systems where linear control theory can fail. However, when feasible, linear control methods are prioritized because of their shorter response time and well-developed linear control algorithms.

Back in the 1990s, many NN applications in active control had already been identified with three usual configurations [409–411]: NN-based model predictive control, in which an NN black-box models the forward dynamics of the system [412]; as an NN-based model-free controller [413]; and in NN-based model reference control, where NN models the plant and optimizes the controller parameters [414]. The first and third configurations use NN in the system modeling stage, while the second and third configurations use NN to learn the optimal controller design.

The following subsections outline these and other applications of ML in ANC and AVC. Section A.2.1 reviews the application in dynamic system modeling, specifically in system identification and reduced order modeling. Section A.2.2 overviews ML use in controller design. Finally, Section A.2.3 summarises the merits of ML algorithms in active control in SD&V.

A.2.1 Dynamic System Modeling with ML

A big part of active control theory relies on model-based control techniques in which control needs to have a mathematical model of the physical system, as in model predictive control and linear optimal control [407]. However, in practical situations, there are two main obstacles:

- The physical model of the system is unknown, or the model parameters which fit the system equation are unknown. In this case, *system identification* techniques are required.
- The physical model is known, but its complexity is unfeasible for real-time control applications. Then, *reduced order modeling* is needed.

A.2.1.1 System Identification

System Identification (SI) refers to the set of techniques that use measured data of a system to model the relationship between the input and output of the system. This description relates to the ML definition of inferring model from data [407, 415]. In fact, some classical SI techniques, such as the eigensystem realization algorithm, Kalman filters, and linear parameter varying, can be considered an early form of ML [416]. From this, it can be reasoned that modern ML methods are used for complex and nonlinear SI.

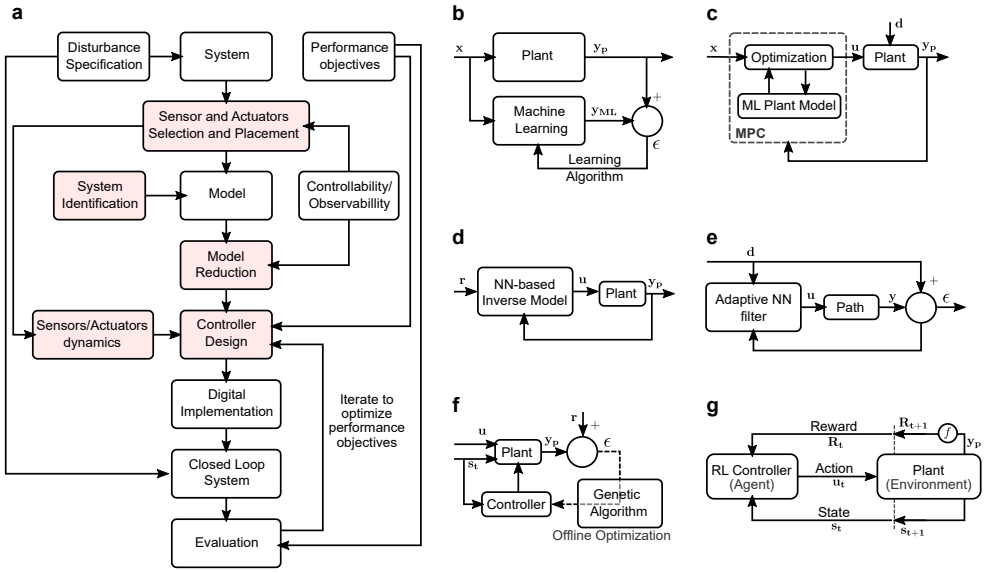


Figure A.2: Applications of active control of vibrations and noise powered by ML. **(a)** Active control workflow highlighting with red shadow the process that can use ML; **(b)** Scheme of ML training in system identification problems; **(c)** Model predictive control based on ML-model; **(d)** Controller design using NN-based inverse model; **(e)** Active control using adaptive NN filter to define control parameters; **(f)** ML control using heuristic methods to optimize control structure and parameters; **(g)** Reinforcement learning applied to active control.

Nonlinear Autoregressive Exogenous Models (NARX) are widely used to model stochastic nonlinear dynamic systems in control and can be formulated based on ML models [417]. ML-based NARX avoids the problem of determining the model structure faced by classic NARX in the polynomial form [418]. Ljung et al. [415] analyzed the similarity of DL and SI concepts and showed that NNs can be described as NARX models. The sequential dynamic structure of RNNs is also suitable for being employed as a NARX [419]. Recently, NARX based on dynamic GP [188] and PCE [420] were developed with the advantage of providing confidence intervals [418].

Autoregressive models such as NARX are well suited for model predictive control as they can predict n-steps ahead. Model predictive control uses the predicted system response, based on the plant model, to optimize the control signal over a finite-time horizon in relation to the control cost function, using a feedforward configuration. NN-based predictive control was applied for vibration control of a tall building using active tuned mass damper [421], combining good aspects of pole-placement and neuro-fuzzy control. NN-based predictive control implemented for vehicle active suspension control is found in [422–424].

Several control approaches work with the state-space representation of the system, which is usually constructed based on first-principle knowledge of the system dynamics [417]. Kalman filter is used to estimate state variables in a linear system, while the variants extended Kalman filter and the unscented Kalman filter are nonlinear state estimators [425]. These state estimators are largely applied in SD&V, such as for SI in vehicle dynamics [426]. The augmented Kalman filter proposed in [427] includes the identification of unknown forces in structural dynamics, as demonstrated in the application to rotor dynamics [428]. Recent developments with ensemble-based estimators such as ensemble Kalman filter [429] and particle filter [430] should be considered for non-Gaussian state-space models, as implemented in [431] for a highly nonlinear mass-spring oscillator.

Nonlinear state-space models can also be obtained by ML black-box models [432]. For instance, Ljung et al. [415] implemented an LSTM to identify a nonlinear state-space model. GPRs have been used for joint input-state estimation in linear [433] and nonlinear [434] systems in SD&V. Reviews on ML-based SI focusing on kernel-based methods and their capabilities for continuous structure selection over traditional SI methods are found in [435, 436]. ML algorithms for nonlinear SI in structural dynamics are discussed in [437, 438].

Sparse system identification of nonlinear dynamics (SINDy) was proposed by Brunton et al. [439] to enable the discovery of governing equations in high-dimensional systems employing sparse regression techniques. SINDy was also applied to discover the governing nonlinear Ordinarial Diferential Equation (ODE) in SD&V systems, including studies on systems with geometrical nonlinearities [440] and the impulsive response of damped oscillator [441]. Different regression processes and excitations were investigated in [442] to experimentally identify the governing equation of an oscillator under uncertainty analysis. Other sparse system identification algorithms with ML are outlined in [435]. The SINDy approach leads to

interpretable data-driven SI and is used for nonlinear reduced-order modeling.

A.2.1.2 Reduced Order Models and Sensors/Actuators Placement

Reduced order model (ROM) uses lower-rank representations of the system without losing valuable information about its dynamics. In this way, ROMs reduce response time and memory requirements of full-scale models, being critical to enabling efficient real-time control. ML algorithms play a significant role in reduce order modeling.

One scenario in which ROMs are applied in control is when there is a high-dimensional numerical model of the system that is computationally expensive for real-time applications. In such cases, ROMs or metamodels are used to speed up the simulation of the system prediction in model-based control. Feedback control might require further reduction in the space-state representation. Component mode synthesis is used for linear ROM of structures from FEM models, while ROM based on proper orthogonal decomposition, dynamic mode decomposition, and nonlinear normal modes can be obtained directly from measured data [443, 205]. Additionally, the ML-based surrogates presented in Section 1.3 can also create less expensive models for model-based control.

Proper orthogonal decomposition applies a coordinate transformation from the physical coordinates to an orthonormal basis formed by the system eigenvectors and is equivalent to PCA in the nomenclature of the ML field. By selecting only the main modal contributions, or first principal components, the system model is represented on a reduced basis, which is convenient to model SD&V problems used in space-state AVC and ANC [444–448]. Modal basis representations also provide useful information on the controllability, observability, and stability of the system, which are key factors in defining the optimal placement of sensors and actuators. PCA has been employed for this purpose in [449, 445]. The location of sensors and actuators is a crucial aspect of active control, as it influences the control efficiency, cost, and stability [403].

Dynamic Mode Decomposition (DMD) extracts simple spatiotemporal coherent modes from either linear or nonlinear dynamic systems based on data-driven regression [450, 205]. However, unlike PCA, the modes extracted by DMD are not guaranteed to be orthogonal, which may result in a less compact decomposition. Nonetheless, as demonstrated in [451], DMD is strongly related to the Koopman operator theory, which describes a nonlinear system on an infinite-dimensional linear basis. Thus, it enables the use of well-known linear control methods in nonlinear systems. Recently, data-driven DMD was applied to extract the modal parameters from a cantilever beam [452]. Fonzi et al. [453] employed DMD to model fluid-structure interactions in an aeroelastic morphing wing and used model predictive control over multiple operating regimes. Moreover, Brunton et al. [205] underlined the recent effort in control to find nonlinear Koopman coordinate systems by means of DL algorithms.

Alternatively, Nonlinear Normal Modes (NNM) can be used for SI and as or-

thogonal bases in ROM of nonlinear dynamical systems [454]. [455] studied the application of PCA and asymptotic NNM for ROM of a structure with nonlinearities. Although the NNM led to a more significant model reduction, the PCA performed better for large vibration amplitudes and parameter variations. Recently, Worden and Green [456] proposed an approach based on ML and optimization to find NNM. The algorithm searches for statistically independent modes and uses a GPR to perform the inversion of the modal transformation, allowing the approximation of modal superposition. In [457], NNMs are obtained by kernel-based independent component analysis and locally linear-embedding manifold learning in a more straightforward black-box procedure, requiring less domain knowledge. A novel approach proposed in [458] employed cycle-GAN to learn forward and backward transformations to modal coordinates with orthogonality restriction.

Real-time predictive control applies other tools combining ML algorithms with ROM. The following examples explore these techniques, which can be employed in online control. Liu et al. [459] developed an automatic updating FEM model using component mode synthesis and GPR. Simpson et al. [443] used an autoencoder to obtain the NNM of a framed structure with hysteresis and used it alongside an LSTM model to predict the system dynamics in near real-time. Using cluster-based ROM, already explored in fluid control [460] and static structural mechanics [461], could have potential use in the SD&V field.

A.2.2 ML-Driven Controller Design

Another application of ML algorithms is in the controller design, that is, in optimizing the control signal or control laws regarding the cost function that quantifies the control performance. While in the last section ML models predict the forward output of the system, the following references use ML to learn effective control laws. ML-based controllers are mainly used to handle nonlinear systems, especially with NNs, as evidenced in the survey on nonlinear ANC in [462]. Several configurations use ML to support the controller design, such as model reference control, inverse-dynamics control, ML control, neuro-fuzzy control, and reinforcement control.

In NN-based model reference control, two NNs form the control system: an NN models the plant to predict the system response, and the other NN defines the controller parameters optimized to minimize the error between system response and the reference signal [414]. Vidya and Dharmana [422] implemented a model reference control of a vehicle suspension using an NN reference algorithm and an RNN controller, claiming that it leads to better adaptivity and stability. The drawback of NN-based reference control is that it uses dynamic backpropagation in the optimization, which is computationally expensive [409].

Adaptive NN controllers are used in noise and vibration control with diverse methodologies. An example is the NN-based inverse dynamics control, which consists of training an NN with the inverse system dynamics and using it to determine the controller parameters, as in a regressor-based control. De Abreu et al. [463] implemented a direct inverse NN control of a vibratory system by training an NN

as the inverse model of the plant, such that the NN receives the current state and the desired state and outputs the actuator signal. Similarly, Ariza-Zambrano and Serpa [464] applied direct inverse NN control to a beam cantilever, in which the NN was trained both with a full-state FEM model and with a ROM to account for dynamic uncertainties in practical scenarios, showing more stable results than H-infinity control. Nerves and Krishnan [465] used NN direct controller to control wind-induced vibrations in a building-TMD (tuned mass damper) system by considering the plant as the output layer of the NN, as in a feedback linearization control. Bani-Hani [466] applied NN to model both a direct forecasting model and an inverse model to control wind-induced vibrations.

Several ANC configurations have employed model-free ML controllers as a nonlinear alternative to the commonly used adaptive filtered-X LMS algorithm. Park [467] tested different configurations of NN as the adaptive controller in a feedback configuration for different ANC datasets. CNN was the one that performed the best, followed by MLP and RNN, all of them with better performance than typical LMS-based controllers. For the case of a feedforward noise control system with a nonlinear primary path, Zhang et al. [468] also obtained better performance with an LSTM-based controller than with Filtered-X LMS. A comparison of the online learning performance of adaptive filters in an ANC application showed the superiority of kernel-based models, such as Kernel-LMS and Kernel affine projection algorithms, over classical LMS and NN algorithms [469, 470].

Zhang and Wang [471] implemented a deep-ANC in a feedforward configuration in which a convolutional RNN is used to estimate the optimal control signal-to-noise cancellation. The supervised training of the network uses the reference signal as input and the ideal anti-noise as the target, both in their spectrogram format. Besides that, the ML algorithm predicts the canceling signal with some frames in advance to compensate for its delay. Compared to typical ANC, the approach improved noise canceling in noise-only and noisy speech scenarios. Other examples of NN applications are found in the review on ANC for nonlinear systems in [462].

Heuristic algorithms, such as genetic algorithms and particle swarm optimization, can search for an arbitrary optimal control law in Machine Learning Control (MLC). According to Hansen et al. [406], MLC can optimally adapt the weights of any nonlinear filter structure, including an NN. As MLC does not rely on a fixed structure of the controller or a model of the system, it gives more flexibility to the optimization, with the downside that it adapts slowly, preventing its online application to a transient system. Chapter 2 of Duriez's book [416] briefly introduces MLC. Wangler and Hansen [472] were pioneers in applying MLC in active control of noise and vibration and were followed by many others in ANC [473–479] and AVC [480–484].

Neuro-fuzzy control systems, especially using ANFIS, have been widely applied in active control in SD&V, for example, in noise control [485–488] and vibration control [489, 490]. Neuro-fuzzy systems usually use expert knowledge to set initial fuzzy rules in an NN-like structure where the neuro-fuzzy parameters are adapted during the training to fit measured data. The resultant neuro-fuzzy systems combine

the advantages of using interpretable explicit rules from fuzzy rules with the learning capabilities of NN.

Finally, noteworthy results have been achieved with RL for control [491]. In RL, the agent (the controller) can interact with the environment (the dynamic system), and its actions will affect the output of the system and, therefore, the value function quantifying the long-term performance of the control, which the algorithm optimizes. In this way, the RL algorithm can interactively learn information about the system and the controller behavior altogether, similar to human learning. Detailed explanation and reference examples on RL for control are presented in [491, 492].

RL for control has gained prominence in applications such as autonomous car control and robot control [493] but has also shown applicability in SD&V, especially for problems with high uncertainty and stochastic behavior [491]. Latifi et al. [494] presented a successful example in which they applied an RL algorithm to manipulate an acoustic field by controlling a centrally-actuated vibrating plate (Chladni plate) and, in this way, they guided a particle towards a target location on the plate surface. In [495], the ANC implemented using the Q-learning algorithm had satisfactory results, showing the great capability to adapt when the secondary path of the noise changed suddenly.

Qiu et al. [496] carried out bending and torsional vibration control via an RL algorithm virtually trained with a FEM model and transferred to an experimental setup where it outperformed a PD control. The vibration control of a rotating machine was also performed through RL using pad actuators in [497]. Gulde et al. [498] implemented a control method with RL to compensate for vibrations in an industrial machine tool. RL-based control achieved good controllability of flexible building in [499, 500]. Although RL shows potential for real-time decision-making control in complex and uncertain scenarios, it demands considerable training time and expensive computational resources and, therefore, its use may be superfluous to applications already mastered with simpler solutions.

A.2.3 On the Merits of ML for Active Control of Noise and Vibration

This section reviewed the main applications of ML in AVC and ANC, namely SI, ROM, and controller design. In general, ML is applied in active control when traditional techniques may fail, as for complex and nonlinear systems.

In SI, ML algorithms avoid the problem from classical SI methods of selecting the appropriate model order[436], as they account for a broader hypothesis space. Thus, ML can either be used as black-box models or to extend versions of classical methods, such as NARX and Kalman filters.

In ROM, ML algorithms are used to reduce data dimensionality, with PCA being one of the most used methods. Data-driven DMD and ML-enhanced NNM have also been recently used for ROM. Order reduction ML algorithms can also help reduce the number of required sensors and actuators while maintaining a good level of control performance.

As for controller design, NN is extensively used to determine the controller parameters due to its great approximation performance. Moreover, NNs can learn from data and adapt to changing conditions and, therefore, be employed for real-time control. A growing research area for controller design is the application of RL algorithms to learn the optimal control policy through trial-and-error interactions with the system. They are better than classic algorithms because they handle complex and uncertain dynamics. However, their use is still limited as they require extensive training and are computationally expensive.

Appendix B

Methodology of Multi-objective Optimization with Chance Constrain

Multi-objective optimization problems are essential in engineering design to find a balance between conflicting objectives while respecting design constrains. This problem can be mathematically represented as follows:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x}) = \left[f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_{m_f}(\boldsymbol{x})\right] \\ \text{s.t.} \quad & g_j(\boldsymbol{x}) \leq 0, \quad \text{for } j = 1, 2, \ldots, p \\ & h_k(\boldsymbol{x}) = 0, \quad \text{for } k = 1, 2, \ldots, q \end{aligned} \tag{B.1}$$

where:

- $\boldsymbol{x} \in \mathbb{R}^n$ represents the vector of n decision variables.
- $\boldsymbol{f}(\boldsymbol{x})$ is a vector of m_f objective functions to be minimized.
- $g_j(\boldsymbol{x})$ and $h_k(\boldsymbol{x})$ denote inequality and equality constraints, respectively, that define the feasible region in the decision variable space.

Several algorithms address such multi-objective optimization challenges. In the present gear transmission optimization, the use of a differentiable NN-based DT enables the use of gradient-based optimization methods. However, the PPTE problem lacks convexity guarantees, which can lead gradient-based methods to ill-conditioning and local minima convergence.

An alternative option lies in evolutionary algorithms, renowned for their efficacy in complex multi-objective optimization scenarios, particularly when addressing non differentiable and non convex functions. These algorithms aim to find a set of solutions known as the Pareto frontier or Pareto set. Pareto frontier are non-dominated solutions, ensuring that improving one objective worsens at least one other objective. This property empowers decision-makers to evaluate the viable solutions based on the revealed trade-offs [501]. Additionally, evolutionary method can be parallelized, which synergizes well with the parallel capabilities of NN-based DTs, thereby expediting the optimization calculations.

Given these considerations, we opt for Genetic Algorithms, a popular class of Evolutionary algorithms, to perform the NVH multi-objective optimization. This choice facilitates efficient exploration of the design space, captures trade-offs, and can be readily applied within the DT framework. In this section, we outline the framework and introduce the method employed in this optimization.

B.1 Genetic Algorithm

In Genetic Algorithm (GA), a population of potential solutions P, known as individuals, evolves over multiple generations to find the best solutions to the problem at hand, converging towards optimal or near-optimal solutions. The evolution process in a GA is a metaheuristic iterative algorithm inspired by natural selection and evolution. Each individual solution $\boldsymbol{x}$ in the GA can be represented as a chromosome, denoted as $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]$.

The main stages of GA are as follows:

1. *Initialization:* Create the initial population of individuals, typically generated randomly or through some heuristic method.
2. *Fitness Evaluation:* Assess the fitness of the population based on the problem's objectives.
3. *Selection:* Determine which individuals survive and reproduce based on their fitness evaluation.
4. *Reproduction:* Create new offspring solutions using genetic operators such as crossover and mutation.
5. *Iteration:* Repeat steps 2 to 4 until stop criteria is met.

The GA can be categorized as elitist or non-elitist, depending on the selection of surviving individuals. An elitist GA ensures that individuals with the best fitness from the current population proceed to the next generation.

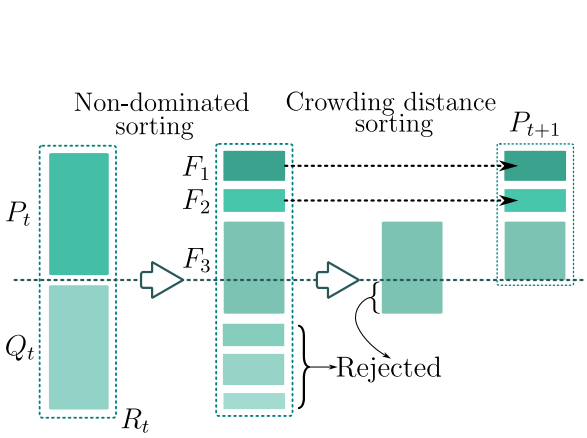
In the crossover operation, genetic material from two parent individuals is combined to generate the crossover child. The specific crossover operator may vary, but it usually consists in selecting a crossover point, and portions of the chromosomes beyond that point are exchanged between the parents, This allows to benefit of traits from different parents. In the mutation operation, random changes are introduced to an individual chromosome, allowing for exploration of new solutions in the population.

The various variations and techniques that can be employed to GAs enhances its flexibility and performance at different scenarios. In our specific case, we have chosen to use the Non-dominated Sorting Genetic Algorithm II (NSGA-II). This decision is based on the algorithm's well-established track record of efficiency and effectiveness in addressing multi-objective optimization challenges within real-world applications [502, 501].

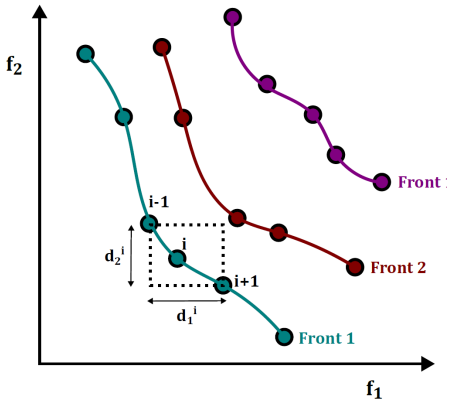
B.2 Non-dominated Sorting Genetic Algorithm II

Non-dominated Sorting Genetic Algorithm II (NSGA-II) [502] is an elitist extension of the GA, designed to efficiently solve multi-objective optimization problems. It incorporates non-dominated sorting and crowding distance calculation to guide the search towards the Pareto frontier while compromising exploration and exploitation.

Figure B.1a depicts the scheme for generating the next generation in NSGA-II, with a population size of $R_t = 2R$. Ranking individuals based on dominance



(a) Workflow to generate offspring population.



(b) Dominance ranking and crowding distance 2D example.

Figure B.1: NSGA-II

relationships, NSGA-II categorizes them into fronts. The leading front comprises non-dominated solutions (Pareto front), while subsequent fronts are dominated by preceding ones, as shown in Figure B.1b.

NSGA-II ensures that the best R individuals progress to the next generation, following an elitist approach. The algorithm also promotes diversity by, at each partially taken front (see front F_3 in the example of Figure B.1a), preferring individuals in less crowded regions. For that, it uses crowding distance to measure the density of solutions in a particular region of the objective space (Figure B.1b). Extreme points are assigned infinite crowding distances to maintain their presence in every generation.

Generating the remaining R offspring entails crossover and mutation operations. Parent selection employs binary tournament selection, pitting two randomly chosen individuals against each other to select the fitter one as parent [298], intensifying competition among individuals.

Therefore, NSGA-II integrates multiple techniques for effective multi-objective optimization. It concurrently optimizes various objectives and ensures Pareto-optimal solutions through non-dominated sorting. Moreover, it maintains a balance between exploration and exploitation by promoting the progress of the best-fit solutions while encouraging diversity.

B.3 Optimization Under Uncertainties

All engineered systems confront uncertainties-both aleatory and epistemic- that should be considered in product design. Overlooking these uncertainties during optimization can lead to sensitive solutions with compromised performance. Two primary strategies for incorporating robustness in optimization are robust optimization and chance constrained optimization.

Robust optimization employs worst-case analysis by considering a range of pos-

sible values for uncertain parameters. It aims to find a solution that performs well across all scenarios within this range. Robust optimization provides a conservative solution that can handle a wide range of uncertainties but may sacrifice optimally to ensure robustness.

Chance constrained optimization, on the other hand, incorporates probabilistic constraints, ensuring that the likelihood of violating specified threshold (failure probability) remains within acceptable bounds. Balancing optimization with risk, this approach offers greater flexibility than robust optimization.

For the gear transmission optimization, chance constrained optimization is favored because manufacturing errors can be well described probabilistically, and a certain level of risk is acceptable for the NVH performance.

Central to the assessment of failure probabilities in chance constrained optimization is uncertainty propagation. A common approach for uncertainty propagation is MCS, which involves running the model with copious inputs randomly sampled from the input probability distributions to evaluate the statistical moments of the resulting outputs. MCS is an accurate but computationally demanding method, requiring a cupious number M of evaluations proportional to the inverse of the desired evaluation accuracy [503]. Thus, achieving an evaluation accuracy of approximately 0.001 requires around $M = 1000$ samples for each considered design.

Harnessing the parallel processing capabilities of NN-based surrogates can mitigate this computational load by simultaneously evaluating multiple samples through batch processing. By reducing the computational cost associated with uncertainty evaluation in optimization, the DT enables extensive and robust domain explorations.

Bibliography

[1] European Parliament, Council of the European Union . 2019. Regulation (EU) 2019/631.
http://data.europa.eu/eli/reg/2019/631/oj. (Cited on page 1.)

[2] European Parliament, Council of the European Union . 2014. Regulation (EU) No 540/2014.
http://data.europa.eu/eli/reg/2014/540/oj. (Cited on page 1.)

[3] Andy Moorhouse, Andy Elliott, and Joshua Meggitt. 2022. Virtual Acoustic Prototyping-A story of four decades. (Cited on page 1.)

[4] Lonny L Thompson. 2006. A review of finite-element methods for time-harmonic acoustics. *The Journal of the Acoustical Society of America* 119, 3 (2006), 1315–1330. (Cited on page 1.)

[5] Michael Grieves and John Vickers. 2017. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary perspectives on complex systems*. Springer, 85–113. (Cited on pages 1, 9, 11 and 12.)

[6] David Cearley, Alfonso Velosa, and Mike Walker. 2017. Top 10 Strategic Technology Trends for 2017: Digital Twins. https://www.gartner.com/en/documents/3647717/top-10-strategic-technology-trends-for-2017-digital-twin (Not cited.)

[7] Marc Kerremans, David Cearley, Alfonso Velosa, Mike Walker, and Brian Burke. 2018. Top 10 Strategic Technology Trends for 2018: Digital Twins. https://www.gartner.com/en/documents/3867164/top-10-strategic-technology-trends-for-2018-digital-twin (Not cited.)

[8] David Cearley, Brian Burke, Alfonso Velosa, and Marc Kerremans. 2019. Top 10 Strategic Technology Trends for 2019: Digital Twins. https://www.gartner.com/en/documents/3904569/top-10-strategic-technology-trends-for-2019-digital-twin (Not cited.)

[9] Edward Glaessgen and David Stargel. 2012. The digital twin paradigm for future NASA and US Air Force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*. 1818. (Cited on pages 9 and 10.)

[10] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. 2020. Characterising the Digital Twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology* 29 (2020), 36–52. (Cited on pages 1, 9, 11, 12 and 13.)

[11] Barbara Zaparoli Cunha, Christophe Droz, Abdel-Malek Zine, Stéphane Foulard, and Mohamed Ichchou. 2023. A review of machine learning methods applied to structural dynamics and vibroacoustic. *Mechanical Systems and Signal Processing* 200 (2023), 110535. (Cited on pages 2, 7, 12, 83, 125 and 129.)

[12] Ágnes Bárkányi, Tibor Chován, Sándor Németh, and János Abonyi. 2021. Modelling for Digital Twins—Potential Role of Surrogate Models. *Processes* 9, 3 (2021). https://doi.org/10.3390/pr9030476 (Cited on pages 2 and 23.)

[13] Yongliang Wang, Xunpeng Qin, Song Huang, Li Lu, Qingkai Zhang, and Jiawei Feng. 2017. Structural-borne acoustics analysis and multi-objective optimization by using panel acoustic participation and response surface methodology. *Applied Acoustics* 116 (2017), 139–151. (Cited on pages 2, 22, 24 and 27.)

[14] Xinhua Liang, Zhongqin Lin, and Ping Zhu. 2007. Acoustic analysis of damping structure with response surface method. *Applied Acoustics* 68, 9 (2007), 1036–1053. (Cited on page 24.)

[15] Rong Guo, Tiantian Mi, Lu Li, and Rui Luo. 2022. Research on aerodynamic performance and noise reduction of high-voltage fans on fuel cell vehicles. *Applied Acoustics* 186 (2022), 108454. (Cited on pages 2, 22, 24 and 27.)

[16] Alice Cicirello, Brian R Mace, Michael J Kingan, and Yi Yang. 2020. Sensitivity analysis of generalised eigenproblems and application to wave and finite element models. *Journal of Sound and Vibration* 478 (2020), 115345. (Cited on pages 2, 23 and 26.)

[17] Stefano Marelli, Paul-Remo Wagner, Christos Lataniotis, and Bruno Sudret. 2021. Stochastic spectral embedding. *International Journal for Uncertainty Quantification* 11, 2 (2021). (Cited on pages 2, 23, 24 and 29.)

[18] Genzi Li, Vikrant Aute, and Shapour Azarm. 2010. An accumulative error based adaptive design of experiments for offline metamodeling. *Structural and Multidisciplinary Optimization* 40, 1 (2010), 137–155. (Cited on pages 2 and 24.)

[19] Yao Lin, Farrokh Mistree, Janet K Allen, Kwok-Leung Tsui, and Victoria CP Chen. 2004. A sequential exploratory experimental design method: development of appropriate empirical models in design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 46946. 1021–1035. (Not cited.)

[20] A Farhang-Mehr and S Azarm. 2005. Bayesian meta-modelling of engineering design simulations: a sequential approach with adaptation to irregularities in the response behaviour. *Internat. J. Numer. Methods Engrg.* 62, 15 (2005), 2104–2126. (Cited on pages 2 and 24.)

[21] Siavash H Khajavi, Naser Hossein Motlagh, Alireza Jaribion, Liss C Werner, and Jan Holmström. 2019. Digital twin: vision, benefits, boundaries, and creation for buildings. *IEEE Access* 7 (2019), 147406–147419. (Cited on page 9.)

[22] Ehab Shahat, Chang T Hyun, and Chunho Yeom. 2021. City digital twin potentials: A review and research agenda. *Sustainability* 13, 6 (2021), 3386. (Cited on page 9.)

[23] Chiara Cimino, Elisa Negri, and Luca Fumagalli. 2019. Review of digital twin applications in manufacturing. *Computers in Industry* 113 (2019), 103130. (Cited on pages 9 and 12.)

[24] Heikki Laaki, Yoan Miche, and Kari Tammi. 2019. Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery. *Ieee Access* 7 (2019), 20325–20336. (Cited on page 9.)

[25] Hendrik Van der Valk, Hendrik Haße, Frederik Möller, Michael Arbter, Jan-Luca Henning, and Boris Otto. 2020. A Taxonomy of Digital Twins.. In *AMCIS*. 1–10. (Cited on page 9.)

[26] Louise Wright and Stuart Davidson. 2020. How to tell the difference between a model and a digital twin. *Advanced Modeling and Simulation in Engineering Sciences* 7, 1 (2020), 1–13. (Not cited.)

[27] DJ Wagg, Keith Worden, RJ Barthorpe, and Paul Gardner. 2020. Digital twins: state-of-the-art and future directions for modeling and simulation in engineering dynamics applications. *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg* 6, 3 (2020). (Not cited.)

[28] Hitesh Hinduja, Shreya Kekkar, Smruti Chourasia, and Hrishikesh Bharadwaj Chakrapani. 2020. Industry 4.0: digital twin and its industrial applications. *Int. J. Sci. Eng. Technol. Open Access J* 8, 4 (2020). (Not cited.)

[29] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. 2021. Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems. arXiv:2003.04919 [physics.comp-ph] (Cited on pages 24, 30 and 33.)

[30] Kendrik Yan Hong Lim, Pai Zheng, and Chun-Hsien Chen. 2020. A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives. *Journal of Intelligent Manufacturing* 31, 6 (2020), 1313–1337. (Cited on page 11.)

[31] Fei Tao, Fangyuan Sui, Ang Liu, Qinglin Qi, Meng Zhang, Boyang Song, Zirong Guo, Stephen C-Y Lu, and Andrew YC Nee. 2019. Digital twin-driven product design framework. *International Journal of Production Research* 57, 12 (2019), 3935–3953. https://doi.org/10.1080/00207543.2018.1443229 (Not cited.)

[32] Rainer Stark, Carina Fresemann, and Kai Lindow. 2019. Development and operation of Digital Twins for technical systems and services. *CIRP Annals* 68, 1 (2019), 129–132. (Cited on page 11.)

[33] Adil Rasheed, Omer San, and Trond Kvamsdal. 2019. Digital twin: Values, challenges and enablers. *arXiv preprint arXiv:1910.01719* (2019). (Cited on page 11.)

[34] Jan-Frederik Uhlenkamp, Karl Hribernik, Stefan Wellsandt, and Klaus-Dieter Thoben. 2019. Digital Twin Applications: A first systemization of their dimensions. In *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. IEEE, 1–8. https://doi.org/10.1109/ICE.2019.8792579 (Not cited.)

[35] Sebastian Haag and Reiner Anderl. 2018. Digital twin–Proof of concept. *Manufacturing Letters* 15 (2018), 64–66. (Not cited.)

[36] Tao Fei, Cheng Jiangfeng, Qi Qinglin, Meng Zhang, He Zhang, and Sui Fangyuan. 2018. Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology* 94, 9-12 (2018), 3563–3576. (Cited on page 11.)

[37] Barbara Rita Barricelli, Elena Casiraghi, and Daniela Fogli. 2019. A survey on digital twin: definitions, characteristics, applications, and design implications. *IEEE access* 7 (2019), 167653–167671. (Cited on page 9.)

[38] Paul Gardner, Mattia Dal Borgo, Valentina Ruffini, Aidan J Hughes, Yichen Zhu, and David J Wagg. 2020. Towards the development of an operational digital twin. *Vibration* 3, 3 (2020), 235–265. (Cited on pages 10, 11, 12, 13, 24 and 31.)

[39] P Aivaliotis, K Georgoulias, Z Arkouli, and S Makris. 2019. Methodology for enabling digital twin using advanced physics-based modelling in predictive maintenance. *Procedia Cirp* 81 (2019), 417–422. (Cited on pages 10 and 13.)

[40] Giuseppe Abbiati, Stefano Marelli, Connor Ligeikis, Richard Christenson, and Božidar Stojadinović. 2022. Training of a Classifier for Structural Component Failure based on Hybrid Simulation and Kriging. *Journal of Engineering Mechanics* 148, 1 (2022). (Cited on page 12.)

[41] TG Ritto and FA Rochinha. 2021. Digital twin, physics-based model, and machine learning applied to damage detection in structures. *Mechanical Systems and Signal Processing* 155 (2021), 107614. (Cited on page 12.)

[42] Keith Worden, Lawrence A Bull, Paul Gardner, Julian Gosliga, Timothy J Rogers, Elizabeth J Cross, Evangelos Papatheou, Weijiang Lin, and Nikolaos Dervilis. 2020. A brief introduction to recent developments in population-based structural health monitoring. *Frontiers in Built Environment* 6 (2020), 146. (Cited on pages 12, 34, 82, 83, 138 and 139.)

[43] LA Bull, PA Gardner, J Gosliga, TJ Rogers, N Dervilis, EJ Cross, E Papatheou, AE Maguire, C Campos, and K Worden. 2021. Foundations of population-based SHM, Part I: Homogeneous populations and forms. *Mechanical Systems and Signal Processing* 148 (2021), 107141. (Cited on pages 12, 34, 82, 138 and 139.)

[44] Qinglin Qi, Fei Tao, Ying Zuo, and Dongming Zhao. 2018. Digital twin service towards smart manufacturing. *Procedia Cirp* 72 (2018), 237–242. (Cited on page 12.)

[45] Yi Cai, Binil Starly, Paul Cohen, and Yuan-Shin Lee. 2017. Sensor data and information fusion to construct digital-twins virtual machine tools for cyber-physical manufacturing. *Procedia manufacturing* 10 (2017), 1031–1042. (Cited on page 12.)

[46] Zhiming Zhang and Chao Sun. 2021. Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating. *Structural Health Monitoring* 20, 4 (2021), 1675–1688. (Cited on page 12.)

[47] Pranav M Karve, Yulin Guo, Berkcan Kapusuzoglu, Sankaran Mahadevan, and Mulugeta A Haile. 2020. Digital twin approach for damage-tolerant mission planning under uncertainty. *Engineering Fracture Mechanics* 225 (2020). (Cited on pages 12 and 31.)

[48] Michael G Kapteyn, David J Knezevic, and Karen Willcox. 2020. Toward predictive digital twins via component-based reduced-order models and interpretable machine learning. In *AIAA Scitech 2020 Forum*. 0418. (Cited on page 13.)

[49] Jyotishman Ghosh, Stéphane Foulard, and Rafael Fietzek. 2017. *Vehicle mass estimation from CAN data and drivetrain torque observer*. Technical Report. SAE Technical Paper. (Cited on page 13.)

[50] Yang Hua, Xinhua Liu, Sida Zhou, Yi Huang, Heping Ling, and Shichun Yang. 2021. Toward sustainable reuse of retired lithium-ion batteries from electric vehicles. *Resources, Conservation and Recycling* 168 (2021), 105249. (Cited on page 14.)

[51] Mikel Ayani, Maria Ganebäck, and Amos HC Ng. 2018. Digital Twin: Applying emulation for machine reconditioning. *Procedia Cirp* 72 (2018), 243–248. (Cited on page 14.)

[52] Tom M Mitchell and Tom M Mitchell. 1997. *Machine learning.* Vol. 1. McGraw-hill New York. (Cited on page 14.)

[53] Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective.* MIT press. (Cited on pages 14 and 22.)

[54] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction.* MIT press. (Cited on page 15.)

[55] Ethem Alpaydin. 2020. *Introduction to machine learning.* MIT press. (Cited on page 15.)

[56] Pedro Domingos. 2012. A few useful things to know about machine learning. *Commun. ACM* 55, 10 (2012), 78–87. (Cited on pages 15 and 23.)

[57] Danilo Bzdok, Martin Krzywinski, and Naomi Altman. 2017. Machine learning: a primer. *Nature methods* 14, 12 (2017), 1119. (Cited on page 15.)

[58] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre GR Day, Clint Richardson, Charles K Fisher, and David J Schwab. 2019. A high-bias, low-variance introduction to machine learning for physicists. *Physics reports* 810 (2019), 1–124. (Cited on pages 16, 21 and 23.)

[59] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning.* MIT press. (Cited on pages 17 and 18.)

[60] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. 2015. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery* 8, 1 (2015), 014008. (Not cited.)

[61] Matthias Feurer and Frank Hutter. 2019. Hyperparameter optimization. In *Automated machine learning.* Springer, Cham, 3–33. (Cited on page 17.)

[62] Michael A Nielsen. 2015. *Neural networks and deep learning.* Vol. 25. Determination press San Francisco, CA. (Cited on pages 17 and 18.)

[63] Russell Reed and Robert J MarksII. 1999. *Neural smithing: supervised learning in feedforward artificial neural networks.* Mit Press. (Cited on page 17.)

[64] Kevin P Murphy. 2022. *Probabilistic machine learning: an introduction.* MIT press. (Cited on pages 17 and 18.)

[65] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444. (Cited on pages 17, 18, 19 and 130.)

[66] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. 2018. Automatic differentiation in machine learning: a survey. *Journal of machine learning research* 18 (2018). (Cited on pages 17 and 31.)

[67] Francois Chollet. 2021. *Deep learning with Python*. Simon and Schuster. (Cited on pages 17, 18, 19 and 22.)

[68] Terrence J Sejnowski. 2018. *The deep learning revolution*. MIT press. (Cited on page 18.)

[69] Christian Janiesch, Patrick Zschech, and Kai Heinrich. 2021. Machine learning and deep learning. *Electronic Markets* 31, 3 (2021), 685–695. (Cited on page 18.)

[70] Yaguo Lei, Bin Yang, Xinwei Jiang, Feng Jia, Naipeng Li, and Asoke K Nandi. 2020. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing* 138 (2020), 106587. (Cited on pages 18, 34, 83, 129, 130, 131, 132, 137, 138, 139, 140 and 141.)

[71] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* (2021). (Cited on page 18.)

[72] Simon Bachstein. 2019. *Uncertainty quantification in deep learning*. Master's thesis. ULM University. (Cited on page 18.)

[73] SN Sivanandam and SN Deepa. 2006. *Introduction to neural networks using Matlab 6.0*. Tata McGraw-Hill Education. (Cited on page 19.)

[74] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 2012. Efficient backprop. In *Neural networks: Tricks of the trade*. Springer, 9–48. (Cited on page 19.)

[75] Md Zahangir Alom, Tarek M Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C Van Essen, Abdul AS Awwal, and Vijayan K Asari. 2019. A state-of-the-art survey on deep learning theory and architectures. *Electronics* 8, 3 (2019), 292. (Cited on page 19.)

[76] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117. (Cited on page 19.)

[77] Carl Edward Rasmussen. 2003. Gaussian processes in machine learning. In *Summer school on machine learning*. Springer, 63–71. (Cited on page 20.)

[78] Robert B Gramacy. 2020. *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. Chapman and Hall/CRC. (Cited on page 20.)

[79] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32. (Cited on pages 20 and 64.)

[80] Wei-Yin Loh. 2011. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery* 1, 1 (2011), 14–23. (Cited on page 20.)

[81] Gilles Louppe. 2014. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502* (2014). (Cited on page 21.)

[82] Christopher M Bishop. 2006. Pattern recognition. *Machine learning* 128, 9 (2006). (Cited on page 21.)

[83] Peter Roßbach. 2018. Neural Networks vs. Random Forests–Does it always have to be Deep Learning. *Germany: Frankfurt School of Finance and Management* (2018). (Cited on page 22.)

[84] Wenqi Chai. 2018. *Global sensitivity analysis on vibro-acoustic composite materials with parametric dependency*. Ph. D. Dissertation. Lyon. (Cited on page 22.)

[85] W Chai, A Saidi, Abdelmalek Zine, C Droz, W You, and Mohamed Ichchou. 2020. Comparison of uncertainty quantification process using statistical and data mining algorithms. *Structural and Multidisciplinary Optimization* 61, 2 (2020), 587–598. (Cited on pages 23 and 26.)

[86] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. 2013. Understanding variable importances in forests of randomized trees. *Advances in neural information processing systems* 26 (2013). (Cited on pages 22, 25, 26, 62 and 64.)

[87] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. ACM, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785 (Cited on page 22.)

[88] Yoav Freund and Robert E Schapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. System Sci.* 55, 1 (1997), 119–139. https://doi.org/10.1006/jcss.1997.1504 (Cited on page 22.)

[89] Maliki Moustapha. 2016. *Adaptive surrogate models for the reliable lightweight design of automotive body structures*. Ph. D. Dissertation. Université Blaise Pascal-Clermont-Ferrand II. (Cited on pages 22 and 29.)

[90] Morteza Kiani and Ali R Yildiz. 2016. A comparative study of non-traditional methods for vehicle crashworthiness and NVH optimization. *Archives of Computational Methods in Engineering* 23, 4 (2016), 723–734. (Cited on page 23.)

[91] Kyung-Joon Cha, Chung-Un Chin, Je-Seon Ryu, and Jae-Eung Oh. 2004. The optimal design for low noise intake system using Kriging method with robust design. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing* 47, 3 (2004), 873–881. (Cited on page 27.)

[92] Andrea Bacigalupo, Giorgio Gnecco, Marco Lepidi, and Luigi Gambarotta. 2020. Machine-learning techniques for the optimal design of acoustic metamaterials. *Journal of Optimization Theory and Applications* 187, 3 (2020), 630–653. (Cited on pages 23 and 27.)

[93] Thomas Diestmann, Nils Broedling, Benedict Götz, and Tobias Melz. 2021. Surrogate Model-Based Uncertainty Quantification for a Helical Gear Pair. In *International Conference on Uncertainty in Mechanical Engineering*. Springer, 191–207. (Cited on pages 23 and 26.)

[94] Giuseppe Ciaburro and Gino Iannace. 2022. Membrane-type acoustic metamaterial using cork sheets and attached masses based on reused materials. *Applied Acoustics* 189 (2022), 108605. (Cited on page 23.)

[95] Dimitrios Ernst Tsokaktsidis, Timo Von Wysocki, Frank Gauterin, and Steffen Marburg. 2019. *Artificial Neural Network predicts noise transfer as a function of excitation and geometry*. Universitätsbibliothek der RWTH Aachen. (Cited on page 23.)

[96] Richard P Dwight, Jouke de Baar, and Iliass Azijli. 2012. A Tutorial on Adaptive Surrogate Modelling for Global Optimization. *Dimension* 8 (2012), 7. (Cited on page 24.)

[97] Haitao Liu, Yew Soon Ong, and Jianfei Cai. 2018. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization* 57, 1 (2018), 393–416. https://doi.org/10.1007/s00158-017-1739-8 (Cited on pages 24 and 28.)

[98] Fenfen Xiong, Shishi Chen, and Ying Xiong. 2014. Dynamic system uncertainty propagation using polynomial chaos. *Chinese Journal of Aeronautics* 27, 5 (2014), 1156–1170. (Cited on page 24.)

[99] Bruno Sudret. 2008. Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety* 93, 7 (2008), 964–979. (Cited on page 24.)

[100] Shahram Azadi, Mohammad Azadi, and Farshad Zahedi. 2009. NVH analysis and improvement of a vehicle body structure using DOE method. *Journal*

of mechanical science and technology 23, 11 (2009), 2980–2989. (Cited on pages 24 and 27.)

[101] H-M Gutmann. 2001. A radial basis function method for global optimization. *Journal of global optimization* 19, 3 (2001), 201–227. (Cited on page 24.)

[102] Morteza Kiani and Ali R Yildiz. 2016. A comparative study of non-traditional methods for vehicle crashworthiness and NVH optimization. *Archives of Computational Methods in Engineering* 23, 4 (2016), 723–734. (Cited on pages 24 and 27.)

[103] Bruno Sudret, Stefano Marelli, and Joe Wiart. 2017. Surrogate models for uncertainty quantification: An overview. In *2017 11th European conference on antennas and propagation (EUCAP)*. IEEE, 793–797. (Cited on pages 24, 25 and 26.)

[104] Maliki Moustapha and Bruno Sudret. 2019. Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework. *Structural and Multidisciplinary Optimization* (2019), 1–20. (Cited on pages 24 and 29.)

[105] Souvik Chakraborty, Sondipon Adhikari, and Ranjan Ganguli. 2021. The role of surrogate models in the development of digital twins of dynamic systems. *Applied Mathematical Modelling* 90 (2021), 662–681. (Cited on page 24.)

[106] Maria Böttcher, Ferenc Leichsenring, Alexander Fuchs, Wolfgang Graf, and Michael Kaliske. 2021. Efficient Utilization of Surrogate Models for Uncertainty Quantification. *PAMM* 20, 1 (2021). (Cited on page 24.)

[107] Barbara Zaparoli Cunha, Abdel-Malek Zine, Mohamed Ichchou, Christophe Droz, and Stéphane Foulard. 2022. On Machine-Learning-Driven Surrogates for Sound Transmission Loss Simulations. *Applied Sciences* 12, 21 (2022), 10727. (Cited on pages 24, 37, 61 and 125.)

[108] Zhihong Liu, Tobias Eichenlaub, and Stephan Rinderknecht. 2023. A survey of sequential adaptive sampling strategy for transmission power loss measurement. *Mechanical Systems and Signal Processing* 183 (2023), 109644. (Cited on page 24.)

[109] Maosen Cao, Nizar F Alkayem, Lixia Pan, and Drahomír Novák. 2016. Advanced methods in neural networks-based sensitivity analysis with their applications in civil engineering. *Artificial neural networks: models and applications, Rijeka, Croatia, IntechOpen* (2016), 335–353. (Cited on page 25.)

[110] Loic Le Gratiet, Stefano Marelli, and Bruno Sudret. 2017. Metamodel-based sensitivity analysis: polynomial chaos expansions and Gaussian processes. In *Handbook of uncertainty quantification*. Springer, 1289–1325. (Cited on page 25.)

[111] Kai Cheng, Zhenzhou Lu, Chunyan Ling, and Suting Zhou. 2020. Surrogate-assisted global sensitivity analysis: an overview. *Structural and Multidisciplinary Optimization* 61, 3 (2020), 1187–1213. (Cited on page 25.)

[112] Jaime Pizarroso, José Portela, and Antonio Muñoz. 2020. NeuralSens: Sensitivity Analysis of Neural Networks. *arXiv preprint arXiv:2002.11423* (2020). (Cited on page 25.)

[113] Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B Fox. 2021. Neural Granger Causality. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 01 (2021), 1–1. (Cited on page 25.)

[114] Moritz Böhle, Fabian Eitel, Martin Weygandt, and Kerstin Ritter. 2019. Layer-wise relevance propagation for explaining deep neural network decisions in MRI-based Alzheimer's disease classification. *Frontiers in aging neuroscience* 11 (2019), 194. (Not cited.)

[115] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10, 7 (2015), e0130140. (Cited on page 25.)

[116] Giuseppe Abbiati, Stefano Marelli, Nikolaos Tsokanas, Bruno Sudret, and B Stojadinović. 2021. A global sensitivity analysis framework for hybrid simulation. *Mechanical Systems and Signal Processing* 146 (2021), 106997. (Cited on page 26.)

[117] Merten Stender, Christian Adams, Mathies Wedler, Antje Grebel, and Nobert Hoffmann. 2021. Explainable machine learning determines effects on the sound absorption coefficient measured in the impedance tube. *The Journal of the Acoustical Society of America* 149, 3 (2021), 1932–1945. (Cited on page 26.)

[118] Christian Soize. 2017. *Uncertainty quantification.* Springer. (Cited on page 26.)

[119] Amir Nobari, Huajiang Ouyang, and Paul Bannister. 2015. Uncertainty quantification of squeal instability via surrogate modelling. *Mechanical Systems and Signal Processing* 60 (2015), 887–908. (Cited on page 26.)

[120] Qing Guo, Yongshou Liu, Yuzhen Zhao, Baohui Li, and Qin Yao. 2019. Improved resonance reliability and global sensitivity analysis of multi-span pipes conveying fluid based on active learning Kriging model. *International Journal of Pressure Vessels and Piping* 170 (2019), 92–101. (Cited on page 26.)

[121] Qing Guo, Yongshou Liu, Bingqian Chen, and Qin Yao. 2021. A variable and mode sensitivity analysis method for structural system using a novel active learning Kriging model. *Reliability Engineering & System Safety* 206

(2021), 107285. https://doi.org/10.1016/j.ress.2020.107285 (Cited on page 26.)

[122] Jorge E Hurtado and Diego A Alvarez. 2001. Neural-network-based reliability analysis: a comparative study. *Computer methods in applied mechanics and engineering* 191, 1-2 (2001), 113–132. (Cited on page 27.)

[123] Liqun Wang, Zengtao Chen, and Guolai Yang. 2020. An interval uncertainty analysis method for structural response bounds using feedforward neural network differentiation. *Applied Mathematical Modelling* 82 (2020), 449–468. (Cited on page 27.)

[124] Jia-Qi Liu, Yun-Wen Feng, Xiao-Feng Xue, and Cheng Lu. 2021. Intelligent Extremum Surrogate Modeling Framework for Dynamic Probabilistic Analysis of Complex Mechanism. *Mathematical Problems in Engineering* 2021 (2021). (Cited on page 27.)

[125] Cheng Lu, Cheng-Wei Fei, Yun-Wen Feng, Yong-Jun Zhao, Xiao-Wei Dong, and Yat-Sze Choy. 2021. Probabilistic analyses of structural dynamic response with modified Kriging-based moving extremum framework. *Engineering Failure Analysis* 125 (2021), 105398. (Not cited.)

[126] Cheng Lu, Yun-Wen Feng, Rhea P Liem, and Cheng-Wei Fei. 2018. Improved kriging with extremum response surface method for structural dynamic reliability and sensitivity analyses. *Aerospace Science and Technology* 76 (2018), 164–175. (Cited on page 27.)

[127] Weizhen You, Alexandre Saidi, Abdel-malek Zine, and Mohamed Ichchou. 2020. Mechanical Reliability Assessment by Ensemble Learning. *Vehicles* 2, 1 (2020), 126–141. (Cited on page 27.)

[128] Jun Lu, Zhenfei Zhan, Haozhan Song, Xu Liu, Xin Yang, and Junqi Yang. 2017. *Design optimization of vehicle body NVH performance based on dynamic response analysis*. Technical Report. SAE Technical Paper. (Cited on page 27.)

[129] Jianhua Zhang, Wuli Chu, Jinghui Zhang, and Yi Lv. 2019. Vibroacoustic optimization study for the volute casing of a centrifugal fan. *Applied Sciences* 9, 5 (2019), 859. (Cited on page 27.)

[130] Inseok Park and Dimitrios Papadimitriou. 2020. Efficient Surrogate-Based NVH Optimization of a Full Vehicle Using FRF Based Substructuring. *SAE International Journal of Advances and Current Practices in Mobility* 2, 2020-01-0629 (2020), 1429–1442. (Cited on page 27.)

[131] Timo von Wysocki, Frank Rieger, Dimitrios Ernst Tsokaktsidis, and Frank Gauterin. 2021. Generating Component Designs for an Improved NVH Performance by Using an Artificial Neural Network as an Optimization Metamodel.

Designs 5, 2 (2021). https://doi.org/10.3390/designs5020036 (Cited on page 27.)

[132] Timo von Wysocki, Michael Leupolz, and Frank Gauterin. 2020. Metamodels Resulting from Two Different Geometry Morphing Approaches Are Suitable to Direct the Modification of Structure-Born Noise Transfer in the Digital Design Phase. *Applied System Innovation* 3, 4 (2020), 47. (Cited on page 27.)

[133] Min Li, Wei Zhou, Jiang Liu, Xilong Zhang, Fuquan Pan, Huan Yang, Mengshan Li, and Dijia Luo. 2021. Vehicle Interior Noise Prediction Based on Elman Neural Network. *Applied Sciences* 11, 17 (2021). https://doi.org/10.3390/app11178029 (Cited on page 27.)

[134] Dimitrios Ernst Tsokaktsidis, Clemens Nau, and Steffen Marburg. 2020. *Time Domain Full Vehicle Interior Noise Calculation from Component Level Data by Machine Learning.* Technical Report. SAE Technical Paper. (Cited on page 27.)

[135] Issah Ibrahim, Rodrigo Silva, MH Mohammadi, Vahid Ghorbanian, and David A Lowther. 2020. Surrogate-based acoustic noise prediction of electric motors. *IEEE Transactions on Magnetics* 56, 2 (2020), 1–4. (Cited on pages 27 and 62.)

[136] Alessandro Casaburo, Dario Magliacano, Giuseppe Petrone, Francesco Franco, and Sergio De Rosa. 2021. Gaussian-based machine learning algorithm for the design and characterization of a porous meta-material for acoustic applications. *Applied Sciences* 12, 1 (2021), 333. (Cited on page 27.)

[137] Eric Brochu, Vlad M Cora, and Nando De Freitas. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599* (2010). (Cited on page 28.)

[138] Roman Garnett. 2022. *Bayesian Optimization.* Cambridge University Press. (Cited on page 28.)

[139] Donald R Jones, Matthias Schonlau, and William J Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13, 4 (1998), 455–492. (Cited on page 28.)

[140] Pradeep Mohanasundaram, Frédéric Gillot, Koji Shimoyama, and Sébastien Besset. 2020. Shape optimization of a disc-pad system under squeal noise criteria. *SN Applied Sciences* 2, 4 (2020), 1–15. (Cited on page 28.)

[141] Dahua Du, Erming He, Feng Li, and Daoqiong Huang. 2020. Using the hierarchical Kriging model to optimize the structural dynamics of rocket engines. *Aerospace Science and Technology* 107 (2020), 106248. https://doi.org/10.1016/j.ast.2020.106248 (Cited on page 28.)

[142] Andrea Bacigalupo, Giorgio Gnecco, Marco Lepidi, and Luigi Gambarotta. 2021. Computational design of innovative mechanical metafilters via adaptive surrogate-based optimization. *Computer Methods in Applied Mechanics and Engineering* 375 (2021), 113623. (Cited on page 28.)

[143] Cheng-Wei Fei, Wen-Zhong Tang, and Guang-Chen Bai. 2014. Novel method and model for dynamic reliability optimal design of turbine blade deformation. *Aerospace Science and Technology* 39 (2014), 588–595. (Cited on page 29.)

[144] Chun-Yi Zhang, Zhe-Shan Yuan, Ze Wang, Cheng-Wei Fei, and Cheng Lu. 2019. Probabilistic fatigue/creep optimization of turbine bladed disk with fuzzy multi-extremum response surface method. *Materials* 12, 20 (2019), 3367. (Cited on page 29.)

[145] Fábio FS Nascentes, Rafael H Lopez, Jose Eduardo S Cursi, Rubens Sampaio, and Leandro FF Miguel. 2018. An efficient global optimization approach for reliability maximization of friction-tuned mass damper-controlled structures. *Shock and Vibration* 2018 (2018). (Cited on page 29.)

[146] Sourav Das, Solomon Tesfamariam, Yangyang Chen, Zhichao Qian, Ping Tan, and Fulin Zhou. 2020. Reliability-based optimization of nonlinear energy sink with negative stiffness and sliding friction. *Journal of Sound and Vibration* 485 (2020), 115560. https://doi.org/10.1016/j.jsv.2020.115560 (Cited on page 29.)

[147] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. 2017. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering* 29, 10 (2017), 2318–2331. (Cited on page 30.)

[148] Rui Wang and Rose Yu. 2021. Physics-guided deep learning for dynamical systems: A survey. *arXiv preprint arXiv:2107.01272* (2021). (Cited on page 30.)

[149] Matthew E Levine and Andrew M Stuart. 2021. A framework for machine learning of model error in dynamical systems. *arXiv preprint arXiv:2107.06658* (2021). (Cited on pages 30 and 31.)

[150] Andrew C. Miller, Nicholas J. Foti, and Emily B. Fox. 2021. Breiman's two cultures: You don't have to choose sides. arXiv:2104.12219 [stat.ML] (Cited on page 30.)

[151] Rahul Rai and Chandan K Sahu. 2020. Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (cps) focus. *IEEE Access* 8 (2020), 71050–71073. (Cited on page 30.)

[152] Rui Wang and Rose Yu. 2021. Physics-guided deep learning for dynamical systems: A survey. *arXiv preprint arXiv:2107.01272* (2021). (Cited on page 30.)

[153] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. 2017. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431* (2017). (Cited on pages 31 and 33.)

[154] Urban Forssell and Peter Lindskog. 1997. Combining semi-physical and neural network modeling: An example ofits usefulness. *IFAC Proceedings Volumes* 30, 11 (1997), 767–770. (Cited on page 31.)

[155] Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. 2021. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences* 118, 21 (2021). (Cited on page 33.)

[156] Yang Yu, Houpu Yao, and Yongming Liu. 2019. Aircraft dynamics simulation using a novel physics-based learning method. *Aerospace Science and Technology* 87 (2019), 254–264. (Not cited.)

[157] J Nagoor Kani and Ahmed H Elsheikh. 2017. DR-RNN: A deep residual recurrent neural network for model reduction. *arXiv preprint arXiv:1709.00939* (2017). (Cited on page 31.)

[158] Abhinav Gupta and Pierre F. J. Lermusiaux. 2021. Neural closure models for dynamical systems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 477, 2252 (Aug 2021), 20201004. https://doi.org/10.1098/rspa.2020.1004 (Cited on page 31.)

[159] JA Wilson and LFM Zorzetto. 1997. A generalised approach to process state estimation using hybrid artificial neural network/mechanistic models. *Computers & chemical engineering* 21, 9 (1997), 951–963. (Cited on page 31.)

[160] Eric J Parish and Karthik Duraisamy. 2016. A paradigm for data-driven predictive modeling using field inversion and machine learning. *J. Comput. Phys.* 305 (2016), 758–774. (Cited on page 31.)

[161] Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy. 2017. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA journal* 55, 7 (2017), 2215–2227. (Not cited.)

[162] JFH Buist, Benjamin Sanderse, Yous van Halder, Barry Koren, and GertJan van Heijst. 2019. Machine Learning for Closure Models in Multiphase-Flow Applications. *Crete, Greece* (2019). (Not cited.)

[163] Brendan D Tracey, Karthikeyan Duraisamy, and Juan J Alonso. 2015. A machine learning strategy to assist turbulence model development. In *53rd AIAA aerospace sciences meeting*. 1287. (Cited on page 31.)

[164] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan Read, Jacob Zwart, Michael Steinbach, and Vipin Kumar. 2019. Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 558–566. (Cited on pages 31 and 33.)

[165] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan S Read, Jacob A Zwart, Michael Steinbach, and Vipin Kumar. 2021. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS Transactions on Data Science* 2, 3 (2021), 1–26. (Cited on page 33.)

[166] Jordan S Read, Xiaowei Jia, Jared Willard, Alison P Appling, Jacob A Zwart, Samantha K Oliver, Anuj Karpatne, Gretchen JA Hansen, Paul C Hanson, William Watkins, et al. 2019. Process-guided deep learning predictions of lake water temperature. *Water Resources Research* 55, 11 (2019), 9173–9190. (Not cited.)

[167] Arka Daw, R Quinn Thomas, Cayelan C Carey, Jordan S Read, Alison P Appling, and Anuj Karpatne. 2020. Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling. In *Proceedings of the 2020 siam international conference on data mining*. SIAM, 532–540. (Cited on pages 32 and 33.)

[168] Ruiyang Zhang, Yang Liu, and Hao Sun. 2020. Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling. *Engineering Structures* 215 (2020), 110704. https://doi.org/10.1016/j.engstruct.2020.110704 (Not cited.)

[169] Sadegh Karimpouli and Pejman Tahmasebi. 2020. Physics informed machine learning: Seismic wave equation. *Geoscience Frontiers* 11, 6 (2020), 1993–2001. https://doi.org/10.1016/j.gsf.2020.07.007 (Not cited.)

[170] Jian Sun, Kristopher A Innanen, and Chao Huang. 2021. Physics-guided deep learning for seismic inversion with hybrid training and uncertainty analysis. *Geophysics* 86, 3 (2021), R303–R317. (Cited on page 31.)

[171] M. Raissi, P. Perdikaris, and G.E. Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378 (2019), 686–707. https://doi.org/10.1016/j.jcp.2018.10.045 (Cited on page 31.)

[172] Ameya D. Jagtap, Ehsan Kharazmi, and George Em Karniadakis. 2020. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering* 365 (2020), 113028. https://doi.org/10.1016/j.cma.2020.113028 (Not cited.)

[173] Guofei Pang, Lu Lu, and George Em Karniadakis. 2019. fPINNs: Fractional physics-informed neural networks. *SIAM Journal on Scientific Computing* 41, 4 (2019), A2603–A2626. (Not cited.)

[174] Zhiping Mao, Ameya D. Jagtap, and George Em Karniadakis. 2020. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering* 360 (2020), 112789. https://doi.org/10.1016/j.cma.2019.112789 (Not cited.)

[175] Xuhui Meng, Zhen Li, Dongkun Zhang, and George Em Karniadakis. 2020. PPINN: Parareal physics-informed neural network for time-dependent PDEs. *Computer Methods in Applied Mechanics and Engineering* 370 (2020), 113250. https://doi.org/10.1016/j.cma.2020.113250 (Not cited.)

[176] Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. 2021. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *J. Comput. Phys.* 426 (2021), 109951. (Not cited.)

[177] Dehao Liu and Yan Wang. 2019. Multi-fidelity physics-constrained neural network and its application in materials modeling. *Journal of Mechanical Design* 141, 12 (2019). (Not cited.)

[178] Yuyao Chen, Lu Lu, George Em Karniadakis, and Luca Dal Negro. 2020. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics express* 28, 8 (2020), 11618–11633. (Not cited.)

[179] Zhiwei Fang and Justin Zhan. 2019. Deep physical informed neural networks for metamaterial design. *IEEE Access* 8 (2019), 24506–24513. (Not cited.)

[180] Maziar Raissi, Hessam Babaee, and Peyman Givi. 2019. Deep learning of turbulent scalar mixing. *Physical Review Fluids* 4, 12 (2019), 124501. (Not cited.)

[181] Hongwei Guo, Xiaoying Zhuang, and Timon Rabczuk. 2021. A deep collocation method for the bending analysis of Kirchhoff plate. *arXiv preprint arXiv:2102.02617* (2021). (Cited on page 31.)

[182] Jie Chen and Yongming Liu. 2021. Probabilistic physics-guided machine learning for fatigue data analysis. *Expert Systems with Applications* 168 (2021), 114316. https://doi.org/10.1016/j.eswa.2020.114316 (Cited on page 32.)

[183] Nikhil Muralidhar, Jie Bu, Ze Cao, Long He, Naren Ramakrishnan, Danesh Tafti, and Anuj Karpatne. 2019. Physics-guided design and learning of neural networks for predicting drag force on particle suspensions in moving fluids. *arXiv preprint arXiv:1911.04240* (2019). (Cited on page 32.)

[184] Ruiyang Zhang, Yang Liu, and Hao Sun. 2020. Physics-informed multi-LSTM networks for metamodeling of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering* 369 (2020), 113226. (Cited on page 32.)

[185] Marcus M Noack and James A Sethian. 2022. Advanced stationary and nonstationary kernel designs for domain-aware gaussian processes. *Communications in Applied Mathematics and Computational Science* 17, 1 (2022), 131–156. (Cited on page 32.)

[186] Elizabeth J Cross and Timothy J Rogers. 2021. Physics-derived covariance functions for machine learning in structural dynamics. *IFAC-PapersOnLine* 54, 7 (2021), 168–173. (Cited on page 32.)

[187] Elizabeth J Cross, SJ Gibson, MR Jones, DJ Pitchforth, S Zhang, and TJ Rogers. 2022. Physics-Informed Machine Learning for Structural Health Monitoring. In *Structural Health Monitoring Based on Data Science Techniques*. Springer, 347–367. (Cited on pages 32 and 138.)

[188] Juš Kocijan. 2012. Dynamic GP models: an overview and recent developments. In *Proceedings of 6th International Conference on Applied Mathematics, Simulation and Modelling*. 38–43. (Cited on pages 32 and 144.)

[189] Steindor Saemundsson, Alexander Terenin, Katja Hofmann, and Marc Peter Deisenroth. 2020. Variational Integrator Networks for Physically Structured Embeddings. arXiv:1910.09349 [stat.ML] (Cited on pages 32 and 33.)

[190] In Huh, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. 2021. Time-Reversal Symmetric ODE Network. arXiv:2007.11362 [cs.LG] (Not cited.)

[191] Aleksandar Botev, Andrew Jaegle, Peter Wirnsberger, Daniel Hennes, and Irina Higgins. 2021. Which priors matter? Benchmarking models for learning latent dynamics. *arXiv preprint arXiv:2111.05458* (2021). (Not cited.)

[192] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. 2019. Augmented neural odes. *arXiv preprint arXiv:1904.01681* (2019). (Not cited.)

[193] Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. 2020. Dissecting neural odes. *arXiv preprint arXiv:2002.08071* (2020). (Cited on page 32.)

[194] Michael Lutter and Jan Peters. 2021. Combining Physics and Deep Learning to learn Continuous-Time Dynamics Models. *arXiv preprint arXiv:2110.01894* (2021). (Cited on page 33.)

[195] Sam Greydanus, Misko Dzamba, and Jason Yosinski. 2019. Hamiltonian Neural Networks. arXiv:1906.01563 [cs.NE] (Cited on page 33.)

[196] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. 2019. Symplectic ode-net: Learning hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077* (2019). (Not cited.)

[197] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. 2020. Dissipative symoden: Encoding hamiltonian dynamics with dissipation and control into deep learning. *arXiv preprint arXiv:2002.08860* (2020). (Not cited.)

[198] Zhengdao Chen, Jianyu Zhang, Martin Arjovsky, and Léon Bottou. 2019. Symplectic recurrent neural networks. *arXiv preprint arXiv:1909.13334* (2019). (Not cited.)

[199] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. 2020. Lagrangian Neural Networks. arXiv:2003.04630 [cs.LG] (Not cited.)

[200] Michael Lutter, Christian Ritter, and Jan Peters. 2019. Deep lagrangian networks: Using physics as model prior for deep learning. *arXiv preprint arXiv:1907.04490* (2019). (Not cited.)

[201] Marc Finzi, Ke Alexander Wang, and Andrew Gordon Wilson. 2020. Simplifying hamiltonian and lagrangian neural networks via explicit constraints. *arXiv preprint arXiv:2010.13581* (2020). (Not cited.)

[202] Ravinder Bhattoo, Sayan Ranu, and NM Krishnan. 2021. Lagrangian Neural Network with Differentiable Symmetries and Relational Inductive Bias. *arXiv preprint arXiv:2110.03266* (2021). (Cited on page 33.)

[203] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. 2021. Benchmarking energy-conserving neural networks for learning dynamics from data. In *Learning for Dynamics and Control*. PMLR, 1218–1229. (Cited on page 33.)

[204] Yunhao Ba, Guangyuan Zhao, and Achuta Kadambi. 2019. Blending diverse physical priors with neural networks. *arXiv preprint arXiv:1910.00201* (2019). (Cited on page 33.)

[205] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. 2020. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics* 52 (2020), 477–508. (Cited on pages 33, 142 and 145.)

[206] Suraj Pawar, Omer San, Burak Aksoylu, Adil Rasheed, and Trond Kvamsdal. 2021. Physics guided machine learning using simplified theories. *Physics of Fluids* 33, 1 (2021), 011701. (Cited on page 33.)

[207] K Kashinath, M Mustafa, A Albert, JL Wu, C Jiang, S Esmaeilzadeh, K Azizzadenesheli, R Wang, A Chattopadhyay, A Singh, et al. 2021. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A* 379, 2194 (2021), 20200093. (Cited on page 33.)

[208] Evan Kodra, Udit Bhatia, Snigdhansu Chatterjee, Stone Chen, and Auroop Ratan Ganguly. 2020. Physics-guided probabilistic modeling of extreme precipitation under climate change. *Scientific reports* 10, 1 (2020), 1–11. (Cited on page 33.)

[209] Tom Beucler, Stephan Rasp, Michael Pritchard, and Pierre Gentine. 2019. Achieving conservation of energy in neural network emulators for climate modeling. *arXiv preprint arXiv:1906.06622* (2019). (Cited on page 33.)

[210] Jonathan Schmidt, Mário RG Marques, Silvana Botti, and Miguel AL Marques. 2019. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials* 5, 1 (2019), 1–36. (Cited on page 33.)

[211] Siddhant Kumar, Stephanie Tan, Li Zheng, and Dennis M Kochmann. 2020. Inverse-designed spinodoid metamaterials. *npj Computational Materials* 6, 1 (2020), 1–10. (Not cited.)

[212] Ruijin Cang, Hechao Li, Hope Yao, Yang Jiao, and Yi Ren. 2018. Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model. *Computational Materials Science* 150 (2018), 212–221. (Cited on page 33.)

[213] Yang Yu, Houpu Yao, and Yongming Liu. 2020. Structural dynamics simulation using a novel physics-guided machine learning method. *Engineering Applications of Artificial Intelligence* 96 (2020), 103947. https://doi.org/10.1016/j.engappai.2020.103947 (Cited on page 33.)

[214] Andreas Hochlehnert, Alexander Terenin, Steindór Sæmundsson, and Marc Deisenroth. 2021. Learning contact dynamics using physically structured neural networks. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2152–2160. (Cited on page 33.)

[215] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. 2021. Extending Lagrangian and Hamiltonian Neural Networks with Differentiable Contact Models. *Advances in Neural Information Processing Systems* 34 (2021). (Cited on page 33.)

[216] Yuan Yin, Vincent Le Guen, Jérémie Dona, Emmanuel de Bézenac, Ibrahim Ayed, Nicolas Thome, and Patrick Gallinari. 2020. Augmenting physical models with deep networks for complex dynamics forecasting. *arXiv preprint arXiv:2010.04456* (2020). (Cited on page 33.)

[217] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359. (Cited on pages 34, 82 and 85.)

[218] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76. (Cited on pages 34 and 82.)

[219] J Gosliga, PA Gardner, LA Bull, N Dervilis, and K Worden. 2021. Foundations of Population-based SHM, Part II: Heterogeneous populations–Graphs, networks, and communities. *Mechanical Systems and Signal Processing* 148 (2021), 107144. (Cited on pages 34, 82, 83, 138 and 139.)

[220] P Gardner, LA Bull, J Gosliga, N Dervilis, and K Worden. 2021. Foundations of population-based SHM, Part III: Heterogeneous populations–Mapping and transfer. *Mechanical Systems and Signal Processing* 149 (2021), 107142. (Not cited.)

[221] George Tsialiamanis, Charilaos Mylonas, Eleni Chatzi, Nikolaos Dervilis, David J Wagg, and Keith Worden. 2021. Foundations of population-based SHM, Part IV: The geometry of spaces of structures and their feature spaces. *Mechanical Systems and Signal Processing* 157 (2021), 107692. (Cited on pages 34, 83, 138 and 139.)

[222] Jiaxian Chen, Ruyi Huang, Zhuyun Chen, Wentao Mao, and Weihua Li. 2023. Transfer learning algorithms for bearing remaining useful life prediction: A comprehensive review from an industrial application perspective. *Mechanical Systems and Signal Processing* 193 (2023), 110239. (Cited on pages 34 and 82.)

[223] P Gardner, X Liu, and K Worden. 2020. On the application of domain adaptation in structural health monitoring. *Mechanical Systems and Signal Processing* 138 (2020), 106550. (Cited on pages 34, 83 and 139.)

[224] Zhenghong Wu, Hongkai Jiang, Hongxuan Zhu, and Xin Wang. 2023. A knowledge dynamic matching unit-guided multi-source domain adaptation network with attention mechanism for rolling bearing fault diagnosis. *Mechanical Systems and Signal Processing* 189 (2023), 110098. (Cited on pages 34 and 83.)

[225] Dongdong Wei, Te Han, Fulei Chu, and Ming Jian Zuo. 2021. Weighted domain adaptation networks for machinery fault diagnosis. *Mechanical Systems and Signal Processing* 158 (2021), 107744. (Cited on pages 34 and 83.)

[226] P Gardner, LA Bull, J Gosliga, J Poole, N Dervilis, and K Worden. 2022. A population-based SHM methodology for heterogeneous structures: Transferring damage localisation knowledge between different aircraft wings. *Mechanical Systems and Signal Processing* 172 (2022), 108918. (Cited on pages 34, 83 and 139.)

[227] Li Ai, Bin Zhang, and Paul Ziehl. 2023. A transfer learning approach for acoustic emission zonal localization on steel plate-like structure using numerical simulation and unsupervised domain adaptation. *Mechanical Systems and Signal Processing* 192 (2023), 110216. (Cited on pages 34 and 83.)

[228] Van Hai Trinh, Johann Guilleminot, Camille Perrot, and Viet Dung Vu. 2022. Learning acoustic responses from experiments: A multiscale-informed transfer learning approach. *The Journal of the Acoustical Society of America* 151, 4 (2022), 2587–2601. (Cited on pages 34 and 83.)

[229] Su Jiang and Louis J Durlofsky. 2023. Use of multifidelity training data and transfer learning for efficient construction of subsurface flow surrogate models. *J. Comput. Phys.* 474 (2023), 111800. (Cited on pages 35 and 83.)

[230] Xin Geng, Peiqing Liu, Tianxiang Hu, Qiulin Qu, Jiahua Dai, Changhao Lyu, Yunsong Ge, and Rinie AD Akkermans. 2023. Multi-fidelity optimization of a quiet propeller based on deep deterministic policy gradient and transfer learning. *Aerospace Science and Technology* 137 (2023), 108288. (Cited on pages 35 and 83.)

[231] Jinhyeok Kim and Jongsoo Lee. 2022. Instance-based transfer learning method via modified domain-adversarial neural network with influence function: Applications to design metamodeling and fault diagnosis. *Applied Soft Computing* 123 (2022), 108934. (Cited on pages 35 and 83.)

[232] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. 2022. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022). (Cited on pages 35 and 84.)

[233] Xu Liu, Yingguang Li, Qinglu Meng, and Gengxiang Chen. 2021. Deep transfer learning for conditional shift in regression. *Knowledge-Based Systems* 227 (2021), 107216. (Cited on pages 35, 84, 85 and 86.)

[234] Barbara Zaparoli Cunha, Mohamed Ichchou, Christophe Droz, Abdel-Malek Zine, and Stéphane Foulard. 2022. Interpretable and Physics-Supported Machine Learning Model for Sound Transmission Loss Analysis. In *International Conference on Noise and Vibration Engineering, ISMA2022*. (Cited on pages 37, 61 and 125.)

[235] András Sobester, Alexander Forrester, and Andy Keane. 2008. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons. (Cited on pages 38 and 40.)

[236] Nestor V Queipo, Raphael T Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P Kevin Tucker. 2005. Surrogate-based analysis and optimization. *Progress in aerospace sciences* 41, 1 (2005), 1–28. (Not cited.)

[237] Atharv Bhosekar and Marianthi Ierapetritou. 2018. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers & Chemical Engineering* 108 (2018), 250–267. (Cited on page 38.)

[238] Andrea Santoni, John Laurence Davy, Patrizio Fausti, and Paolo Bonfiglio. 2020. A review of the different approaches to predict the sound transmission loss of building partitions. *Building Acoustics* 27, 3 (2020), 253–279. (Cited on pages 40, 43 and 48.)

[239] Frank J Fahy and Paolo Gardonio. 2007. *Sound and structural vibration: radiation, transmission and response* (2 ed.). Elsevier, Chapter 5, 277–374. (Cited on pages 42, 46, 53 and 67.)

[240] Lothar Cremer. 1942. Theorie der Schalldämmung dünner Wände bei schrägem Einfall. *Akustische Zeitschrift* 7, 3 (1942), 81–104. (Cited on page 46.)

[241] Jean-Loup Christen, Mohamed Ichchou, Abdelmalek Zine, and Bernard Troclet. 2016. Wave finite element formulation of the acoustic transmission through complex infinite plates. *Acta Acustica united with Acustica* 102, 6 (2016), 984–991. (Cited on page 46.)

[242] Jean-Loup Christen, Mohamed Ichchou, Bernard Troclet, Olivier Bareille, and Morvan Ouisse. 2016. Global sensitivity analysis of analytical vibroacoustic transmission models. *Journal of Sound and Vibration* 368 (2016), 121–134. (Cited on pages 46, 47, 62 and 67.)

[243] K Renji. 2005. Sound transmission loss of unbounded panels in bending vibration considering transverse shear deformation. *Journal of Sound and Vibration* 283, 1-2 (2005), 478–486. (Cited on pages 47, 48 and 53.)

[244] K Renji, PS Nair, and S Narayanan. 1997. Critical and coincidence frequencies of flat panels. *Journal of sound and vibration* 205, 1 (1997), 19–32. (Cited on page 48.)

[245] Noureddine Atalla, Franck Sgard, and Celse Kafui Amedin. 2006. On the modeling of sound radiation from poroelastic materials. *The Journal of the Acoustical Society of America* 120, 4 (2006), 1990–1995. (Cited on page 48.)

[246] Dilal Rhazi and Noureddine Atalla. 2010. A simple method to account for size effects in the transfer matrix method. *The journal of the acoustical society of america* 127, 2 (2010), EL30–EL36. (Cited on page 48.)

[247] FX Xin, TJ Lu, and CQ Chen. 2010. Sound transmission through simply supported finite double-panel partitions with enclosed air cavity. *Journal of Vibration and Acoustics* 132, 1 (2010). (Cited on page 48.)

[248] Chong Wang. 2015. Modal sound transmission loss of a single leaf panel: asymptotic solutions. *The Journal of the Acoustical Society of America* 138, 6 (2015), 3964–3975. (Cited on page 48.)

[249] COMSOL AB. Stockholm, Sweden. COMSOL Multiphysics®, Version 5.6. www.comsol.com (Cited on page 50.)

[250] COMSOL AB. 2020. COMSOL Multiphysics Reference Manual. (Cited on page 50.)

[251] Masaaki Imaizumi and Kenji Fukumizu. 2019. Deep neural networks learn non-smooth functions effectively. In *The 22nd international conference on artificial intelligence and statistics*. PMLR, 869–878. (Cited on page 54.)

[252] Carl Edward Rasmussen. 2003. Gaussian processes in machine learning. In *Summer school on machine learning*. Springer, 63–71. (Cited on pages 54 and 87.)

[253] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer. (Cited on page 54.)

[254] Christopher K Williams and Carl Edward Rasmussen. 2006. *Gaussian processes for machine learning*. Vol. 2. MIT press Cambridge, MA. (Cited on page 54.)

[255] Merten Stender, Merten Tiedemann, David Spieler, Daniel Schoepflin, Norbert Hoffmann, and Sebastian Oberst. 2021. Deep learning for brake squeal: Brake noise detection, characterization and prediction. *Mechanical Systems and Signal Processing* 149 (2021), 107181. (Cited on page 62.)

[256] Toshimitsu Homma and Andrea Saltelli. 1996. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety* 52, 1 (1996), 1–17. (Cited on pages 62 and 63.)

[257] Jie Zhang, Xinbiao Xiao, Xiaozhen Sheng, Chunyan Zhang, Ruiqian Wang, and Xuesong Jin. 2016. SEA and contribution analysis for interior noise of a high speed train. *Applied Acoustics* 112 (2016), 158–170. (Cited on page 62.)

[258] Nam H Kim and Jun Dong. 2006. Shape sensitivity analysis of sequential structural–acoustic problems using FEM and BEM. *Journal of sound and vibration* 290, 1-2 (2006), 192–208. (Not cited.)

[259] F Scarpa. 2000. Parametric sensitivity analysis of coupled acoustic-structural systems. *J. Vib. Acoust.* 122, 2 (2000), 109–115. (Cited on page 62.)

[260] RI Cukier, CM Fortuin, Kurt E Shuler, AG Petschek, and J Ho Schaibly. 1973. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I Theory. *The Journal of chemical physics* 59, 8 (1973), 3873–3878. (Cited on pages 62 and 64.)

[261] Il'ya Meerovich Sobol'. 1990. On sensitivity estimation for nonlinear mathematical models. *Matematicheskoe modelirovanie* 2, 1 (1990), 112–118. (Cited on pages 62 and 63.)

[262] Fanny Sarrazin, Francesca Pianosi, and Thorsten Wagener. 2016. Global Sensitivity Analysis of environmental models: Convergence and validation. *Environmental Modelling & Software* 79 (2016), 135–152. (Cited on page 62.)

[263] Zacharie Laly, Noureddine Atalla, Sid-Ali Meslioui, and Khalid El Bikri. 2019. Sensitivity analysis of micro-perforated panel absorber models at high sound pressure levels. *Applied Acoustics* 156 (2019), 7–20. (Cited on page 62.)

[264] Hyeong Rae Lee, Ho Yong Kim, Ju Hyun Jeon, and Yeon June Kang. 2019. Application of global sensitivity analysis to statistical energy analysis: Vehicle model development and transmission path contribution. *Applied Acoustics* 146 (2019), 368–389. (Cited on page 62.)

[265] Jean-Loup Christen, Mohamed Ichchou, Bernard Troclet, Olivier Bareille, and Morvan Ouisse. 2017. Global sensitivity analysis and uncertainties in SEA models of vibroacoustic systems. *Mechanical Systems and Signal Processing* 90 (2017), 365–377. (Cited on page 62.)

[266] Greer B Humphrey, Holger R Maier, Wenyan Wu, Nick J Mount, Graeme C Dandy, Robert J Abrahart, and Christian W Dawson. 2017. Improved validation framework and R-package for artificial neural network models. *Environmental Modelling & Software* 92 (2017), 82–106. (Cited on page 62.)

[267] Saman Razavi, Anthony Jakeman, Andrea Saltelli, Clémentine Prieur, Bertrand Iooss, Emanuele Borgonovo, Elmar Plischke, Samuele Lo Piano, Takuya Iwanaga, William Becker, et al. 2021. The future of sensitivity analysis: An essential discipline for systems modeling and policy support. *Environmental Modelling & Software* 137 (2021), 104954. (Cited on pages 62 and 64.)

[268] Bruno Sudret. 2008. Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety* 93, 7 (2008), 964–979. (Cited on page 62.)

[269] Amandine Marrel, Bertrand Iooss, Beatrice Laurent, and Olivier Roustant. 2009. Calculations of sobol indices for the gaussian process metamodel. *Reliability Engineering & System Safety* 94, 3 (2009), 742–751. (Cited on page 62.)

[270] Marc Jaxa-Rozen and Jan Kwakkel. 2018. Tree-based ensemble methods for sensitivity analysis of environmental models: A performance comparison with Sobol and Morris techniques. *Environmental Modelling & Software* 107 (2018), 245–266. (Cited on pages 62 and 64.)

[271] W Chai, A Saidi, Abdelmalek Zine, C Droz, W You, and Mohamed Ichchou. 2020. Comparison of uncertainty quantification process using statistical and data mining algorithms. *Structural and Multidisciplinary Optimization* 61, 2 (2020), 587–598. (Cited on pages 62 and 65.)

[272] Ilya M Sobol. 2001. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and computers in simulation* 55, 1-3 (2001), 271–280. (Cited on page 63.)

[273] Emanuele Borgonovo and Elmar Plischke. 2016. Sensitivity analysis: a review of recent advances. *European Journal of Operational Research* 248, 3 (2016), 869–887. (Cited on page 63.)

[274] Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and Stefano Tarantola. 2010. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer physics communications* 181, 2 (2010), 259–270. (Cited on pages 63 and 64.)

[275] Andrea Saltelli, Stefano Tarantola, and KP-S Chan. 1999. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics* 41, 1 (1999), 39–56. (Cited on page 64.)

[276] Erwan Scornet. 2020. Trees, forests, and impurity-based variable importance. *arXiv:2001.04295* (2020). (Cited on pages 64 and 73.)

[277] Jon Herman and Will Usher. 2017. SALib: An open-source Python library for Sensitivity Analysis. *The Journal of Open Source Software* 2, 9 (jan 2017). https://doi.org/10.21105/joss.00097 (Cited on page 65.)

[278] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. 2018. A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*. Springer, 270–279. (Cited on pages 82 and 85.)

[279] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 1–40. (Cited on page 82.)

[280] Jeroen Van Baar, Alan Sullivan, Radu Cordorel, Devesh Jha, Diego Romeres, and Daniel Nikovski. 2019. Sim-to-real transfer learning using robustified controllers in robotic tasks involving complex dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 6001–6007. (Cited on page 83.)

[281] Jiang Hua, Liangcai Zeng, Gongfa Li, and Zhaojie Ju. 2021. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors* 21, 4 (2021), 1278. (Cited on page 83.)

[282] Gengxiang Chen, Yingguang Li, and Xu Liu. 2020. Transfer learning under conditional shift based on fuzzy residual. *IEEE Transactions on Cybernetics* 52, 2 (2020), 960–970. (Cited on page 84.)

[283] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? *Advances in neural information processing systems* 27 (2014). (Cited on pages 86 and 95.)

[284] David Duvenaud. 2014. *Automatic model construction with Gaussian processes*. Ph. D. Dissertation. University of Cambridge. (Cited on page 89.)

[285] C-L Hwang and Abu Syed Md Masud. 2012. *Multiple objective decision making—methods and applications: a state-of-the-art survey*. Vol. 164. Springer Science & Business Media. (Cited on page 100.)

[286] Su-chul Kim, Sang-gon Moon, Jong-hyeon Sohn, Young-jun Park, Chan-ho Choi, and Geun-ho Lee. 2020. Macro geometry optimization of a helical gear pair for mass, efficiency, and transmission error. *Mechanism and Machine Theory* 144 (2020), 103634. (Not cited.)

[287] Daniel Miler, Dragan Žeželj, Antonio Lončar, and Krešimir Vučković. 2018. Multi-objective spur gear pair optimization focused on volume and efficiency. *Mechanism and Machine Theory* 125 (2018), 185–195. (Not cited.)

[288] Chanho Choi, Hyoungjong Ahn, Young-jun Park, Geun-ho Lee, and Su-chul Kim. 2021. Influence of gear tooth addendum and dedendum on the helical gear optimization considering mass, efficiency, and transmission error. *Mechanism and Machine Theory* 166 (2021), 104476. (Not cited.)

[289] V Savsani, RV Rao, and DP Vakharia. 2010. Optimal weight design of a gear train using particle swarm optimization and simulated annealing algorithms. *Mechanism and machine theory* 45, 3 (2010), 531–541. (Not cited.)

[290] Sa'id Golabi, Javad Jafari Fesharaki, and Maryam Yazdipoor. 2014. Gear train optimization based on minimum volume/weight design. *Mechanism and machine theory* 73 (2014), 197–217. (Cited on page 100.)

[291] P. Garambois, J. Perret-Liaudet, and E. Rigaud. 2017. NVH robust optimization of gear macro and microgeometries using an efficient tooth contact model. *Mechanism and Machine Theory* 117 (2017), 78–95. (Cited on pages 100 and 107.)

[292] SMT - Smart Manufacturing Technology. [n. d.]. Masta. `www.smartmt.com/masta`. Accessed: 2023-06-30. (Cited on pages 104 and 107.)

[293] J Derek Smith. 2003. *Gear noise and vibration*. CRC press. (Cited on pages 104 and 106.)

[294] Yi Guo, Tugan Eritenel, Tristan M Ericson, and Robert G Parker. 2014. Vibro-acoustic propagation of gear dynamics in a gear-bearing-housing system. *Journal of Sound and Vibration* 333, 22 (2014), 5762–5785. (Cited on pages 105 and 106.)

[295] Blas Blanco, José Luís Escalona, Robert Lambert, Robert Ling, and Ole Balling. 2023. On the definition and effect of optimum gear microgeometry modifications for the gearbox of an offshore 10-MW wind turbine. *Wind Energy* (2023). (Cited on pages 106 and 108.)

[296] Gert Goch, Kang Ni, Yue Peng, and Anke Guenther. 2017. Future gear metrology based on areal measurements and improved holistic evaluations. *CIRP Annals* 66, 1 (2017), 469–474. (Cited on page 108.)

[297] ISO 1328-1:2013 2013. *Cylindrical gears — ISO system of flank tolerance classification — Part 1: Definitions and allowable values of deviations relevant to flanks of gear teeth*. Standard. International Organization for Standardization. (Cited on page 114.)

[298] J. Blank and K. Deb. 2020. pymoo: Multi-Objective Optimization in Python. *IEEE Access* 8 (2020), 89497–89509. (Cited on pages 115 and 153.)

[299] Barbara Zaparoli Cunha, Christophe Droz, Abdelmalek Zine, Mohamed Ichchou, and Stéphane Foulard. 2022. Neural Network-based Surrogates of Gear Whine Noise for Uncertainty Propagation. In *7th European Conference on Structural Control*, Vol. 1. 192–200. (Cited on page 125.)

[300] Barbara Z. Cunha, Abdel-Malek Zine, Mohamed Ichchou, Christophe Droz, and Stéphane Foulard. 2022. On machine learning-driven surrogates for sound transmission loss simulations. *The Journal of the Acoustical Society of America* 152, 4-Supplement (10 2022), A135–A135. https://doi.org/10.1121/10.0015797 (Cited on page 125.)

[301] Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, and Robert X Gao. 2019. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing* 115 (2019), 213–237. (Cited on pages 129, 130 and 131.)

[302] Fuh-Gwo Yuan, Sakib Ashraf Zargar, Qiuyi Chen, and Shaohan Wang. 2020. Machine learning for structural health monitoring: challenges and opportunities. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020*, Vol. 11379. International Society for Optics and Photonics, 1137903. (Cited on pages 129 and 138.)

[303] Charles R Farrar and Keith Worden. 2012. *Structural health monitoring: a machine learning perspective*. John Wiley & Sons. (Cited on pages 129 and 131.)

[304] Scott W Doebling, Charles R Farrar, Michael B Prime, and Daniel W Shevitz. 1996. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review. (1996). (Cited on page 130.)

[305] Ruonan Liu, Boyuan Yang, Enrico Zio, and Xuefeng Chen. 2018. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing* 108 (2018), 33–47. (Cited on pages 130, 131, 132, 140 and 141.)

[306] Samir Khan and Takehisa Yairi. 2018. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing* 107 (2018), 241–265. (Cited on pages 130, 131 and 141.)

[307] Hoon Sohn. 2007. Effects of environmental and operational variability on structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365, 1851 (2007), 539–560. (Cited on pages 130 and 132.)

[308] R Fuentes, EJ Cross, PA Gardner, LA Bull, TJ Rogers, RJ Barthorpe, H Shi, N Dervilis, CR Farrar, and K Worden. 2020. Structural Health Monitoring and Damage Identification. *Handbook of Experimental Structural Dynamics* (2020), 1–72. (Cited on pages 130, 131, 132, 133 and 134.)

[309] Aiwina Heng, Sheng Zhang, Andy CC Tan, and Joseph Mathew. 2009. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical systems and signal processing* 23, 3 (2009), 724–739. (Cited on page 131.)

[310] Anders Rytter. 1993. Vibrational based inspection of civil engineering structures. (1993). (Cited on page 131.)

[311] Arman Malekloo, Ekin Ozer, Mohammad AlHamaydeh, and Mark Girolami. 2021. Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. *Structural Health Monitoring* (2021). (Cited on pages 131 and 132.)

[312] Mohsen Azimi, Armin Dadras Eslamlou, and Gokhan Pekcan. 2020. Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. *Sensors* 20, 10 (2020), 2778. (Cited on pages 131 and 140.)

[313] Yi-zhou Lin, Zhen-hua Nie, and Hong-wei Ma. 2017. Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil and Infrastructure Engineering* 32, 12 (2017), 1025–1046. (Not cited.)

[314] Yuequan Bao and Hui Li. 2021. Machine learning paradigm for structural health monitoring. *Structural Health Monitoring* 20, 4 (2021), 1353–1372. (Cited on page 131.)

[315] Gyungmin Toh and Junhong Park. 2020. Review of vibration-based structural health monitoring using deep learning. *Applied Sciences* 10, 5 (2020), 1680. (Cited on page 131.)

[316] Majdi Flah, Itzel Nunez, Wassim Ben Chaabene, and Moncef L Nehdi. 2021. Machine learning algorithms in civil structural health monitoring: A systematic review. *Archives of Computational Methods in Engineering* 28, 4 (2021), 2621–2643. (Cited on page 131.)

[317] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, Mohammed Hussein, Moncef Gabbouj, and Daniel J Inman. 2021. A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications. *Mechanical systems and signal processing* 147 (2021). (Cited on pages 134 and 136.)

[318] Rongrong Hou and Yong Xia. 2021. Review on the new development of vibration-based damage identification for civil engineering structures: 2010–2019. *Journal of Sound and Vibration* 491 (2021). (Not cited.)

[319] X. W. Ye, T. Jin, and C. B. Yun. 2019. A review on deep learning-based structural health monitoring of civil infrastructures. *Smart Structures and Systems* 24, 5 (2019), 567–585. (Cited on page 131.)

[320] Eloi Figueiredo and James Brownjohn. 2022. Three decades of statistical pattern recognition paradigm for SHM of bridges. *Structural Health Monitoring* (2022). (Cited on page 131.)

[321] Yazhou Xie, Majid Ebad Sichani, Jamie E Padgett, and Reginald DesRoches. 2020. The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthquake Spectra* 36, 4 (2020), 1769–1801. (Cited on page 131.)

[322] Hoon Sohn, Keith Worden, and Charles R Farrar. 2002. Statistical damage classification under changing environmental and operational conditions. *Journal of intelligent material systems and structures* 13, 9 (2002), 561–574. (Cited on page 132.)

[323] Hoon Sohn, Charles R Farrar, Francois M Hemez, Devin D Shunk, Daniel W Stinemates, Brett R Nadler, and Jerry J Czarnecki. 2003. A review of structural health monitoring literature: 1996–2001. *Los Alamos National Laboratory, USA* 1 (2003). (Cited on page 132.)

[324] LAL Janssen and I Lopez Arteaga. 2020. Data processing and augmentation of acoustic array signals for fault detection with machine learning. *Journal of Sound and Vibration* 483 (2020), 115483. (Cited on pages 132 and 134.)

[325] Ozhan Gecgel, Stephen Ekwaro-Osire, João Paulo Dias, Abdul Serwadda, Fisseha M Alemayehu, and Abraham Nispel. 2019. Gearbox fault diagnostics

using deep learning with simulated data. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 1–8. (Cited on pages 135 and 140.)

[326] Chunwei Zhang, Asma A Mousavi, Sami F Masri, Gholamreza Gholipour, Kai Yan, and Xiuling Li. 2022. Vibration feature extraction using signal processing techniques for structural health monitoring: A review. *Mechanical Systems and Signal Processing* 177 (2022), 109175. (Cited on page 132.)

[327] Peiming Shi, Xiaoci Guo, Dongying Han, and Rongrong Fu. 2020. A sparse auto-encoder method based on compressed sensing and wavelet packet energy entropy for rolling bearing intelligent fault diagnosis. *Journal of Mechanical Science & Technology* 34, 4 (2020). (Cited on page 132.)

[328] Chuan Li, René-Vinicio Sanchez, Grover Zurita, Mariela Cerrada, Diego Cabrera, and Rafael E Vásquez. 2016. Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mechanical Systems and Signal Processing* 76 (2016), 283–293. (Cited on pages 132 and 135.)

[329] Jinjiang Wang, Junfei Zhuang, Lixiang Duan, and Weidong Cheng. 2016. A multi-scale convolution neural network for featureless fault diagnosis. In *2016 International Symposium on Flexible Automation (ISFA)*. IEEE, 65–70. (Cited on page 132.)

[330] David Verstraete, Andrés Ferrada, Enrique López Droguett, Viviana Meruane, and Mohammad Modarres. 2017. Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings. *Shock and Vibration* 2017 (2017). (Cited on page 132.)

[331] Sandip Kumar Singh, Sandeep Kumar, and JP Dwivedi. 2017. Compound fault prediction of rolling bearing using multimedia data. *Multimedia Tools and Applications* 76, 18 (2017), 18771–18788. (Cited on page 132.)

[332] MM Reda Taha, Aboelmagd Noureldin, JL Lucero, and TJ Baca. 2006. Wavelet transform for structural health monitoring: a compendium of uses and features. *Structural health monitoring* 5, 3 (2006), 267–295. (Cited on page 132.)

[333] Luyang Jing, Ming Zhao, Pin Li, and Xiaoqiang Xu. 2017. A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox. *Measurement* 111 (2017), 1–10. (Cited on pages 132, 135 and 140.)

[334] Weifang Sun, Bin Yao, Nianyin Zeng, Binqiang Chen, Yuchao He, Xincheng Cao, and Wangpeng He. 2017. An intelligent gear fault diagnosis methodology using a complex wavelet enhanced convolutional neural network. *Materials* 10, 7 (2017), 790. (Cited on pages 132 and 140.)

[335] Hyunseok Oh, Byung Chul Jeon, Joon Ha Jung, and Byeng D Youn. 2016. Smart diagnosis of journal bearing rotor systems: Unsupervised feature extraction scheme by deep learning. In *Annual Conference of the PHM Society*. 1–8. (Cited on pages 132 and 140.)

[336] Marcus Varanis and Robson Pederiva. 2018. Statements on wavelet packet energy–entropy signatures and filter influence in fault diagnosis of induction motor in non-stationary operations. *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 40, 2 (2018), 1–10. (Cited on page 132.)

[337] Kishore K Reddy, Soumalya Sarkar, Vivek Venugopalan, and Michael Giering. 2016. Anomaly detection and fault disambiguation in large flight data: a multi-modal deep auto-encoder approach. In *Annual Conference of the PHM Society*, Vol. 8. 1–8. (Cited on pages 132, 133, 134 and 141.)

[338] Wihan Booyse, Daniel N Wilke, and Stephan Heyns. 2020. Deep digital twins for detection, diagnostics and prognostics. *Mechanical Systems and Signal Processing* 140 (2020). (Cited on pages 132 and 136.)

[339] Wenjun Sun, Siyu Shao, Rui Zhao, Ruqiang Yan, Xingwu Zhang, and Xuefeng Chen. 2016. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement* 89 (2016), 171–178. (Cited on pages 132 and 135.)

[340] Chen Lu, Zhen-Ya Wang, Wei-Li Qin, and Jian Ma. 2017. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Processing* 130 (2017), 377–388. (Cited on pages 135 and 141.)

[341] Jie Tao, Yilun Liu, and Dalian Yang. 2016. Bearing fault diagnosis based on deep belief network and multisensor information fusion. *Shock and Vibration* 2016 (2016). (Cited on pages 135 and 141.)

[342] Linxia Liao, Wenjing Jin, and Radu Pavel. 2016. Enhanced Restricted Boltzmann Machine With Prognosability Regularization for Prognostics and Health Assessment. *IEEE Transactions on Industrial Electronics* 63, 11 (2016), 7076–7083. https://doi.org/10.1109/TIE.2016.2586442 (Cited on pages 132 and 137.)

[343] A-M Yan, Gaëtan Kerschen, Pascal De Boe, and J-C Golinval. 2005. Structural damage diagnosis under varying environmental conditions—part I: a linear analysis. *Mechanical Systems and Signal Processing* 19, 4 (2005), 847–864. (Cited on page 133.)

[344] A-M Yan, Gaëtan Kerschen, P De Boe, and J-C Golinval. 2005. Structural damage diagnosis under varying environmental conditions—part II: local PCA for non-linear cases. *Mechanical Systems and Signal Processing* 19, 4 (2005), 865–880. (Cited on page 133.)

[345] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. 2014. A review of novelty detection. *Signal processing* 99 (2014), 215–249. (Cited on page 133.)

[346] Kilian Vos, Zhongxiao Peng, Christopher Jenkins, Md Rifat Shahriar, Pietro Borghesani, and Wenyi Wang. 2022. Vibration-based anomaly detection using LSTM/SVM approaches. *Mechanical Systems and Signal Processing* 169 (2022), 108752. (Cited on page 133.)

[347] Adam Santos, Eloi Figueiredo, MFM Silva, CS Sales, and JCWA Costa. 2016. Machine learning algorithms for damage detection: Kernel-based approaches. *Journal of Sound and Vibration* 363 (2016), 584–599. (Cited on pages 133 and 140.)

[348] Antoni Lis, Ziemowit Dworakowski, and Piotr Czubak. 2021. An anomaly detection method for rotating machinery monitoring based on the most representative data. *Journal of Vibroengineering* 23, 4 (2021), 861–876. (Cited on page 133.)

[349] Eloi Figueiredo and Elizabeth Cross. 2013. Linear approaches to modeling nonlinearities in long-term monitoring of bridges. *Journal of Civil Structural Health Monitoring* 3, 3 (2013), 187–194. (Cited on page 133.)

[350] MLD Wong, LB Jack, and AK Nandi. 2006. Modified self-organising map for automated novelty detection applied to vibration signal monitoring. *Mechanical systems and signal processing* 20, 3 (2006), 593–610. (Cited on page 133.)

[351] Gabriel Michau and Olga Fink. 2021. Unsupervised transfer learning for anomaly detection: Application to complementary operating condition transfer. *Knowledge-Based Systems* 216 (2021), 106816. (Cited on pages 133 and 134.)

[352] Markos Markou and Sameer Singh. 2003. Novelty detection: a review—part 1: statistical approaches. *Signal processing* 83, 12 (2003), 2481–2497. (Cited on page 133.)

[353] Markos Markou and Sameer Singh. 2003. Novelty detection: a review—part 2:: neural network based approaches. *Signal processing* 83, 12 (2003), 2499–2521. (Cited on page 133.)

[354] N Dervilis, EJ Cross, RJ Barthorpe, and K Worden. 2014. Robust methods of inclusive outlier analysis for structural health monitoring. *Journal of Sound and Vibration* 333, 20 (2014), 5181–5195. (Cited on page 133.)

[355] Ville Lämsä and Tapani Raiko. 2010. Novelty detection by nonlinear factor analysis for structural health monitoring. In *2010 IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 468–473. (Cited on page 133.)

[356] Mehrisadat Makki Alamdari, Thierry Rakotoarivelo, and Nguyen Lu Dang Khoa. 2017. A spectral-based clustering for structural health monitoring of the Sydney Harbour Bridge. *Mechanical Systems and Signal Processing* 87 (2017), 384–400. (Cited on page 133.)

[357] Yacine Bel-Hadj and Wout Weijtjens. 2023. Anomaly detection in vibration signals for structural health monitoring of an offshore wind turbine. In *European Workshop on Structural Health Monitoring*. Springer, 348–358. (Cited on pages 134 and 141.)

[358] Irwanda Laory, Thanh N Trinh, Ian FC Smith, and James MW Brownjohn. 2014. Methodologies for predicting natural frequency variation of a suspension bridge. *Engineering Structures* 80 (2014), 211–221. (Cited on page 134.)

[359] Mohsen Mousavi and Amir H Gandomi. 2021. Prediction error of Johansen cointegration residuals for structural health monitoring. *Mechanical Systems and Signal Processing* 160 (2021), 107847. (Cited on pages 134 and 140.)

[360] Mohsen Mousavi and Amir H Gandomi. 2021. Structural health monitoring under environmental and operational variations using MCD prediction error. *Journal of Sound and Vibration* 512 (2021), 116370. (Cited on pages 134 and 140.)

[361] James Hensman, Robin Mills, SG Pierce, Keith Worden, and M Eaton. 2010. Locating acoustic emission sources in complex structures using Gaussian processes. *Mechanical Systems and Signal Processing* 24, 1 (2010), 211–223. (Cited on page 134.)

[362] SJS Hakim, H Abdul Razak, and SA Ravanfar. 2015. Fault diagnosis on beam-like structures from modal parameters using artificial neural networks. *Measurement* 76 (2015), 45–61. (Cited on pages 134 and 136.)

[363] Shao-Fei Jiang, Chun-Ming Zhang, and Shuai Zhang. 2011. Two-stage structural damage detection using fuzzy neural networks and data fusion techniques. *Expert systems with applications* 38, 1 (2011), 511–519. (Cited on page 134.)

[364] Guoqing Gui, Hong Pan, Zhibin Lin, Yonghua Li, and Zhijun Yuan. 2017. Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. *KSCE Journal of Civil Engineering* 21, 2 (2017), 523–534. (Cited on page 135.)

[365] Oliver R de Lautour and Piotr Omenzetter. 2010. Damage classification and estimation in experimental structures using time series analysis and pattern recognition. *Mechanical Systems and Signal Processing* 24, 5 (2010), 1556–1569. (Cited on page 135.)

[366] Evangelos Papatheou, Graeme Manson, Robert J Barthorpe, and Keith Worden. 2014. The use of pseudo-faults for damage location in SHM: An experimental investigation on a Piper Tomahawk aircraft wing. *Journal of Sound and Vibration* 333, 3 (2014), 971–990. (Cited on page 135.)

[367] Osama Abdeljaber, Onur Avci, Serkan Kiranyaz, Moncef Gabbouj, and Daniel J Inman. 2017. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration* 388 (2017), 154–170. (Cited on pages 135 and 140.)

[368] Zhiqiang Chen, Chuan Li, and René-Vinicio Sánchez. 2015. Multi-layer neural network with deep belief network for gearbox fault diagnosis. *Journal of Vibroengineering* 17, 5 (2015), 2379–2392. (Cited on pages 135 and 141.)

[369] D Yu, ZM Chen, KS Xiahou, MS Li, TY Ji, and QH Wu. 2018. A radically data-driven method for fault detection and diagnosis in wind turbines. *International Journal of Electrical Power & Energy Systems* 99 (2018), 577–584. (Cited on pages 135 and 141.)

[370] Pang-jo Chun, Hiroaki Yamashita, and Seiji Furukawa. 2015. Bridge damage severity quantification using multipoint acceleration measurement and artificial neural networks. *Shock and Vibration* 2015 (2015). (Cited on page 136.)

[371] Yaguo Lei, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. 2018. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical systems and signal processing* 104 (2018), 799–834. (Cited on pages 136, 137 and 140.)

[372] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou. 2011. Remaining useful life estimation–a review on the statistical data driven approaches. *European journal of operational research* 213, 1 (2011), 1–14. (Not cited.)

[373] Andrew KS Jardine, Daming Lin, and Dragan Banjevic. 2006. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing* 20, 7 (2006), 1483–1510. (Cited on page 136.)

[374] Narendhar Gugulothu, Vishnu Tv, Pankaj Malhotra, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2017. Predicting remaining useful life using time series embeddings based on recurrent neural networks. *arXiv preprint arXiv:1709.01073* (2017). (Cited on page 136.)

[375] Amgad Muneer, Shakirah Mohd Taib, Sheraz Naseer, Rao Faizan Ali, and Izzatdin Abdul Aziz. 2021. Data-Driven Deep Learning-Based Attention Mechanism for Remaining Useful Life Prediction: Case Study Application to Turbofan Engine Analysis. *Electronics* 10, 20 (2021), 2453. (Cited on pages 136 and 141.)

[376] Bingxi Zhao and Qi Yuan. 2021. A novel deep learning scheme for multi-condition remaining useful life prediction of rolling element bearings. *Journal of Manufacturing Systems* 61 (2021), 450–460. (Cited on page 136.)

[377] Andre S Yoon, Taehoon Lee, Yongsub Lim, Deokwoo Jung, Philgyun Kang, Dongwon Kim, Keuntae Park, and Yongjin Choi. 2017. Semi-supervised learning with deep generative models for asset failure prediction. *arXiv preprint arXiv:1709.00845* (2017). (Cited on pages 136, 138 and 141.)

[378] Yongmeng Zhu, Jiechang Wu, Jun Wu, and Shuyong Liu. 2022. Dimensionality reduce-based for remaining useful life prediction of machining tools with multisensor fusion. *Reliability Engineering & System Safety* 218 (2022). (Cited on pages 136 and 141.)

[379] Kai Goebel, Bhaskar Saha, Abhinav Saxena, N Mct, and N Riacs. 2008. A comparison of three data-driven techniques for prognostics. In *62nd meeting of the society for machinery failure prevention technology (mfpt)*. 119–131. (Cited on page 136.)

[380] Tarak Benkedjouh, Kamal Medjaher, Noureddine Zerhouni, and Saïd Rechak. 2013. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. *Engineering Applications of Artificial Intelligence* 26, 7 (2013), 1751–1760. (Cited on page 137.)

[381] Maor Farid. 2022. Data-driven method for real-time prediction and uncertainty quantification of fatigue failure under stochastic loading using artificial neural networks and Gaussian process regression. *International Journal of Fatigue* 155 (2022). (Cited on page 137.)

[382] Chaochao Chen, George Vachtsevanos, and Marcos E Orchard. 2012. Machine remaining useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach. *Mechanical Systems and Signal Processing* 28 (2012), 597–607. (Cited on page 137.)

[383] Merten Stender, Merten Tiedemann, David Spieler, Daniel Schoepflin, Norbert Hoffmann, and Sebastian Oberst. 2021. Deep learning for brake squeal: Brake noise detection, characterization and prediction. *Mechanical Systems and Signal Processing* 149 (2021), 107181. (Cited on pages 137 and 141.)

[384] MathWorks. 2021. Similarity-Based Remaining Useful Life Estimation (R2012b). `https://bit.ly/2Bhxpj2`. Accessed: 2021-11-23. (Cited on page 137.)

[385] Pankaj Malhotra, Vishnu Tv, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder. *arXiv preprint arXiv:1608.06154* (2016). (Cited on pages 137 and 141.)

[386] LA Bull, K Worden, and N Dervilis. 2020. Towards semi-supervised and probabilistic classification in structural health monitoring. *Mechanical Systems and Signal Processing* 140 (2020), 106653. (Cited on page 138.)

[387] LA Bull, TJ Rogers, C Wickramarachchi, EJ Cross, K Worden, and N Dervilis. 2019. Probabilistic active learning: an online framework for structural health monitoring. *Mechanical Systems and Signal Processing* 134 (2019), 106294. (Cited on page 138.)

[388] Lawrence Bull, Keith Worden, Graeme Manson, and Nikolaos Dervilis. 2018. Active learning for semi-supervised structural health monitoring. *Journal of Sound and Vibration* 437 (2018), 373–388. (Cited on page 138.)

[389] Aidan J Hughes, Lawrence A Bull, Paul Gardner, Robert James Barthorpe, Nikolaos Dervilis, and Keith Worden. 2022. On risk-based active learning for structural health monitoring. *Mechanical Systems and Signal Processing* 167 (2022), 108569. (Cited on page 138.)

[390] Achmad Widodo and Bo-Suk Yang. 2007. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing* 21, 6 (2007), 2560–2574. (Cited on page 140.)

[391] Sandeep Sony, Kyle Dunphy, Ayan Sadhu, and Miriam Capretz. 2021. A systematic review of convolutional neural network-based structural condition assessment techniques. *Engineering Structures* 226 (2021), 111347. (Cited on page 140.)

[392] Osama Abdeljaber, Onur Avci, Mustafa Serkan Kiranyaz, Boualem Boashash, Henry Sodano, and Daniel J Inman. 2018. 1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data. *Neurocomputing* 275 (2018), 1308–1317. (Cited on page 140.)

[393] Wei Zhang, Chuanhao Li, Gaoliang Peng, Yuanhang Chen, and Zhujun Zhang. 2018. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mechanical Systems and Signal Processing* 100 (2018), 439–453. (Cited on page 140.)

[394] Wei Zhang, Gaoliang Peng, Chuanhao Li, Yuanhang Chen, and Zhujun Zhang. 2017. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors* 17, 2 (2017), 425. (Not cited.)

[395] Diego Cabrera, Fernando Sancho, Chuan Li, Mariela Cerrada, René-Vinicio Sánchez, Fannia Pacheco, and José Valente de Oliveira. 2017. Automatic feature extraction of time-series applied to fault severity assessment of helical gearbox in stationary and non-stationary speed operation. *Applied Soft Computing* 58 (2017), 53–64. (Not cited.)

[396] Wenjun Sun, Rui Zhao, Ruqiang Yan, Siyu Shao, and Xuefeng Chen. 2017. Convolutional discriminative feature learning for induction motor fault diagnosis. *IEEE Transactions on Industrial Informatics* 13, 3 (2017), 1350–1359. (Cited on page 140.)

[397] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 2021. 1D convolutional neural networks and applications: A survey. *Mechanical systems and signal processing* 151 (2021), 107398. (Cited on page 140.)

[398] Turker Ince, Serkan Kiranyaz, Levent Eren, Murat Askar, and Moncef Gabbouj. 2016. Real-time motor fault detection by 1-D convolutional neural networks. *IEEE Transactions on Industrial Electronics* 63, 11 (2016), 7067–7075. (Cited on page 140.)

[399] Yang Fu, Yun Zhang, Haiyu Qiao, Dequn Li, Huamin Zhou, and Jürgen Leopold. 2015. Analysis of feature extracting ability for cutting state monitoring using deep belief networks. *Procedia Cirp* 31 (2015), 29–34. (Cited on page 141.)

[400] Yaguo Lei, Feng Jia, Jing Lin, Saibo Xing, and Steven X Ding. 2016. An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data. *IEEE Transactions on Industrial Electronics* 63, 5 (2016), 3137–3147. (Cited on page 141.)

[401] Christopher C Fuller, Stephen J Elliott, and Philip A Nelson. 1996. *Active control of vibration*. Academic press. (Cited on page 141.)

[402] Wodek Gawronski. 2004. *Advanced structural dynamics and active control of structures*. Springer Science & Business Media. (Not cited.)

[403] Rabih Alkhatib and MF Golnaraghi. 2003. Active structural vibration control: a review. *Shock and Vibration Digest* 35, 5 (2003), 367. (Cited on pages 141 and 145.)

[404] CR Fuller and AH Von Flotow. 1995. Active control of sound and vibration. *IEEE Control Systems Magazine* 15, 6 (1995), 9–19. (Cited on page 141.)

[405] Jordan Cheer. 2021. Active Sound Control in the Automotive Interior. *Future Interior Concepts* (2021), 53–69. (Cited on page 141.)

[406] Colin Hansen, Scott Snyder, Xiaojun Qiu, Laura Brooks, and Danielle Moreau. 2012. *Active Control of Noise and Vibration*. CRC Press. (Cited on pages 141 and 147.)

[407] Steven L Brunton and J Nathan Kutz. 2019. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press. (Cited on page 142.)

[408] Steve Brunton. 2020. Data-Driven Control with Machine Learning [Youtube Playlist]. shorturl.at/kmBV4. Accessed: 2022-01-06. (Cited on page 142.)

[409] Abubakar S. Umar, Muntaqa D. Alhassan, Kabiru Aminu, and Salahuddeen G. Ahmad. 2015. Modelling and Control of Dynamical Systems Using Neural Network – A Review. *International Journal of Soft Computing and Engineering (IJSCE)* 5, 4 (2015). (Cited on pages 142 and 146.)

[410] W Thomas Miller, Paul J Werbos, and Richard S Sutton. 1995. *Neural networks for control*. MIT press. (Not cited.)

[411] Kenneth J Hunt, D Sbarbaro, R Żbikowski, and Peter J Gawthrop. 1992. Neural networks for control systems—a survey. *Automatica* 28, 6 (1992), 1083–1112. (Cited on page 142.)

[412] Donald Soloway and Pamela J Haley. 1996. Neural generalized predictive control. In *Proceedings of the 1996 IEEE international symposium on intelligent control*. IEEE, 277–282. (Cited on page 142.)

[413] Kumpati S Narendra and Snehasis Mukhopadhyay. 1997. Adaptive control using neural networks and approximate models. *IEEE Transactions on neural networks* 8, 3 (1997), 475–485. (Cited on page 142.)

[414] S Narendra Kumpati, Parthasarathy Kannan, et al. 1990. Identification and control of dynamical systems using neural networks. *IEEE Transactions on neural networks* 1, 1 (1990), 4–27. (Cited on pages 142 and 146.)

[415] Lennart Ljung, Carl Andersson, Koen Tiels, and Thomas B Schön. 2020. Deep learning and system identification. *IFAC-PapersOnLine* 53, 2 (2020), 1175–1181. (Cited on pages 142 and 144.)

[416] Thomas Duriez, Steven L Brunton, and Bernd R Noack. 2017. *Machine learning control-taming nonlinear dynamics and turbulence*. Springer. (Cited on pages 142 and 147.)

[417] Johan Schoukens and Lennart Ljung. 2019. Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine* 39, 6 (2019), 28–99. (Cited on page 144.)

[418] K Worden, WE Becker, TJ Rogers, and EJ Cross. 2018. On the confidence bounds of Gaussian process NARX models and their higher-order frequency response functions. *Mechanical Systems and Signal Processing* 104 (2018), 188–223. (Cited on page 144.)

[419] Hava T Siegelmann, Bill G Horne, and C Lee Giles. 1997. Computational capabilities of recurrent NARX neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 27, 2 (1997), 208–215. (Cited on page 144.)

[420] Minas D Spiridonakos and Eleni N Chatzi. 2015. Metamodeling of dynamic nonlinear structural systems through polynomial chaos NARX models. *Computers & Structures* 157 (2015), 99–113. (Cited on page 144.)

[421] Mohsin Jamil, Muhammad Nasir Khan, Saqib Jamshed Rind, Qasim Awais, and Muhammad Uzair. 2021. Neural network predictive control of vibrations in tall structure: An experimental controlled vision. *Computers & Electrical Engineering* 89 (2021), 106940. (Cited on page 144.)

[422] Vishnu Vidya and Meher Madhu Dharmana. 2017. Model reference based intelligent control of an active suspension system for vehicles. In *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE, 1–5. (Cited on pages 144 and 146.)

[423] Jing Xu and Juntao Fei. 2010. Neural network predictive control of vehicle suspension. In *The 2nd International Conference on Information Science and Engineering*. IEEE, 1319–1322. (Not cited.)

[424] Ikbal Eski and Şahin Yıldırım. 2009. Vibration control of vehicle active suspension system using a new robust neural network control system. *Simulation Modelling Practice and Theory* 17, 5 (2009), 778–793. (Cited on page 144.)

[425] Qiang Li, Ranyang Li, Kaifan Ji, and Wei Dai. 2015. Kalman filter and its application. In *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. IEEE, 74–77. (Cited on page 144.)

[426] Giulio Reina and Arcangelo Messina. 2019. Vehicle dynamics estimation via augmented Extended Kalman Filtering. *Measurement* 133 (2019), 383–395. (Cited on page 144.)

[427] E Lourens, Edwin Reynders, Guido De Roeck, Geert Degrande, and Geert Lombaert. 2012. An augmented Kalman filter for force identification in structural dynamics. *Mechanical systems and signal processing* 27 (2012), 446–460. (Cited on page 144.)

[428] Donglin Zou, Han Zhao, Gaoyu Liu, Na Ta, and Zhushi Rao. 2019. Application of augmented Kalman filter to identify unbalance load of rotor-bearing system: Theory and experiment. *Journal of Sound and Vibration* 463 (2019), 114972. (Cited on page 144.)

[429] Geir Evensen. 2003. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics* 53, 4 (2003), 343–367. (Cited on page 144.)

[430] Vikas Namdeo and CS Manohar. 2007. Nonlinear structural dynamical system identification using adaptive particle filters. *Journal of Sound and Vibration* 306, 3-5 (2007), 524–563. (Cited on page 144.)

[431] M Khalil, A Sarkar, and S Adhikari. 2007. Data assimilation in structural dynamics: Extended-, ensemble Kalman and particle filters. In *Proceedings of the first International Conference Uncertainty in Structural Dynamics*. (Cited on page 144.)

[432] Max Schüssler. 2022. *Machine learning with nonlinear state space models*. Ph. D. Dissertation. Universität Siegen. (Cited on page 144.)

[433] Rajdip Nayek, Souvik Chakraborty, and Sriram Narasimhan. 2019. A Gaussian process latent force model for joint input-state estimation in linear structural systems. *Mechanical Systems and Signal Processing* 128 (2019), 497–530. (Cited on page 144.)

[434] Timothy J Rogers, Keith Worden, and Elizabeth J Cross. 2020. Bayesian joint input-state estimation for nonlinear systems. *Vibration* 3, 3 (2020), 281–303. (Cited on page 144.)

[435] Alessandro Chiuso and Gianluigi Pillonetto. 2019. System identification: A machine learning perspective. *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019), 281–304. (Cited on page 144.)

[436] Gianluigi Pillonetto, Francesco Dinuzzo, Tianshi Chen, Giuseppe De Nicolao, and Lennart Ljung. 2014. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica* 50, 3 (2014), 657–682. (Cited on pages 144 and 148.)

[437] Gaetan Kerschen, Keith Worden, Alexander F Vakakis, and Jean-Claude Golinval. 2006. Past, present and future of nonlinear system identification in structural dynamics. *Mechanical systems and signal processing* 20, 3 (2006), 505–592. (Cited on page 144.)

[438] Jean-Philippe Noël and Gaëtan Kerschen. 2017. Nonlinear system identification in structural dynamics: 10 more years of progress. *Mechanical Systems and Signal Processing* 83 (2017), 2–35. (Cited on page 144.)

[439] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences* 113, 15 (2016), 3932–3937. (Cited on page 144.)

[440] Marco Didonna, Merten Stender, Antonio Papangelo, Filipe Fontanela, Michele Ciavarella, and Norbert Hoffmann. 2019. Reconstruction of governing equations from vibration measurements for geometrically nonlinear systems. *Lubricants* 7, 8 (2019), 64. (Cited on page 144.)

[441] Merten Stender, Sebastian Oberst, and Norbert Hoffmann. 2019. Recovery of differential equations from impulse response time series data for model

identification and feature extraction. *Vibration* 2, 1 (2019), 25–46. (Cited on page 144.)

[442] Yaxiong Ren, Christian Adams, and Tobias Melz. 2022. Uncertainty Analysis and Experimental Validation of Identifying the Governing Equation of an Oscillator Using Sparse Regression. *Applied Sciences* 12, 2 (2022), 747. (Cited on page 144.)

[443] Thomas Simpson, Nikolaos Dervilis, and Eleni Chatzi. 2021. Machine Learning Approach to Model Order Reduction of Nonlinear Systems via Autoencoder and LSTM Networks. *Journal of Engineering Mechanics* 147, 10 (2021). https://doi.org/10.1061/(asce)em.1943-7889.0001971 (Cited on pages 145 and 146.)

[444] Randolph H Cabell and CR Fuller. 1999. A principal component algorithm for feedforward active noise and vibration control. *Journal of Sound and Vibration* 227, 1 (1999), 159–181. (Cited on page 145.)

[445] Bruce Moore. 1981. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control* 26, 1 (1981), 17–32. (Cited on page 145.)

[446] R. Cabell, D. Palumbo, and J. Vipperman. 2001. A principal component feedforward algorithm for active noise control: flight test results. *IEEE Transactions on Control Systems Technology* 9, 1 (2001), 76–83. https://doi.org/10.1109/87.896748 (Not cited.)

[447] YANG Hao. 2020. *Comprehensive Robust Analysis for Controller Tuning in Principal Component Active Control Systems*. Ph. D. Dissertation. University OF Leicester. (Not cited.)

[448] AS Al-Dmour and KS Mohammad. 2002. Active control of flexible structures using principal component analysis in the time domain. *Journal of sound and vibration* 253, 3 (2002), 545–569. (Cited on page 145.)

[449] Michael Papadopoulos and Ephrahim Garcia. 1998. Sensor placement methodologies for dynamic testing. *AIAA journal* 36, 2 (1998), 256–263. (Cited on page 145.)

[450] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. 2016. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM. (Cited on page 145.)

[451] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. 2009. Spectral analysis of nonlinear flows. *Journal of fluid mechanics* 641 (2009), 115–127. (Cited on page 145.)

[452] Akira Saito and Tomohiro Kuno. 2020. Data-driven experimental modal analysis by Dynamic Mode Decomposition. *Journal of Sound and Vibration* 481 (2020), 115434. (Cited on page 145.)

[453] Nicola Fonzi, Steven L Brunton, and Urban Fasel. 2020. Data-driven nonlinear aeroelastic models of morphing wings for control. *Proceedings of the Royal Society A* 476, 2239 (2020), 20200079. (Cited on page 145.)

[454] Gaëtan Kerschen, Maxime Peeters, Jean-Claude Golinval, and Alexander F Vakakis. 2009. Nonlinear normal modes, Part I: A useful framework for the structural dynamicist. *Mechanical systems and signal processing* 23, 1 (2009), 170–194. (Cited on page 146.)

[455] M Amabili and Cyril Touzé. 2007. Reduced-order models for nonlinear vibrations of fluid-filled circular cylindrical shells: comparison of POD and asymptotic nonlinear normal modes methods. *Journal of fluids and structures* 23, 6 (2007), 885–903. (Cited on page 146.)

[456] K Worden and PL Green. 2017. A machine learning approach to nonlinear modal analysis. *Mechanical Systems and Signal Processing* 84 (2017), 34–53. (Cited on page 146.)

[457] Nikolaos Dervilis, Thomas E Simpson, David J Wagg, and Keith Worden. 2019. Nonlinear modal analysis via non-parametric machine learning tools. *Strain* 55, 1 (2019), e12297. (Cited on page 146.)

[458] George Tsialiamanis, MD Champneys, Nikolaos Dervilis, David J Wagg, and Keith Worden. 2022. On the application of generative adversarial networks for nonlinear modal analysis. *Mechanical Systems and Signal Processing* 166 (2022), 108473. (Cited on page 146.)

[459] Yang Liu, Yan Li, Dejun Wang, and Shaoyi Zhang. 2014. Model updating of complex structures using the combination of component mode synthesis and Kriging predictor. *The Scientific World Journal* 2014 (2014). (Cited on page 146.)

[460] Hao Li, Daniel Fernex, Richard Semaan, Jianguo Tan, Marek Morzyński, and Bernd R Noack. 2021. Cluster-based network model. *Journal of Fluid Mechanics* 906 (2021). (Cited on page 146.)

[461] Thomas Daniel, Fabien Casenave, Nissrine Akkari, and David Ryckelynck. 2020. Model order reduction assisted by deep neural networks (ROM-net). *Advanced Modeling and Simulation in Engineering Sciences* 7, 1 (2020), 1–27. (Cited on page 146.)

[462] Lu Lu, Kai-Li Yin, Rodrigo C. de Lamare, Zongsheng Zheng, Yi Yu, Xiaomin Yang, and Badong Chen. 2021. A survey on active noise control in the past decade–Part II: Nonlinear systems. *Signal Processing* 181 (2021), 107929.

https://doi.org/10.1016/j.sigpro.2020.107929 (Cited on pages 146 and 147.)

[463] GCM De Abreu, Rafael Luís Teixeira, and José F Ribeiro. 2000. A neural network-based direct inverse control for active control of vibrations of mechanical systems. In *Proceedings. Vol. 1. Sixth Brazilian Symposium on Neural Networks*. IEEE, 107–112. (Cited on page 146.)

[464] William Camilo Ariza-Zambrano and Alberto Luiz Serpa. 2021. Direct inverse control for active vibration suppression using artificial neural networks. *Journal of Vibration and Control* 27, 1-2 (2021), 31–42. (Cited on page 147.)

[465] Allan C Nerves and R Krishnan. 1994. Active control of wind-induced vibrations in tall buildings using neural networks. In *Proceedings of IECON'94-20th Annual Conference of IEEE Industrial Electronics*, Vol. 2. IEEE, 1292–1297. (Cited on page 147.)

[466] Khaldoon A Bani-Hani. 2007. Vibration control of wind-induced response of tall buildings with an active tuned mass damper using neural networks. *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures* 14, 1 (2007), 83–108. (Cited on page 147.)

[467] Samuel Kyung Won Park. 2018. *Comparison of Neural Networks and Least Mean Squared Algorithms for Active Noise Canceling*. Ph. D. Dissertation. Clemson University. (Cited on page 147.)

[468] Kun Zhang, Gangming Lyu, and Xinming Luo. 2020. A Deep Recurrent Neural Network Controller For Nonlinear Active Noise Control Systems. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. IEEE, 2393–2396. (Cited on page 147.)

[469] Weifeng Liu, Puskal P. Pokharel, and Jose C. Principe. 2008. The Kernel Least-Mean-Square Algorithm. *IEEE Transactions on Signal Processing* 56, 2 (2008), 543–554. https://doi.org/10.1109/TSP.2007.907881 (Cited on page 147.)

[470] Weifeng Liu and José C Príncipe. 2008. Kernel affine projection algorithms. *EURASIP Journal on Advances in Signal Processing* 2008 (2008), 1–12. (Cited on page 147.)

[471] Hao Zhang and DeLiang Wang. 2021. Deep ANC: A deep learning approach to active noise control. *Neural Networks* 141 (2021), 1–10. https://doi.org/10.1016/j.neunet.2021.03.037 (Cited on page 147.)

[472] Corey T Wangler and Colin H Hansen. 1994. Genetic algorithm adaptation of non-linear filter structures for active sound and vibration control. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 3. IEEE, III–505. (Cited on page 147.)

[473] Cheng-Yuan Chang and Deng-Rui Chen. 2010. Active noise cancellation without secondary path identification by using an adaptive genetic algorithm. *IEEE transactions on Instrumentation and Measurement* 59, 9 (2010), 2315–2327. (Cited on page 147.)

[474] Muhammad Asif Zahoor Raja, Muhammad Saeed Aslam, Naveed Ishtiaq Chaudhary, and Wasim Ullah Khan. 2018. Bio-inspired heuristics hybrid with interior-point method for active noise control systems without identification of secondary path. *Frontiers of Information Technology & Electronic Engineering* 19, 2 (2018), 246–259. (Not cited.)

[475] Wasim Ullah Khan, ZhongFu Ye, Naveed Ishtiaq Chaudhary, and Muhammad Asif Zahoor Raja. 2018. Backtracking search integrated with sequential quadratic programming for nonlinear active noise control systems. *Applied Soft Computing* 73 (2018), 666–683. (Not cited.)

[476] Muhammad Asif Zahoor Raja, Muhammad Saeed Aslam, Naveed Ishtiaq Chaudhary, Muhammad Nawaz, and Syed Muslim Shah. 2019. Design of hybrid nature-inspired heuristics with application to active noise control systems. *Neural Computing and Applications* 31, 7 (2019), 2563–2591. (Not cited.)

[477] Nirmal Kumar Rout, Debi Prasad Das, and Ganapati Panda. 2016. Particle swarm optimization based nonlinear active noise control under saturation nonlinearity. *Applied Soft Computing* 41 (2016), 275–289. (Not cited.)

[478] Nithin V George and Ganapati Panda. 2012. A particle-swarm-optimization-based decentralized nonlinear active noise control system. *IEEE Transactions on Instrumentation and Measurement* 61, 12 (2012), 3378–3386. (Not cited.)

[479] Nirmal Kumar Rout, Debi Prasad Das, and Ganapati Panda. 2019. PSO based adaptive narrowband ANC algorithm without the use of synchronization signal and secondary path estimate. *Mechanical Systems and Signal Processing* 114 (2019), 378–398. (Cited on page 147.)

[480] MS Saad. 2014. Evolutionary optimization and real-time self-tuning active vibration control of a flexible beam system. *Ph. D. thesis, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia* (2014). (Cited on page 147.)

[481] Hadi Nobahari, Seyed A Hosseini Kordkheili, and Sajad Saraygord Afshari. 2014. Hardware-in-the-loop optimization of an active vibration controller in a flexible beam structure using evolutionary algorithms. *Journal of Intelligent Material Systems and Structures* 25, 10 (2014), 1211–1223. (Not cited.)

[482] Asan G. A. Muthalif, Khairul A. M. Nor, Azni Nabela Wahid, and Abdelrahman Ali. 2021. Optimization of Piezoelectric Sensor-Actuator for Plate Vibration Control Using Evolutionary Computation: Modeling, Simulation

and Experimentation. *IEEE Access* 9 (2021), 100725–100734. https://doi.org/10.1109/ACCESS.2021.3096972 (Not cited.)

[483] Medhat HA Awadalla. 2018. Spiking neural network and bull genetic algorithm for active vibration control. *International Journal of Intelligent Systems and Applications* 10, 2 (2018), 17. (Not cited.)

[484] Javad Katebi, Mona Shoaei-parchin, Mahdi Shariati, Nguyen Thoi Trung, and Majid Khorami. 2020. Developed comparative analysis of metaheuristic optimization algorithms for optimal active control of structures. *Engineering with Computers* 36, 4 (2020), 1539–1558. (Cited on page 147.)

[485] Yang-Yin Lin, Jyh-Yeong Chang, and Chin-Teng Lin. 2013. A TSK-type-based self-evolving compensatory interval type-2 fuzzy neural network (TSCIT2FNN) and its applications. *IEEE Transactions on Industrial Electronics* 61, 1 (2013), 447–459. (Cited on page 147.)

[486] Qi-Zhi Zhang, Woon-Seng Gan, and Ya-li Zhou. 2006. Adaptive recurrent fuzzy neural networks for active noise control. *Journal of Sound and Vibration* 296, 4-5 (2006), 935–948. (Not cited.)

[487] Qi-Zhi Zhang and Woon-Seng Gan. 2004. Active noise control using a simplified fuzzy neural network. *Journal of sound and vibration* 1, 272 (2004), 437–449. (Not cited.)

[488] Navid Azadi and Abdolreza Ohadi. 2012. Filtered gradient active fuzzy neural network noise control in an enclosure backed by a clamped plate. *International Journal of Adaptive Control and Signal Processing* 26, 5 (2012), 451–468. (Cited on page 147.)

[489] Sy Dzung Nguyen, Quoc Hung Nguyen, and Seung-Bok Choi. 2015. Hybrid clustering based fuzzy structure for vibration control–Part 1: A novel algorithm for building neuro-fuzzy system. *Mechanical Systems and Signal Processing* 50 (2015), 510–525. (Cited on page 147.)

[490] Devdutt Singh. 2018. Passenger body vibration control in active quarter car model using ANFIS based super twisting sliding mode controller. *Simulation Modelling Practice and Theory* 89 (2018), 100–118. (Cited on page 147.)

[491] Lucian Buşoniu, Tim de Bruin, Domagoj Tolić, Jens Kober, and Ivana Palunko. 2018. Reinforcement learning for control: Performance, stability, and deep approximators. *Annual Reviews in Control* 46 (2018), 8–28. https://doi.org/10.1016/j.arcontrol.2018.09.005 (Cited on page 148.)

[492] Frank L Lewis, Draguna Vrabie, and Kyriakos G Vamvoudakis. 2012. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine* 32, 6 (2012), 76–105. (Cited on page 148.)

[493] Jens Kober, J Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274. (Cited on page 148.)

[494] Kourosh Latifi, Artur Kopitca, and Quan Zhou. 2020. Model-free control for dynamic-field acoustic manipulation using reinforcement learning. *IEEE Access* 8 (2020), 20597–20606. (Cited on page 148.)

[495] Behrooz Raeisy and Shapoor Golbahar Haghighi. 2012. Active noise controller with reinforcement learning. In *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*. IEEE, 074–079. (Cited on page 148.)

[496] Zhi-cheng Qiu, Guo-hao Chen, and Xian-min Zhang. 2021. Reinforcement learning vibration control for a flexible hinged plate. *Aerospace Science and Technology* 118 (2021), 107056. (Cited on page 148.)

[497] Zhang Tao, Deng Yian, Hu Fan, Zhang Xiangqi, Wei Yaoyao, Liu Tianlin, Paul Bolchover, Yu Tao, Zheng Guishui, Chen Rongbing, et al. 2020. Reducing Vibration of A Rotating Machine with Deep Reinforcement Learning. In *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 932–937. (Cited on page 148.)

[498] Ralf Gulde, Marc Tuscher, Akos Csiszar, Oliver Riedel, and Alexander Verl. 2019. Reinforcement Learning Approach to Vibration Compensation for Dynamic Feed Drive Systems. In *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*. IEEE, 26–29. (Cited on page 148.)

[499] Soheila Sadeghi Eshkevari, Soheil Sadeghi Eshkevari, Debarshi Sen, and Shamim N Pakzad. 2021. RL-Controller: a reinforcement learning framework for active structural control. *arXiv preprint arXiv:2103.07616* (2021). (Cited on page 148.)

[500] Hejia Gao, Wei He, Youmin Zhang, and Changyin Sun. 2020. Vibration Control based on Reinforcement Learning for a Flexible Building-like Structure System with Active Mass Damper against Disturbance Effects. In *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2380–2385. (Cited on page 148.)

[501] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and evolutionary computation* 1, 1 (2011), 32–49. (Cited on pages 151 and 152.)

[502] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197. (Cited on page 152.)

[503] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. 2009. *Robust optimization*. Vol. 28. Princeton university press. (Cited on page 154.)

dernière page de la thèse

AUTORISATION DE SOUTENANCE

Vu les dispositions de l'arrêté du 25 mai 2016 modifié par l'arrêté du 26 août 2022,

Vu la demande des directeurs de thèse

Messieurs M. ICHCHOU et A. ZINE

et les rapports de

M. S. COGAN
Chargé de Recherche HDR - FEMTO-ST, Département Mécanique appliquée,
24 chemin de l'Epitaphe, 25000 BESANSON

et de

M. S. DE ROSA
Professeur - University of Naples Federico II, Department of Industrial Engineering,
Via Claudio 21, 80125 Napoli, Italia

Madame ZAPAROLI CUNHA Barbara

est autorisée à soutenir une thèse pour l'obtention du grade de **DOCTEUR**

Ecole doctorale Mécanique, Energétique, Génie Civil, Acoustique

Fait à Ecully, le 27 novembre 2023

Pour le directeur de l'École Centrale de Lyon
Le directeur de la recherche

Christophe CORRE