Data-Driven Modeling of Tracked Order Vibration in Turbofan Engine

Manu Krishnan

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Pablo A. Tarazaga, Co-chair
Serkan Gugercin, Co-chair
Mayuresh Patil
Michael Philen

December 15, 2021
Blacksburg, Virginia

Keywords: Turbofan engine vibration, order analysis, DMD, data-driven, ROM

Data-Driven Modeling of Tracked Order Vibration in Turbofan Engine

Manu Krishnan

(ABSTRACT)

Aircraft engines are one of the most heavily instrumented parts of an aircraft, and the data from various types of instrumentation across these engines are continuously monitored both offline and online for potential anomalies. Vibration monitoring in aircraft engines is traditionally performed using an order tracking methodology. Currently, there are no representative and efficient physics-based models with the adequate fidelity to perform vibration predictions in aircraft engines, given various parametric dependencies existing among different attributes such as temperature, pressure, and external conditions. This gap in research is primarily attributed to the limited understanding of mutual interactions of different variables and the nonlinear nature of engine vibrations. The objective of the current study is three-fold: (i) to present a preliminary investigation of tracked order vibrations in aircraft engines and statistically analyze them in the context of their operating environment, (ii) to develop data-driven modeling methodology to approximate a dynamical system from input-output data, and (iii) to leverage these data-driven modeling methodologies to develop highly accurate models for tracked order vibration in a turbo-fan engine valid over a wide range of operating conditions.

Off-the-shelf data-driven modeling techniques, such as machine learning methods (eg., regression, neural networks), have several drawbacks including lack of interpretability and limited scope, when applying them to a complex multiscale multi-physical dynamical system. Moreover, for dynamical systems with external forcing, the identified model should not only be suitable for a specific forcing function, but should also generally approximate the input-output behavior of the data source. The author proposes a novel methodology known as Wavelet-based Dynamic Mode Decomposition (WDMD). The methodology entails using wavelets in conjunction with input-output dynamic mode decomposition (ioDMD). Similar to time-delay embedded DMD (Delay-DMD), WDMD builds on the ioDMD framework without the restrictive assumption of full state measurements. The author demonstrates the present methodology's applicability by modeling the input-output response of an Euler-Bernoulli finite element beam model, followed by an experimental investigation.

As a first step towards modeling the tracked order vibration amplitudes of turbofan engines, the interdependencies and cross-correlation structure between various thermo-mechanical variables and tracked order vibration are analyzed. The order amplitudes are further contextualized in terms of their operating regime, and exploratory data analyses are performed to quantify the variability within each operating condition (OC). The understanding of complex correlation structures is leveraged and subsequently utilized to model tracked order

vibrations. Switching linear dynamical system (SLDS) models are developed using individual data-driven models constructed using WDMD, and its performance in approximating the dynamics of the 1^{st} order amplitudes are compared with the state-of-the-art time-delay embedded dynamic mode decomposition (Delay-DMD) and Lasso regression.

A parametric approach is proposed to improve the model further by leveraging previously developed WDMD and Delay-DMD methods and a parametric interpolation scheme. In particular, a recently developed pole-residue interpolation scheme is adopted to interpolate between several linear, data-driven reduced-order models (ROMs), constructed using WDMD and Delay-DMD surrogates, at known parameter samples. The parametric modeling approach is demonstrated by modeling the transverse vibration of an axially loaded finite element (FE) beam, where the axial loading is the parameter. Finally, a parametric modeling strategy for tracked order amplitudes is presented by constructing locally valid ROMs at different parametric samples corresponding to each pass-off test. The performance of the parametric-ROM is quantified and compared with the previous frameworks.

This work was supported by the Rolls-Royce Fellowship, sponsored by the College of Engineering, Virginia Tech.

Data-Driven Modeling of Tracked Order Vibration in Turbofan Engine

Manu Krishnan

(GENERAL AUDIENCE ABSTRACT)

Vibrations in commercial aircraft engines are of utmost importance as they directly translate to aviation health and safety, and hence are continuously monitored both online and offline for potential abnormalities. Notably, this is of increased interest with the abundance of air transportation in today's world. However, there is limited understanding of the complex higher vibration in aircraft engines. Vibration engineers often face ambiguity when interpreting higher vibrations. This can often lead to a lengthy investigative process resulting in longer downtime and increased testbed occupancy, ultimately leading to revenue loss. It is often hypothesized that prior engine running conditions such as shutdown/cooling time between one engine run to another engine run affect the vibration profile. Nonetheless, there exists a gap in understanding tying together various historical operational conditions, temperature, pressures, and current operational conditions with the expected vibration in the engine.

This study aims to fill some of these gaps in our understanding by proposing a data-driven strategy to model the vibrations in commercial aircraft engines. Subsequently, this data-driven model can serve as a baseline model to compare the observed vibrations with the model predicted vibration and supplement physics-based models. The data for the present study is generated by operating a commercial turbofan engine in a testbed. With the advent of machine learning and data fusion, various data-driven techniques exist to model dynamical systems. However, the complexity of the turbofan engine vibrations calls for developing new techniques applicable towards modeling the vibration characteristics of a turbofan engine. Specifically, this dissertation details the development of a novel methodology called Wavelet-based Dynamic Mode Decomposition (WDMD) and applies the technique to model input-output characteristics of various dynamical systems ranging from a numerical finite element (FE) beam to an experimental free-free beam to shaft vibrations in a turbofan engine. The study finally presents an improved modeling framework by incorporating the existing techniques with parametric dependencies. This enables the existing method to consider slight differences existing from one engine run to another, such as the history of the engine, the shutdown time, and the outside environmental parameters.

Dedication

Dedicated to my mom Suja, my dad P.K Nair and my brother Anoop

Acknowledgments

First and foremost, I would like to express my deepest gratitude and sincere thanks to my Ph.D. supervisor and role model, Dr. Pablo Tarazaga, for his inspirational guidance and support. I am so fortunate to work with such an understanding and supportive advisor with whom I can share my problems and lookup for advice, be it academics or life. In addition to shaping my goals and transforming me as a researcher, you taught me the spirit of cheerfulness, the art of communication, and how to be an extraordinary leader. I am so grateful to you and Vanessa for all the holidays and providing a home away from home. From the bottom of my heart: Thank you, Dr. T.

Secondly, I would like to thank my co-supervisor, Dr. Serkan Gugercin, for his steadfast support, encouragement, and valuable discussions during various stages of this work. You have been extremely patient with me, especially while reviewing my manuscripts. I really appreciate all your help. My sincere thanks to Dr. Mayuresh Patil and Dr. Michael Philen for serving on my thesis committee. Your guidance and suggestions have helped me shape this work. Special thanks to Dr. Ibrahim Sever for his vision and assistance in securing funding, without whom this work would have never crystallized. I would also like to recognize the effort of Glyn Fox, Rolls-Royce for getting me up to speed with gas turbines and valuable insights during various stages of this project. I would like to thank the Rolls-Royce fellowship and John R. Jones III fellowship for financially supporting me.

I acknowledge the support of my labmates, namely: Dr. Sriram, Dr. Mohammad, Tim, Dr. Ellis, Dr. Sheyda, Dr. Rodrigo, Mostafa, Lucas, Sourabh, Dr. Jeremy, and Murat. I would like to take this moment to especially thank my friend and mentor, Dr. Sriram Malladi, for all the countless corridor discussions on traveling waves and for teaching me the fundamentals of experimental modal analysis and vibration testing.

I would like to thank my friends in Blacksburg, Dr. Vibin, Jopaul, Shravan, Ajit, Swetha, Rutuja, Vaishakhi, Dr. Bijo, Sachin, Sathya, Dr. Sheril, Dr. Onima, Dr. Ayush, Sasha, and Akshat, for providing such a lovely positive environment. Special thanks to Dr. Vibin Abraham, Jopaul Mathew, and Vaishakhi Suresh for all the delicious homemade meals and for taking care of me over the last few years. I would like to thank my girlfriend, Vaishakhi Suresh, for her unwavering emotional support and all the positivity she has brought to my life. Special thanks are reserved to my friend Jithin Thampi for his support and friendship.

Finally, I would like to take this moment to thank my family members back in India for all the support and love. My mom, Suja S. S, has immense willpower to challenge all the adversities in life, and I would like to take this opportunity to thank her for inspiring me to do good and for her steadfast support. I would like to thank my dad P.K Nair and my brother Anoop for being supportive and enabling me to pursue my dreams.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Order tracking has traditionally been the most popular tool for analyzing vibrational or acoustical signals originating from rotating machines [1, 2]. Conventional methodologies like Operational Modal Analysis (OMA) [3] and Experimental Modal Analysis (EMA) [4] are not suitable for analysis of rotating machines as the residual imbalance and misalignment creates excitation containing harmonics with frequency multiples of its rotating speed, thereby affecting the OMA and EMA procedure [5]. This led to the development of order tracking methodology to determine the critical speeds where structural resonances are excited by the rotation of the machine. The analysis is often carried out by accelerating (run-up) and decelerating (coast-down) the machine under test while recording the vibrational or acoustical signals. The Order-based Modal Analysis (OBMA) method combines Order Tracking (OT) techniques and the OMA method. First, the OT method extracts the amplitude and phase of forced vibration response under different speeds. After being processed, the response curves of different engine orders can be taken as the pseudo-FRF curve for resonance identification. Although OT is fundamentally used as a system identification method, it has also found application in vibration monitoring with the pseudo-FRF (characteristic curve) as a baseline to identify potential anomalies [6].

Vibration monitoring using OT methodology has an inherent disadvantage, as it focuses only on shaft orders. The OT method assumes that the shaft imbalances excite the engine structure. While this condition holds true in almost all cases, the input excitation forces may not be constant from one engine run to another due to a range of issues, thus leading to deviation of the pseudo-FRF (hereinafter called as characteristic curve) and impeding its use for monitoring purposes. This deviation in the resulting pseudo-FRF amplitude can be attributed to one of the following: (i) variations in the loading due to changes in the rotor imbalance, (ii) structural or material changes due to the difference in thermal state, and (iii) changes in the boundary conditions due to moving parts of the engine.

Traditionally, it has been known among industrial practitioners that a range of other factors such as operational conditions and historical parameters shape engine vibration [7]. However, there exists a void in the literature regarding a model for tracked order vibrations unifying the concepts of operating conditions and historical parameters. This is partly due to the complex nature of the multi-physics problem and the associated challenges of incorporating the thermo-mechanical effects. Developing accurate physics-based models of the complex multi-physics problem is not always feasible or straightforward. In such situations, are data-

driven models a viable alternative? How robust are data-driven models in representing complex multi-physics interactions, as in the case of modeling the tracked order vibration of turbofan engine? In a recent effort by the author [8, 9, 10], novel data-driven strategies using Lasso regression [10] and Dynamic mode decomposition (DMD) [8, 9] have been developed to build models to predict the tracked order vibration response.

Data-driven models rely entirely on the measured input-output responses [11, 12]. Hence, their success depends on a deeper understanding of the cross-correlation and interaction between the target variable (tracked order vibration) and predictor variables (measured temperature, pressure, and other parameters from the engine). However, engine vibration is a complex problem depending on several unknowns and various parameters interacting over an extended period, which poses a challenge in building a data-driven model [8, 13]. Detecting cross-correlations between signals is the most common way to improve the understanding of a complex system [14]. The multi-scale detrended cross-correlation analysis (MS-DCCA) technique is used to analyze interdependencies among various parameters and the tracked order vibration [13], thereby paving the way for its use as a tool for feature selection in the present study. Once the right features are identified, data-driven modeling methodologies can be employed for model building. This is a heavily studied topic with varying approaches based on the eventual goal. At the forefront are supervised machine learning methods such as regression [15], artificial neural networks [16], and LSTM [17], which minimize an appropriate cost function [16]. However, neural networks lack the required capabilities to capture the interdependencies and long-scale correlations, as they work on current time series without taking the effect of the previous time instances' parameters on the current or future time instants [18]. This led to the development of RNNs and LSTMs, which retain information from the previous time instants. Nonetheless, these complicated designs pose their own challenges, especially regarding the vanishing gradients and the heavy computations required in the training process.

In recent years the operator theoretic framework based on Koopman operator [19] has garnered interest in the dynamical system community [20, 21, 22]. One such method is the Dynamical mode decomposition (DMD) [20], which became a popular tool among researchers for data-driven identification, owing to its ability to decompose high dimensional fluid flow data into its coherent spatio-temporal structures [22, 23]. However, DMD require full-state information, thereby limiting the application of DMD to systems with high-dimensional data, such as fluid flow. For low dimensional systems and for systems with limited measurement, variants of DMD known as time-delay DMD [24, 25, 26] have found successful application.

This dissertation aims to model the tracked order vibration in a commercial turbofan engine using various data-driven strategies. Various linear operator theoretic framework method such as DMD related methods are utilized to approximate the input-output dynamics of the tracked order vibration. Additionally, owing to the inherent low dimensional nature of the turbofan engine system, new methodology such as wavelet-based DMD is developed. The efficacy of this methodology is tested by applying it to model various classes of dynamical system such as a cantilever beam undergoing vibration, and a free-free beam excited with

MFC patches. Finally, to improve the quality of the fit in various operating conditions, a new parametric approach using recently developed pole-residue interpolation framework is proposed.

The detailed structure and organization of the dissertation is explained in Section 1.1. This is followed by the list of attributions in Section 1.2 that describes the contributions of all investigators of the project. In Section 1.3, the list of major achievements of the present work are outlined. and the author's prior work, Wavelet-based DMD [21]

1.1 Structure and Contents

This is a manuscript style dissertation, comprising of six chapters that are composed of manuscripts that are either under revision or ready to be submitted to various peer-reviewed journals in the pertinent area of turbofan engine vibrations and data-driven modeling.

Chapter 1 describes the motivation and outlines the content and structure of the dissertation.

Chapter 2 provides an overview of the literature in the data-driven modeling framework, specifically applicable to modeling a dynamical system.

Chapter 3 is the first manuscript titled "Data-driven modeling and analysis of tracked order vibration in turbofan engines under different operating conditions", which will be submitted to the American Institute of Aeronautics and Astronautics (AIAA) Journal. First, the paper provides a brief background about the order analysis and necessitates the need for a data-driven modeling strategy tying together various operating conditions. Second, the paper uses the dataset collected during the pass-off tests in a commercial turbofan engine and briefly summarizes the results of analysing the tracked order vibration with respect to various operational conditions. Finally, the paper presents a data-driven modeling strategy using switching linear dynamical system (SLDS) in conjunction with Lasso regression and time-delay embedded DMD (Delay-DMD). Although, preliminary results of applying off-the shelf models like Delay-DMD and Lasso Regression is satisfactory, the performance of the data-driven algorithm can still be improved. This paved to the development of wavelet-based dynamic mode decomposition (WDMD) which is the subject matter of Chapter 4.

Chapter 4 is the second manuscript titled "A wavelet-based dynamic mode decomposition for modeling mechanical systems from partial observations", which has been submitted and currently under review by the Mechanical Systems and Signal Processing (MSSP) Journal. The WDMD methodology builds on input-output DMD (ioDMD) and utilizes the maximal overlap discrete wavelet transform (MODWT) coefficients of the measured responses as observables, thereby enlarging the state dimensions. In other words, the MODWT coefficients of the measured output serve as auxiliary state vectors. The efficacy of the WDMD methodology in approximating a linear dynamical system from input-output data is explored

by applying it on a simple numerical case and an experimental case study.The paper begins with a mathematical overview of the dynamic mode decomposition technique and proceeds to rigorously explain our methodology, wavelet-based dynamic mode decomposition (WDMD). A numerical case study of modeling the input-output response of a finite element (FE) beam is demonstrated using the WDMD methodology. The paper concludes by demonstrating the efficacy of WDMD in modeling practical dynamical systems by taking the example of an experimental free-free beam excited by MFC patches.

Chapter 5 is the third manuscript titled "Wavelet-based Dynamic mode decomposition for tracked order vibration modeling in turbofan engine", which will also be submitted to the American Institute of Aeronautics and Astronautics (AIAA) Journal. The paper discusses applying the recently developed WDMD methodology to model the 1^{st} order amplitudes in a turbofan engine. The paper presents a numerical example of modeling the order amplitudes an FE beam excited by an unbalanced rotor. The paper also illustrates the efficacy of WDMD methodology under various noise scenarios. The paper concludes by demonstrating the tracked order vibration data-driven models obtained using WDMD. The efficacy of WDMD data-driven model is compared with the Delay-DMD and studied in rigorous detail.

Chapter 6 is the fourth and final manuscript titled "Data-driven modeling of tracked order vibrations in turbofan engine using pole-residue interpolation of reduced order surrogates", which will be submitted to American Institute of Aeronautics and Astronautics (AIAA) Journal after further analysis. The paper discusses the recently developed pole-residue interpolation scheme and proposes applying the same to model the 1^{st} order amplitudes in a turbofan engine. First the paper provides a brief overview of the pole-residue interpolation scheme. Second, the paper presents a numerical case study of an axially loaded FE beam undergoing transverse vibration, where the axial load is the parameter. Finally, the efficacy of the proposed pole-residue interpolation scheme in modeling the tracked order response is summarized by comparing it with our previously developed techniques.

Chapter 7 discusses the conclusions of the present study and identifies avenues for further extension and research.

In addition to the chapters, appendix A discusses the various features of the toolbox titled "Contextual characteristic curve and parametric extraction for order amplitudes" (C3PEO). All the journal manuscripts in Chapter 3, 4, 5 and 6 are reformatted to abide with the conventional dissertation format.

1.2 Attributions

The work presented here in this dissertation has largely benefited from collaboration with several mentors mentioned below:

Dr. Pablo Tarazaga is the principal investigator and the chair of the advisory committee for the work presented here. In addition to conceptualization, funding acquisition, and project management, he has closely supervised and reviewed the findings and work presented here.

Dr. Serkan Gugercin is the co-chair of the advisory committee for the work presented here. Dr. Gugercin played a large role in assisting the principal investigator with methodology development, project management and reviewing the work presented here.

Dr. Ibrahim A. Sever is the Rolls-Royce principal investigator for the present work and assisted Dr. Tarazaga in both conceptualization and project management. He has provided valuable assistance and closely supervised the work presented here.

1.3 Achievements

The key findings and achievements of this work are summarized as follows:

- Developed a new data-driven modeling technique called wavelet-based dynamic mode decomposition (WDMD) for modeling mechanical systems from partial observations.
- WDMD methodology achieved the same accuracy as input-output DMD with measurements as few as 5%.
- The performance of the DMD algorithms are affected by noise and WDMD had a relatively lower detoriation as compared with time-delay-embedded DMD (Delay-DMD).
- Analysis of the order amplitudes in turbofan engine demonstrated higher variability in specific operating regimes of the pass-off test, especially near the resonance RPM zone.
- Most of the sensor data are collinear and multi-scale detrended partial cross-correlation analysis (MS-DPCCA) was utilized to evaluate the interdependencies and cross-correlation structure between various thermo-mechanical variables and tracked order vibration across multiple time scales. Variables such as oil pressure demonstrated higher degree of correlation to order amplitudes corresponding to high pressure shaft RPM.
- Leveraged the understanding of complex correlation structures and subsequently utilized them to model data-driven model for tracked order vibrations.
- Developed the idea of recreating the engine dynamics over the whole pass-off tests using locally valid models constructed using various data-driven techniques such as WDMD, Delay-DMD and Lasso regression.

- The quality of the fit using WDMD locally valid models were superior as compared to the ones using Delay-DMD and lasso regression.
- It is hypothesized that parametric dependencies exist in the form of historical operations affecting engine vibrations. Developed a parametric framework using the recently developed pole-residue interpolation apporach and defined new empirical parameters.
- Applied the parametric framework to model the tracked order vibration, which resulted in better predictive models in the higher vibration operating conditions, as compared with the generalized framework.
- Developed a toolbox, known as C3PEO, that ties together all the methods and findings presented earlier. This toolbox can be used by the industry for analysis and as a predictive tool.

Chapter 2

Background and literature review

This chapter serves as a primer of the work that follows, complementing the literature review in each of the manuscripts. A fundamental description of the order tracking analysis is presented followed by the brief motivation of the present work. A detailed review of the data-driven modeling techniques is further carried out in the remainder of the chapter.

2.1 Vibration monitoring in turbofan engine

This section provides a brief overview of the vibration monitoring in turbofan engines. In particular, the following section details the traditional order analysis principle and presents an example data set from a commercial turbofan engine undergoing several pass-off tests within a span of few days.

2.1.1 Order analysis

In the present section, we provide a brief background about tracked order vibration. Consider measured acceleration response from an aircraft engine operating at a constant speed. Fourier transform on this 'steady-state' vibration signal indicates the frequency content of the vibrations at that particular speed. If the same procedure is repeated for a prescribed window length while the engine is undergoing a slow coast-up from idle to maximum power, we can construct a simple waterfall plot of all frequency spectra of the analyzed time samples. Since the rotational speed is varying with the time ($(\omega = \omega(t))$, the frequency components are also time-varying and are called orders. The orders can be defined as a time-varying phasor with a phase value related to the rotational frequency of the reference shaft. It generates a sinusoidal function with varying frequencies. In a turbofan engine with concentrically rotating shafts, the frequency content of interest often lies in the fundamental shaft harmonics (order = 1). Mathematically, we can define a p^{th} order as,

$$\psi(t) = Z_p(t)e^{i2\pi\frac{p}{T}t}, \tag{2.1}$$

where $\psi(t)$ is the order component; $Z_p(t)$ is a complex number representing the amplitude and phase of the p^{th} order, and T is the time period of the primary order (in sec). The

value of $Z_p(t)$ is not a constant, and it usually depends on the rotational speed and other structural conditions like temperature, initial conditions, etc. For a turbofan engine, the original measured vibration signal $y(t)$ is comprised of several different orders as,

$$y(t) = \sum_{p=0}^{p=\infty} Z_p(t) e^{i2\pi \frac{p}{T} t}. \tag{2.2}$$

When viewed alongside the RPM profile, the order amplitude time series reflects important information such as an imbalance in the rotor shafts, bearing failure, misalignment, and resonance amplification. Hence, the order amplitudes are of interest, and there exists a variety of methods in the literature, called order tracking methods, to extract these order amplitudes given the raw vibration signal $y(t)$ and the RPM profile. The order n is set in advance and, in the case of aircraft engines, are integral multiples of the rotational frequency of the reference shaft. In the present study, we focus our attention on the fundamental shaft orders (i.e., order $(n) = 1$). Traditionally, order tracking techniques can be classified into 4 categories: (i) Time-domain sampling-based Fast Fourier Transform order tracking, (ii) Resampling-based order tracking, (iii) Time Variant Discrete Fourier Transform, and (iv) Vold-Kalman filter-based order tracking (VK). Interested readers are directed to [27, 28] for a rigorous explanation of each technique.

2.1.2 Tracked order analysis

Towards understanding the phenomenon of tracked order vibrations and its variability within equivalent loading conditions, it is pertinent to understand the working of a typical turbofan engine. Commercial turbofan engine consist of a compressor section at the front of the engine for drawing in the air and subsequently pressurizing it to $40 - 50$ times the atmospheric pressure. This air is further delivered into a combustion chamber where it gets mixed with the fuel. The hot air, as a result of combustion, is expelled from the rear of the engine, where the rapidly-expanding heated gas provides the kinetic force. Turbines, mounted at the back of the engine, extract energy from these exhaust gases to drive the compressor section. The compressor section and the turbine section are connected via a shaft. Many modern gas-turbine engines involve multiple compression stages such as low pressure (LP), intermediate pressure (IP), and high pressure (HP) stages. Each of these stages are connected to the corresponding turbine via concentric shafts which rotate independently. [29].

The motivation for the present work can be summarized in the Figure 2.1. The figure shows Intermediate Pressure(IP) shaft speed recorded by a transducer during a traditional engine pass off test. The test series consists of a set of 5 engine pass off tests, conducted during a span of few days and the tracked order vibration magnitude plotted alongside the IP shaft speed. The tracked order vibration plot displays variability, largely in amplitude, among different runs albiet the structure being excited under "equivalent" forcing conditions at each run as shown in fig. 2.1. It can be discerned from the figure that, there exists significant

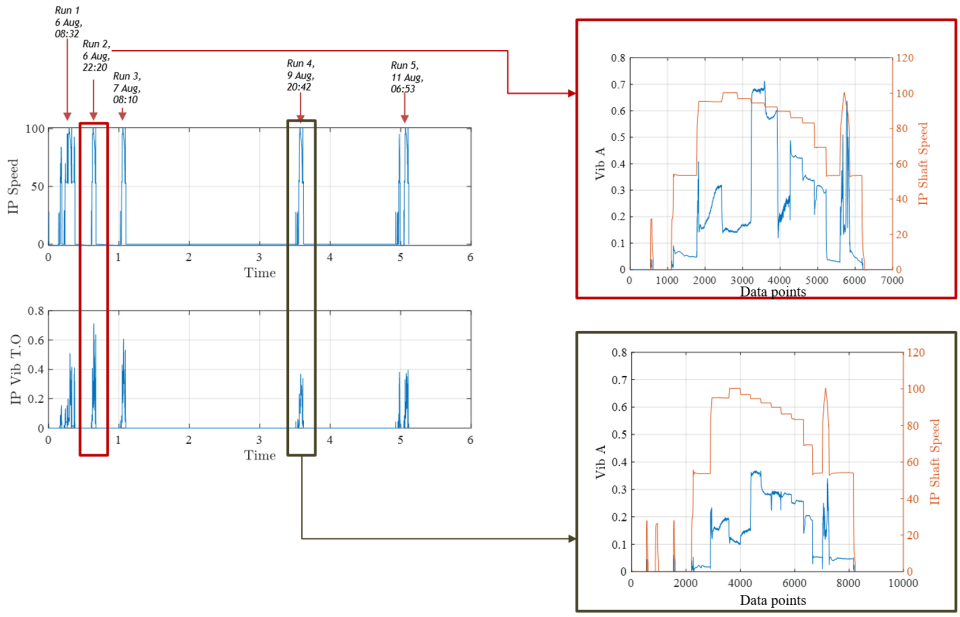


Figure 2.1: Series of five different engine pass off tests with zoomed in views of Run 2 and Run 4 showing the variability in IP Vibrations, even at expected "similar loading" conditions.

variability of tracked order vibration magnitude at the same levels of speed. In the case of Run 2, tracked order vibration is almost twice as in the case of Run 4. Further, significant variations are observed even within each run, which depicts the nonlinear characteristics of engine vibration.

2.2 Data-driven modeling techniques

The recent and rapid development of measurement techniques and consequent availability of measurement data has propelled the development of many data-driven methods for modeling and predicting dynamics. At the forefront of data-driven methods are machine learning methods, such as deep neural networks (DNN), which have found their applicability from image recognition [30] to effectively predicting the future state of dynamical systems [31, 32, 33]. A key challenge to data-driven methods such as DNNs is the lack of interpretability of the resulting model, which further impedes the applicability of such methods in practical cases. Nonetheless, alternate methods such as symbolic regression directly identify the structure of a nonlinear dynamical system from data but are computationally expensive and seriously limit their applicability to small scale problems [34, 35]. In the next section, the various data-driven methodologies amenable towards dynamical systems and their interpretation thereof are surveyed, and their basic ideas are presented in brief detail. Focus is given towards physically interpretable methods rather than complicated black-box models that require a

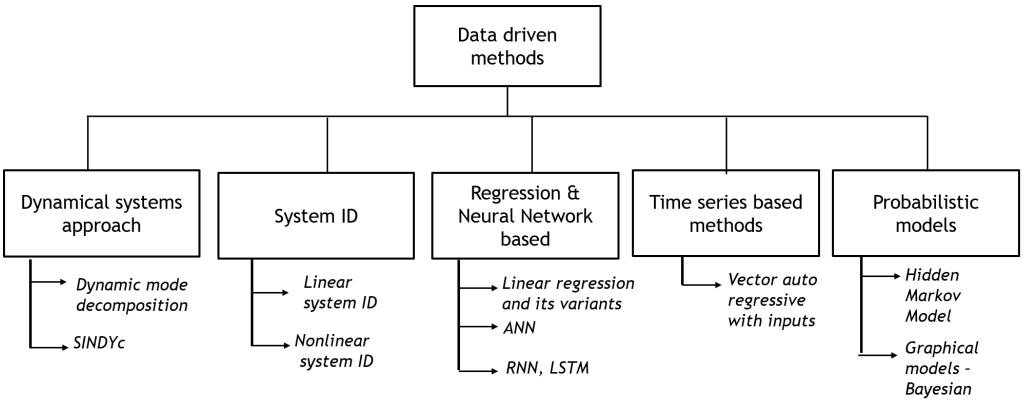


Figure 2.2: Existing literature on data driven models for dynamical system

significant amount of training data sets and lack physical interpretation.

The primary motivation of the present work, as explained before, is to build a data-driven model of the engine response using the sensor data collected from various types of instrumentation within the engine. Any modeling methodology selected should ideally predict the tracked order vibration given a different set of input conditions. The challenges for the present work are two-fold; i.) Identifying the most critical sensor information, and ii.) Building the data-driven engine vibration model. Once these two tasks are accomplished, sensitivity and exchange factors can be established for each tracked order vibration. The most significant challenge in modeling a dynamical system stems from the fact that most of the sensor data are temporally and spatially correlated. The temporal correlation, especially for the tracked order vibration, can extend for a long time period, thus resembling a long memory process. Nevertheless, the literature review to model the tracked order vibration, using the historical sensor data, can be summarized as shown in fig. 2.2. Only those methods that are amenable towards dynamical system implementation and which can incorporate the sequential behavior of the data are discussed in the literature survey. This shortens the survey into five broad categories, which have a lot in common and overlapping research. These are:

1. Dynamical systems approach
2. System identification approach
3. Machine learning methods - Regression & Neural network based
4. Time series based methods
5. Probabilistic models

Each of these methods are discussed in brief detail with their relative advantages and disadvantages with examples of their usage in literature.

2.2.1 Dynamical system approach

Modeling nonlinear dynamical systems using data is fundamental in a variety of engineering and scientific fields. Spectral analysis (or, referred to as the decomposition technique) for dynamical systems is a popular approach to extract information concerning (low-dimensional) dynamics from data. The class of mathematical methods which utilizes the systems theory approach and approximates the overall dynamics from a linear system theory can be classified under these methods. The two main methods that can naturally be extended to the present work are i.) Dynamic Mode Decomposition (DMD) and ii.) Sparse identification of nonlinear dynamical system (SINDy) algorithm.

Dynamic mode decomposition is an equation-free data-driven approach that was first developed for fluids that evolve on their own. However, many real-life dynamical systems such as the aero-engine involve controls and are driven by a set of inputs. Towards the same Proctor et al. [36], developed a variant of DMD known as DMD with controls (DMDc) to incorporate the input signals into the DMD framework. Recently, Benner et al. [37], developed an extension of DMDc, known as Input-Output DMD (ioDMD), by providing means to incorporate outputs alongside inputs and states. The pitfall of using a plain run of DMD on complex systems has been mentioned [38] especially in the framework of video processing for feature extraction. Details regarding the theory of dynamic mode decomposition and its different variants will be explained in the future chapter in great details.

Advanced machine learning methods have paved the way for new innovation in nonlinear system identification. The sparsity promoting models [39, 40] have been popular as they provide a trade off between the model complexity and data fit. Sparse regularization, which has shown its efficacy for parameter and structure identification, can promote robustness and generalizability in models. In 2016, Brunton et al. [41], addressed the discovery of dynamical system from the perspective of sparse regression and named their method as sparse identification of nonlinear dynamical system (SINDy), which was later extended to include input variables [42]. The SINDy method in particular leverages the fact that there only only few relevant terms that define the dynamics, thereby making the governing equations sparse in a high-dimensional nonlinear function space. A brief overview of the SINDy algorithm is as follows. Consider a dynamical system of the form,

$$\dot{x} = f(x(t)), \tag{2.3}$$

where $x(t) \in \Re^N$ is the state of the system at time t and $f : \Re^N \rightarrow \Re^N$ defines the dynamics of the system. A key challenge in the study of dynamical systems is to find the structure of the function f. SINDy addresses this issue by forming a library of candidate functions and

utilizing sparse regression finds the right candidate. Given the snapshot data,

$$\mathbf{X} = \begin{pmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \cdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{pmatrix}, \ \dot{\mathbf{X}} = \begin{pmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \cdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{pmatrix}, \tag{2.4}$$

the SINDy methods buids the library of candidate nonlinear functions $\Theta(X) = [\theta_1(\mathbf{X})....\theta_p(\mathbf{X})]$. The algorithm uses the thresholded least squares to find the sparse coefficient vectors $\mathbf{E} = [\epsilon_1, \epsilon_2...., \epsilon_n]$ to approximately solve,

$$\dot{\boldsymbol{X}} = \Theta(X)\mathbf{E}. \tag{2.5}$$

The basic framework of SINDy with input controls is similar to SINDy with the difference that nonlinear terms of the control variable are also considered while building the candidate reference nonlinear functions [42]. The disadvantage with SINDy method is that it requires all the states of the system being measured. However, for most of the complex dynamical system such as the aero-engine, only data representing some form of the internal dynamics is readily available and this seriously restricts its applicability in the current work. Authors of the original SINDy paper have recently extended SINDy to problems where the original states are not directly observable, by combining the Hankel methodology within the SINDy framework [43], thus elevating the method as a good candidate to be used in modeling aeroengine vibrations.

2.2.2 System ID based methods - Nonlinear

System identification can be defined as the methodology for building mathematical models of dynamic systems using measurements of the system's input and output signals. The present work aims to build a data driven model of the aero engine vibrations, using the data collected from different types of instrumentation housed inside the engine. System identification provides a principled solution to this problem. The advantages of system identification are many: it is applicable to all systems, it is often quick, and can be made to track changes in the system. The biggest challenges with system id is to obtain the basic structure of the model, i.e., whether the system contains any non-linear term, if so, what type of non-linearity, time-delay terms, etc.[44].

Mathematically, system ID can be explained as follows. Given a dynamical system with input $u(t)$ and output $y(t)$, the system identification tries to relate the output to the past inputs and outputs [45, 46]. This can be related as follows,

$$\hat{y}(t) = \tilde{g}(y(t-1), u(t-1), y(t-2), u(t-2), \ldots). \tag{2.6}$$

The model is nonlinear if the function $\tilde{g}$ is nonlinear. The nonlinear system identification can be formulated from the perspective of an extended nonlinear linear regression problem. Mathematically, the problem translates to find the nonlinear relationship between the system outputs $y(t)$ and a known regression vector $\psi(t)$:

$$y(t) = g(\psi(t)) + e(t). \tag{2.7}$$

The regression vector $\psi(t)$ is a function of all inputs $u(s)$ and outputs $y(s)$ prior to time t:

$$\psi(t) = h(y(t-1), u(t-1), y(t-2), u(t-2), ...). \tag{2.8}$$

Dynamical systems can either be linear or nonlinear and can often be formulated in continuous-time or discrete-time with lumped-parameter or distributed-parameter. There are various classification of system ID methods, the most popular: (i) linear system ID techniques, (ii) nonlinear system ID techniques. The autoregressive moving average with exogenous input (ARMAX) model (including subsets of AR (autoregressive), ARMA (autoregressive moving average), and ARX (autoregressive with exogenous input) models [47]) is the most commonly used representations for linear system identification. For the nonlinear system ID, the nonlinear autoregressive moving average with exogenous input (NARMAX) models [48], volterra models [49, 50] and the state space models [51, 52] are the most popular representations for nonlinear dynamic system identification in the discrete-time domain [53]. For a brief review of different system ID methods, with a special focus on nonlinear system identification, the interested readers are directed to [44].

Nonlinear state-space methods and nonlinear AR models with exogenous inputs are popular choices amongst researchers for modeling complex behavior, such as the case of gas turbine vibrations. Noël et al., [52] modeled and experimentally verified the bouc-wen nonlinear system using an approach based on the nonlinear state-space method. As explained in the earlier section, the tracked order vibration exhibits hysteresis behavior, and capturing the essence can be challenging even with state-of-the-art methods. Rahmoune et al., [54] combined the NARX model with neural networks to build a meta-model for control and diagnostics of gas turbine vibration. It should be noted that nonlinear system identification is a rich topic with challenging problems, which motivates researchers to apply many of the developed and interpretable linear methods towards nonlinear systems. For a good review of the applicability of linear methods in the context of nonlinear systems, interested readers are directed towards [55].

2.2.3 Machine learning based methods

Supervised machine learning in the context of a dynamical system can be defined as the means of modeling the input-output relations of the system using conventional supervised methods such as Support Vector Machines, linear regression, etc. The complexity of mod-

eling a dynamical system stems from the time element(sequential data) associated with the measurements/states of the system. This temporal component, which is omnipresent in almost all dynamical systems, makes the whole problem challenging and is considered by Yang and Wu [56] as one of the top 10 challenging problems in data mining due to its unique properties. The data set consists of multiple sensor information in a real engine, and thus they constitute a multivariate time series modeling and prediction problem. Multivariate time-series modeling and forecasting is an important problem with numerous applications in several real-world domains such as aviation, finance, climatic studies, healthcare, etc. [57]. The major challenge in addressing engine vibration using a multivariate time series analysis lies in capturing the proper temporal variations for each flight sensor as well as the possible correlations between them at each time step.

The vibration signal produced in the running process of the gas turbine engine has the characteristics of the time series. Traditional time series models, such as Auto-Regressive(AR) model, Moving Average(MA) model, and the Auto-Regressive Moving Average(ARMA), is widely used because of their simple linear structure [58, 59]. The linear models' applicability is limited to linear systems, describing linear relationships between variables. Since the models to be considered are from systems, it calls for a multivariate time series analysis. Vector autoregressive (VAR) time series models can account for the exogenous inputs. Intelligent models [60, 61, 62] such as Artificial Neural Network (ANN), can process complex nonlinear data and capture the inherent characteristics of data, has been widely studied in recent years [63]. The recent advancements in a deep neural network have further catalyzed the research of dynamical systems in the purview of machine learning, with the development of recurrent neural networks in parallel [64]. The recurrent neural network suffers from capturing long-term dependencies in certain stochastic processes due to the problem of vanishing gradients [65]. This led to the development of long short term memory (LSTM) networks in 1997 [65], which can learn the dependency structure over longer sequences. LSTM RNN was first introduced by S. Hochrieter & J. Schmidhuber [65]. The paper addressed the limitations in the recurrent neural network, i.e., the vanishing/exploding gradients. This study paved the way for much interesting application for LSTM. LSTM RNN's have been widely used with strong performance in many areas such as image recognition [66], music composition [67], and audiovisual emotion recognition [68]. Central to the success of LSTM in capturing long-term dependencies are its gated structure which has the dual ability to conserve the information as well as forget fruitless dependencies over long periods of sequenced data [17].

There exists a slew of successful applications of time series and machine learning methods towards modeling aviation data. VAR models have been used in the framework of aviation data modeling in recent years. Melynk et al. [69] used a VAR-based approach to model the Flight Operations Quality Assurance (FOQA) data and used them to detect abnormal activities in the flight. Shirazi et al. predicted vibration rate in a real gas turbine using multi-layer perceptron neural network and cuckoo optimization algorithm and conducted a sensitivity analysis using cosine amplitude method [70]. In parallel, LSTM [18, 71, 72] methods have made significant progress and have found applicability in the field of aviation

time series modeling [71]. Goel et al. [73] did an empirical comparison between VAR and LSTM for modeling a multivariate aviation time-series and concluded that VARs outperform LSTM, at least in the case study they considered. One of the significant findings of the study is that since LSTMs are highly non-linear models with several choices for their architecture, identifying a suitable architecture for a problem domain can be more challenging compared to VAR models. Elsaid et al. [74] utilized long short term memory (LSTM) recurrent neural networks (RNNs) of different types to predict engine vibrations and other critical aviation parameters. In the same study, LSTM models were used to predict vibration values for 5, 10, and 20 seconds in the future, with 3.3%, 5.51% and 10.19% mean absolute error, respectively. The authors further improved the prediction error by using an ant colony optimization technique to address the vanishing gradient problem.

2.2.4 Probability based methods - HMM and particle filtering

The whole suit of methods toward learning the parameters of the model using a probabilistic framework can be categorised under these subsection. The advantage with the probabilistic model stems from the notion that, it offers the dual provision of simultaneously learning the parameters of the model, as well to manipulate uncertainty in data, models, predictions and decisions. For interest readers in probabilistic modeling in the context of machine learning and system identification, are directed towards [75, 76, 77]. These methods can be explained in the basis of a nonlinear system identification problem, particularly of the form shown below,

$$\begin{aligned} x_t &= f(x_{t-1}, u_t, v_t, \theta) \\ y_t &= g(x_t, u_t, \theta) + e_t \\ x_0 &\sim p(x_0|\theta) \\ \theta &\sim p(\theta) \end{aligned} \tag{2.9}$$

where $y_t \in Y$ denotes the observed output and u_t denotes a known input. The unknown variables x_t describes the system's evolution over time and static parameters are represented as θ.

The methods that are based out of probabilistic laws for linear dynamical system aim to model the joint distribution of the states and static parameters conditioned on the observed data. The conditional distribution of the unknown $x_{0:T}\&\theta$ conditioned on the data $y_{1:T}$, where the subscript T denotes the last data point, can be denoted as $p(x_{0:T}|y_{1:T})$. Once the conditional probability is computed, one can infer meaningful statistical parameters such as mean values, variances, or estimates. The challenge for such a conditional probability is that there exist no closed form solution and one has to resort to sampling and approximation algorithms such as Monte Carlo Markov chain and Gibbs sampling [78, 79]. Eq. 2.9 can also be expressed as,

$$\begin{aligned} &x_t|x_{t-1}, \theta \sim p(x_t|x_{t-1}, \theta), \\ &y_t|x_t, \theta \sim p(y_t|x_t, \theta), \\ &\theta \sim p(\theta). \end{aligned} \tag{2.10}$$

The full probabilistic model of eq. (2.10) can be derived and simplified by using the laws of conditional probability, and the final expression is as shown [80],

$$p(x_{0:T}, \theta, y_{1:T}) = \left(\prod_{t=1}^{T} \underbrace{p(y_t|x_t, \theta)}_{observation} \right) \underbrace{\left(\prod_{t=1}^{T} \underbrace{p(x_t|x_{t-1}, \theta)}_{dynamics} \right) \underbrace{p(x_0|\theta)}_{initial} \underbrace{p(\theta)}_{param}}_{prior} . \tag{2.11}$$

The main aim of such a probabilistic framework is to estimate the "posterior distribution", given the observed data, which can be expressed as $p(x_{0:T}, \theta|y_{1:T})$. Using the principles of conditonal probability, it can be seperated as,

$$p(x_{0:T}, \theta|y_{1:T}) = \underbrace{p(x_{0:T}|\theta, y_{1:T})}_{state\ inf} \underbrace{p(\theta|y_{1:T})}_{param.\ inf} . \tag{2.12}$$

The parametric inference, essentially the posterior distribution can be solved using the bayesian principle as,

$$p(\theta|y_{1:T}) \propto L(\theta|y_{1:T}) \times p(\theta). \tag{2.13}$$

With the advancement in computational complexity, eq. (2.13) can easily be solved using techniques such as Gibbs sampling, Metropolis - Hastings etc, provided we know how to form the likelihood function 'L' and use it iteratively in the Gibbs sampling methodology.

As mentioned before, the learning of a nonlinear dynamical system can be cast as two separate problems i.) Nonlinear state estimation ii.) Parameter estimation. In literature the nonlinear state estimation is addressed using sequential Monte Carlo methods, which are commonly referred to as particle methods [80]. But most of the practical problems can be assumed to have a linear structure, which can be exploited using the Rao-Blackwellized particle filter [81]. These type of models find their applicability in localization and target tracking applications. The parameter estimation for linear dynamical systems have been well studied in literature and expectation maximization [82] is still a popular method, to estimate the parameters such as the state space matrices. The disadvantage with many of the likelihood based approaches such as EM and MCMC is that the learning can be slow and it can suffer from local optima. More recently, a new class of methods called "spectral algorithm" have evolved, which rely on factorization of a matrix of observable moment, which has the advantage of being fast, simple, and globally optimal [83, 84]. For a detailed overview of different statistical learning methods for dynamical system and its utilization for building a hybrid model using state of the art recurrent neural networks, the interested readers are directed towards to [85].

2.3 Challenges associated with modeling the dynamic response of an aero-engine using a data-driven approach

The main aim of the present work is to model the aero-engine vibrational response from a holistic point of view, by considering the various interplay dynamics and their effect on the tracked engine order. Towards the same, the complex dynamics between various operational parameters has to be modeled using various state of the art modeling techniques. But, modeling a dynamical system can be extremely challenging for a number of reasons [85, 86]:

1. *Transitions/Observations are noisy* : In practice most of the dynamical system are stochastic, with transition/behaviour which are highly random. This entails extracting the model dynamics from noise and which has the risk of learning spurious correlations and performing poorly.

2. *State's not directly observed* : Most of the dynamical system are "partially observable", i.e., it is not possible to observe the states directly, but only the observations, which could be some linear or nonlinear combination of the states. Thus, in state estimation or inference, this would require learning the states from noisy observations without ever actually seeing it.

3. *For nonlinear system's, the unknown structure of f() or g()* : In simple linear systems, it is often easier to learn the relationship between one state and the next. But in the case of more complex systems, like the aero-engine, the unknown structure of the f and g further complicates the process.

4. *Generalizing to unseen inputs and states* : The core idea of learning a dynamical model from the collected data set is to simulate the behaviour of the system to future inputs and also to characterise the system to its states and inputs. But if nonlinearities are involved, the system characteristics would evolve as a function of the inputs, which adds a new layer of complexity to the whole process, which cannot be captured using just one type of input data set.

Chapter 3

Data-driven modeling and analysis of tracked order vibration in turbofan engines under different operating conditions

Vibration monitoring in aircraft engines is traditionally performed using order tracking methodology. Currently, there are no representative physics-based models for tracked order vibration in aircraft engines. This gap in research is primarily attributed to the limited understanding of mutual interactions and the non-linear nature of engine vibrations. The objective of the current study is two-fold : (i) to present a preliminary investigation of tracked order vibrations in aircraft engines, and to statistically analyze them in the context of their operating environment, and (ii) to leverage emerging data-driven modeling techniques to develop accurate models for the tracked order vibration in turbofan engine valid over a wide range of operating conditions. Towards the same, we employ multi-scale detrended partial cross-correlation analysis to evaluate the interdependencies and cross-correlation structure between various thermo-mechanical variables and tracked order vibration across multiple time scales. We leverage the understanding of complex correlation structures and subsequently utilize them to model tracked order vibrations. We develop data-driven models using state-of-the-art time-delay embedded dynamic mode decomposition and compare its performance with Lasso regression.

The contents of this chapter are reproduced from, ***M. Krishnan, I. Sever, and P. A. Tarazaga, "Data-driven modeling and analysis of tracked order vibration in turbo-fan engine under different operating conditions"****, which is currently under review by AIAA Journal.*

3.1 Introduction

Data-driven models rely entirely on the measured input-output responses. Hence, their success depends on a deeper understanding of the cross-correlation and interaction between the target variable (tracked order vibration) and predictor variables (measured temperature, pressure, and other parameters from the engine). However, engine vibration is a complex problem depending on several unknowns and various parameters interacting over an extended period, which poses a challenge in building a data-driven model [8, 13]. Detecting cross-correlations between signals is the most common way to improve the understanding of a complex system [14]. Due to its simplicity, Pearson cross-correlation analysis has been employed in the literature to analyze the inter-dependencies of various signals. However, most of the signals obtained from nature and many complex dynamical systems are highly non-stationary and involve non-linear interactions, thus changing their correlation structure over time. This motivates the need for a cross-correlation index spanning over multiple time scales to capture the changing correlation structure [87]. The multi-scale detrended cross-correlation analysis (MS-DCCA) technique has become increasingly popular to analyze non-stationary signals in various fields such as climatology [14, 88], ecology, and economy [87] for detecting cross-correlations across multiple time scale. The main advantage of the detrended method is that it allows the detection of cross-correlations between noisy signals with non-stationarities [87] that can mask the true cross-correlations in the fluctuations of signals. This makes it an ideal candidate for analyzing interdependencies among various parameters and the tracked order vibration [13], thereby paving the way its use as a tool for feature selection in the present study.

Once the right features are identified, data-driven modeling methodologies can be employed for model building. This is a heavily studied topic with varying approaches based on the eventual goal. At the forefront are supervised machine learning methods such as regression [15], artificial neural networks [16], and LSTM [17], which minimizes an appropriate cost function [16]. However, neural networks lack the required capabilities to capture tho interdependencies and long scale correlations as they work on current time series without taking the effect of the previous time instances' parameters on the current or future time instants [18]. This led to the development of RNNs and LSTMs, which retain information from the previous time instants. Nonetheless, these complicated designs pose their own challenges, especially regarding the vanishing gradients and the heavy computations required in the training process. In recent years the operator theoretic framework based on Koopman operator [19] has garnered interest in the dynamical system community [20, 21, 22]. One such method is the Dynamical mode decomposition (DMD) [20], which became a popular tool among researchers for data-driven identification, owing to its ability to decompose high dimensional fluid flow data into its coherent spatio-temporal structures [22, 23]. For low dimensional systems, variants of DMD known as time-delay DMD and Wavelet-based DMD have found successful application [24, 25, 26].

In this paper, we analyze the tracked order vibration in turbofan engines under different

operating conditions (OC) in the context of their temperature evolution. We utilize Gaussian mixture models to cluster various subpopulations within each OC and analyze their effect on tracked order vibrations. Furthermore, we present a feature selection methodology utilizing the MS-DPCCA procedure to identify the most important variables in building a data-driven model for the tracked order vibration. This is a preliminary investigation towards applying a data-driven method in solving a complex dynamical system consisting of millions of moving parts. Finally, we provide preliminary results of applying the Delay-DMD procedure and Lasso regression models in building a switching linear dynamical system model for the tracked order vibration in aero-engine.

The paper is organized as follows. First, a brief description of tracked order vibration, MS-DPCCA, DMD, and Delay-DMD are presented in section 3.2. In section 3.3, we introduce the experimental data-set alongside definitions of all the operational conditions. Section 3.4 analyses the tracked order vibration in the context of different operating conditions and applied MS-DPCCA towards understanding the interdependencies among various parameters and the tracked order vibrations. Finally, in section 3.5, the results of modeling the tracked order engine vibrations using Delay-DMD and Lasso regression are presented.

3.2 Background

3.2.1 Tracked order vibration

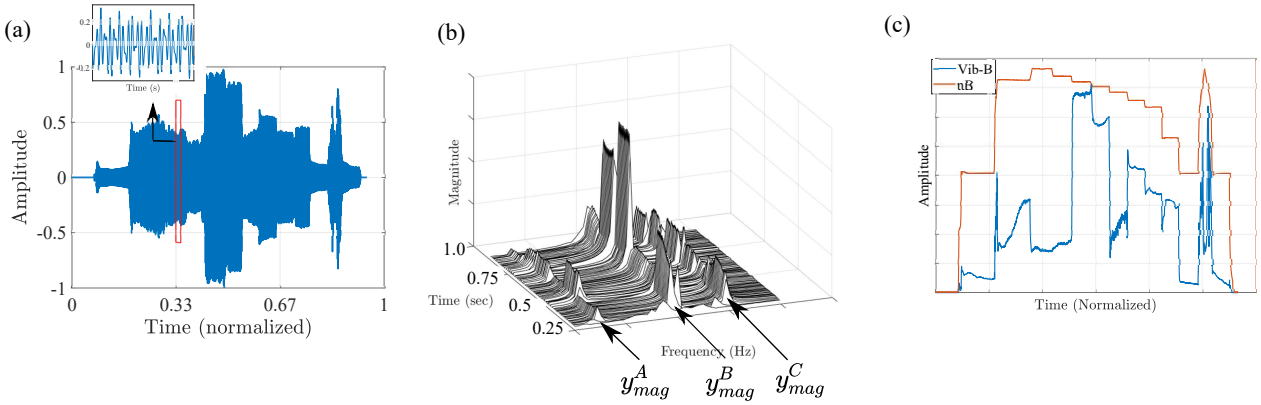


Figure 3.1: Schematic diagrams illustrating the order tracking principle: (a) Time-domain acceleration data (b) Waterfall plot of the frequency spectra, and (c) Tracked order vibration amplitudes plotted alongside the corresponding shaft rotations.

In the present section, we provide a brief background about tracked order vibration. Consider measured acceleration response from an aircraft engine operating at a constant speed. Fourier transform on this 'steady-state' vibration signal indicates the frequency content of

the vibrations at that particular speed. If the same procedure is repeated for a prescribed window length while the engine is undergoing a slow coast-up from idle to maximum power, we can construct a simple waterfall plot of all frequency spectra of the analyzed time samples. Since the rotational speed is varying with the time ($(\omega = \omega(t))$, the frequency components are also time-varying and are called orders. The orders can be defined as a time-varying phasor with a phase value related to the rotational frequency of the reference shaft. It generates a sinusoidal function with varying frequencies. In a turbofan engine with concentrically rotating shafts, the frequency content of interest often lies in the fundamental shaft harmonics (order = 1). This translates to three predominant frequencies of interest, corresponding to each of the shaft speed as shown in Figure 3.1 (b). Mathematically, a single order is described as,

$$\psi(t) = Z_p(t)e^{i2\pi\frac{p}{T}t}, \tag{3.1}$$

where $\psi(t)$ is the measured single order signal, $Z_p(t)$ is a complex number representing the amplitude and phase of the p^{th} order, and T is the time period of the primary order in seconds. The value of $Z_n(t)$ is not a constant, and it usually depends on the rotational speed and other structural conditions like temperature, initial conditions, etc. For a rotatory machine, several orders are present due to several components and their harmonics, and hence the final acceleration signal can be thought of as a summation of all the time-varying phasors as,

$$y(t) = \sum_{n=-\infty}^{n=\infty} Z_p(t)e^{i2\pi\frac{n}{m}t}. \tag{3.2}$$

The objective of the order tracking step is to estimate $Z_p(t)$ by splitting up the response $x(t)$ into its sinusoidal component. The order n is set in advance and, in the case of aircraft engines, are integral multiples of the rotational frequency of the reference shaft. In the present study, we focus our attention on the fundamental shaft orders (i.e., order (n) = 1). Traditionally, order tracking techniques can be classified into 4 categories: (i) Time-domain sampling-based Fast Fourier Transform order tracking, (ii) Resampling-based order tracking, (iii) Time Variant Discrete Fourier Transform, and (iv) Vold-Kalman filter-based order tracking (VK). Interested readers are directed to [27, 28] for a rigorous explanation of each technique.

Figure 3.1 demonstrates the procedure of obtaining tracked order responses from a measured vibration signal. Windowed fourier transform translates the original time-domain into corresponding frequency samples which are stacked one behind each other as shown in Figure 3.1 (b). The present case demonstrates an engine with 3 concentrically rotating shafts, whose fundamental orders are denoted as 'y^A_{mag}', 'y^B_{mag}', and 'y^C_{mag}', respectively. The individual tracked order signals $Z_p(t)$ are extracted and are subsequently analyzed. For demonstration purposes, the 'y^B_{mag}' tracked order signature is overlaid with its corresponding shaft speed and is shown in Figure 3.1 (c).

As mentioned before, the tracked order curves plotted against its corresponding shaft rpm (hereafter called characteristic curves) functions parallel to the traditional frequency response

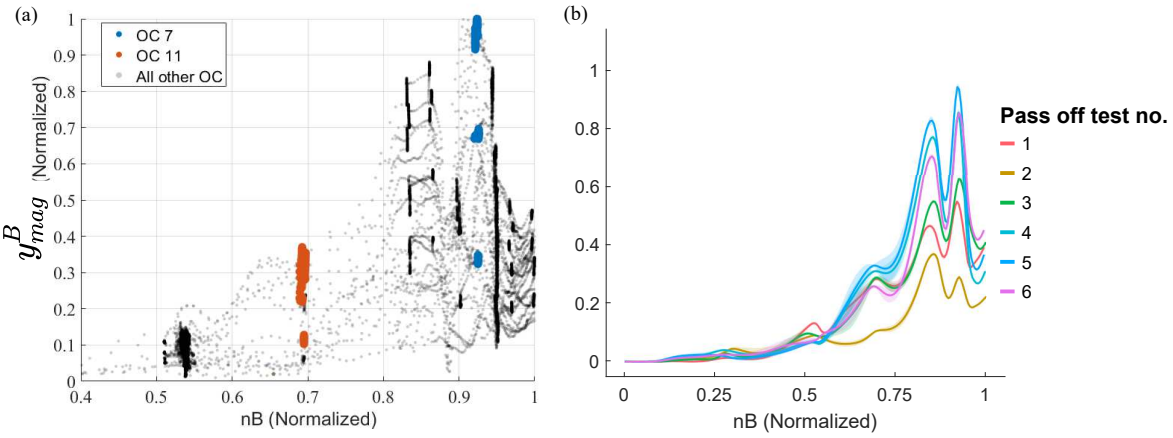


Figure 3.2: Characteristic curve of B in : (a) raw unfiltered form, (b) filtered and smoothed curve

curve that can be used for system identification, anomaly detection, and modal analysis. Figure 3.2 shows the tracked order values plotted against its corresponding shaft rpm values for an engine undergoing several pass-off tests within few days. As mentioned before, the pass-off test consists of a series of controlled maneuvers where engines are run at a different set of speeds, as shown in Figure 3.1 (c). For demonstration, two specific speeds designated as operating conditions (i.e., OC7 and OC11) are marked with blue and red, respectively. The plot also illustrates variability and dispersion in tracked order vibration within each operating condition. Figure 3.2 (b) shows the smoothed characteristic curve for individual runs, further illustrating the variability from engine run to run for a particular tracked order vibration. Figure 3.2 (b) also reveals the shortcomings of the current order tracking methodology by bringing out the variability in vibrations even when excited by identical conditions, albeit in different pass-off tests. The characteristic curves also point to resonance peaks in the normalized speed range $0.8 - 0.85$, and the speed ranges $0.94 - 0.96$.

3.2.2 Multi-scale detrended partial cross correlational analysis (MS-DPCCA)

Measurements made on the aircraft engine provide a window through which internal dynamics and causal relationships can be studied. In the present study, we utilize MS-DPCCA to analyze interdependencies between various gas path variables such as temperature, pressure, and the corresponding tracked order vibration signal. MS-DPCCA is selected for analyzing the signals as the methodology can provide: (i) cross-correlation at different scales, (ii) extract "intrinsic" relationship between two considered time series with possible influences of other common-coupled signals removed. For the first point, detrended-cross correlation analysis (DCCA) [87] may be solution. DCCA has found applications ranging from climatic

studies [88] to finance studies and even to analyze gas path signals inside an aero-engine [13]. However, DCCA, as mentioned before, does not take into account the effect of common-couple signals, and aero-engine is essentially a dynamical system with a huge number of common-coupled signals. Thus, in the present analysis, authors utilize MS-DPCCA, which can remove the effect of external signals while analyzing two coupled signals. The method of de-trended cross-correlation is explained next.

Consider an N dimensional system sampled discretely as $x^j(t_i)$ where $i = 1, ..., K$ represents the time indices and $j = 1, ..., N$ denote the variable index. Each time series $x^j(t)$ can be considered a random walk and their profile at any time instant k can be written as follows,

$$P_k^j \equiv \sum_{i=1}^{k} x^j(t_i). \tag{3.3}$$

The total sample size K can be divided into $K - s$ overlapping boxes, each with sample sizes of $s + 1$, starting at i and ending at $i + s$. Furthermore, the local detrended walk in each box i is calculated by subtracting with a polynomial fit as,

$$W^j(t_{(i-1)(s+1)+k-i+1}) = P_k^j - \widetilde{P_{k,l}^J}. \tag{3.4}$$

Similarly, de-trended walks W^j are obtained for each of the time series. Furthermore, the covariance between any two detrended walks are obtained as,

$$F^2_{j_1,j_2}(s) \equiv \frac{\sum_{l=1}^{(K-s)(s+1)} W_l^{j_1} W_l^{j_2}}{(K-s)(s+1)}, \tag{3.5}$$

where j_1 and j_2 represent the two different series. Extending the same procedure for the all the signals leads to formation of covariance matrix as,

$$\mathbf{F}^2(s) = \begin{pmatrix} F^2_{1,1}(s) & F^2_{1,2}(s) & \dots & F^2_{1,N}(s) \\ F^2_{2,1}(s) & F^2_{2,2}(s) & \dots & F^2_{2,N}(s) \\ \vdots & \vdots & & \vdots \\ F^2_{N,1}(s) & F^2_{N,2}(s) & \dots & F^2_{N,N}(s) \end{pmatrix}. \tag{3.6}$$

The de-trended cross-correlation between any two series $x^{j_1}(t)$ and $x^{j_2}(t)$ can be represented as [89],

$$\rho_{j_1,j_2}(s) \equiv \frac{F^2_{j_1,j_2}(s)}{F_{j_1,j_1}(s) \cdot F_{j_2 j_2}(s)}, \tag{3.7}$$

where $\rho_{j_1,j_2}(s)$ represents the s-correlation between the two series j_1 and j_2 at a particular scale s and ranges between -1 and 1. The above mentioned procedure is extended to include interactions between each pair of time series in the dynamical system leading to formation

of a detrended correlation coefficient (DCCA) matrix as,

$$\boldsymbol{\rho}(s) = \begin{pmatrix} \rho_{1,1}(s) & \rho_{1,2}(s) & \dots & \rho_{1,N}(s) \\ \rho_{2,1}(s) & \rho_{2,2}(s) & \dots & \rho_{2,N}(s) \\ \vdots & \vdots & & \vdots \\ \rho_{N,1}(s) & \rho_{N,2}(s) & \dots & \rho_{N,N}(s) \end{pmatrix}. \tag{3.8}$$

The DCCA coefficients reflect the relationship between the two time series and might give erroneous results if these time series are correlated with other signals, such as in the case for the aero engine [14]. To account for the effect of other signals, partial cross-correlation is incorporated into the DCCA procedure, yielding DPCCA [14]. This involves inverting the $\boldsymbol{\rho}(s)$ matrix as,

$$\mathbf{C}(s) = \boldsymbol{\rho}^{-1}(s) = \begin{pmatrix} C_{1,1}(s) & C_{1,2}(s) & \dots & C_{1,N}(s) \\ C_{2,1}(s) & C_{2,2}(s) & \dots & C_{2,N}(s) \\ \vdots & \vdots & & \vdots \\ C_{N,1}(s) & C_{N,2}(s) & \dots & C_{N,N}(s) \end{pmatrix}. \tag{3.9}$$

This entails writing the DPCCA correlation between any two signals j_1 and j_2 as a function of the scale s and can be written as,

$$\boldsymbol{\rho}_{DPCCA}\,(j_1, j_2; s) = \frac{-C_{j_1,j_2}(s)}{\sqrt{C_{j_1,j_1}(s) \cdot c_{j_2,j_2}(s)}}, \tag{3.10}$$

where ρ_{DPCCA} quantifies the intrinsic relations between the time series at a scale s. This procedure is repeated for different values of time scales.

3.2.3 Time-delay embedded dynamic mode decomposition (Delay-DMD)

In this section, we provide a brief introduction to classical DMD framework [20, 90]. To this aim, consider the time invariant ordinary differential equation of the form,

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)), \tag{3.11}$$

where state $x(t) \in \Re^N$ is the state vector and $f : \Re^N \to \Re^N$ is a nonlinear map. The most fundamental form of DMD, as given by [20], approximates the discrete dynamical linear system of the form,

$$\mathbf{x}(t_{k+1}) \approx \mathbf{A}\mathbf{x}(t_k), \tag{3.12}$$

with $\mathbf{A} \in \Re^{N\times N}$ assuming that this holds for all k from $k = 0, 1, 2, .., K-1$. The two data snapshot matrices $\mathbf{X}_0$ and $\mathbf{X}_1$ are defined as,

$$\mathbf{X}_0 = \begin{bmatrix}\mathbf{x}(t_0) & \mathbf{x}(t_1) & \cdots & \mathbf{x}(t_{K-1})\end{bmatrix} \in \Re^{N\times K} \text{ and } \mathbf{X}_1 = \begin{bmatrix}\mathbf{x}(t_1) & \mathbf{x}(t_2) & \cdots & \mathbf{x}(t_K)\end{bmatrix} \in \Re^{N\times K}, \tag{3.13}$$

where $\mathbf{X}_1$ is a unit time step advanced version of $\mathbf{X}_1$. The goal of DMD algorithm is to find the best-fit solution of an operator $\mathbf{A}$ such that the following relation is (approximately) satisfied,

$$\mathbf{X}_1 \approx \mathbf{A}\mathbf{X}_0. \tag{3.14}$$

The above mentioned equation can be written explicitly as follows,

$$\mathbf{A} = \mathbf{X}_1\mathbf{X}_0{}^{\dagger}, \tag{3.15}$$

where $\mathbf{X}_0{}^{\dagger} \in \Re^{K\times N}$ denotes the Moore-Penrose inverse of matrix $\mathbf{X}_0 \in \Re^{N\times K}$. The practical computations of the $\mathbf{A}$ matrix involves algebraic assumptions and singular value decomposition of the $\mathbf{X}_0$ matrix, which are not mentioned here for brevity. Interested readers are directed to [20, 91] for more rigorous exercise on practical algorithm and computations.

For systems driven by external inputs $\mathbf{u}(t)$, DMDc [36] and its variant, the input-output DMD (ioDMD), approximates a linear discrete dynamical system of the form,

$$\begin{aligned}\mathbf{x}(t_{k+1}) &= \mathbf{A}\mathbf{x}(t_k) + \mathbf{B}\mathbf{u}(t_k),\\ \mathbf{y}(t_k) &= \mathbf{C}\mathbf{x}(t_k) + \mathbf{D}\mathbf{u}(t_k),\end{aligned} \tag{3.16}$$

where $\mathbf{A} \in \Re^{N\times N}, \mathbf{B} \in \Re^{N\times M}, \mathbf{C} \in \Re^{P\times N}, \mathbf{D} \in \Re^{P\times M}$, assuming M inputs, N states and P outputs are measured. In addition to the $\mathbf{X}_0$ and $\mathbf{X}_1$ matrix, the system also includes input data snapshot matrix $\mathbf{U}_0$ and the output $\mathbf{Y}_0$ matrix as given,

$$\mathbf{U_0} = \begin{bmatrix}\mathbf{u}(t_0) & \mathbf{u}(t_1) & ... & \mathbf{u}(t_{K-1})\end{bmatrix} \in \Re^{M\times K}, \text{ and } \mathbf{Y}_0 = \begin{bmatrix}\mathbf{y}(t_0) & \mathbf{y}(t_1) & ... & \mathbf{y}(t_{K-1})\end{bmatrix} \in \Re^{P\times K}. \tag{3.17}$$

This would entail writing eq. (6.13) in terms of its matrix counterpart as,

$$\begin{bmatrix}\mathbf{X}_1\\ \mathbf{Y_0}\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\begin{bmatrix}\mathbf{X_0}\\ \mathbf{U_0}\end{bmatrix}. \tag{3.18}$$

Similar to DMD, the state space matrices can be written explicitly by computing the Moore-Penrose inverse of the matrix $\begin{bmatrix}\mathbf{X_0}\\ \mathbf{U_0}\end{bmatrix}$ as follows,

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix} = \begin{bmatrix}\mathbf{X_1}\\ \mathbf{Y_0}\end{bmatrix}\begin{bmatrix}\mathbf{X_0}\\ \mathbf{U_0}\end{bmatrix}^{\dagger}. \tag{3.19}$$

For computations, assumptions involved and regularizations, interested readers are directed

to [21, 37].

For underlying systems with nonlinear dynamics, the estimated linear approximation may not provide a good approximation. In that case, it is possible to create an auxiliary state consisting of nonlinear functions of the original state measurement, $g = (g_1, ..., g_D)$, where D is generally larger than the dimension of the original state N [92, 93]. One can then proceed to computing the DMD on the new auxiliary state snapshot matrices as,

$$\mathbf{g}(\mathbf{X}_1) = \begin{pmatrix} | & | & & | \\ \mathbf{g}(\mathbf{x}_1) & \mathbf{g}(\mathbf{x}_2) & \cdots & \mathbf{g}(\mathbf{x}_{M-1}) \\ | & | & & | \end{pmatrix} \; ; \quad \mathbf{g}(\mathbf{X}_2) = \begin{pmatrix} | & | & & | \\ \mathbf{g}(\mathbf{x}_2) & \mathbf{g}(\mathbf{x}_3) & \cdots & \mathbf{g}(\mathbf{x}_M) \\ | & | & & | \end{pmatrix}. \tag{3.20}$$

This procedure is known as the Extended Dynamic Mode Decompostion (EDMD) [92]. The classic DMD is a special case of EDMD in the case where the measurement consists of the identity g(x) = x. The authors' recently developed WDMD which is also a special case of EDMD wherein the measurement consists of the wavelet coefficients of the state vector. The authors applied WDMD for dealing with partial state information and reported results [21]. One major limitation of the DMD algorithm is that it requires knowledge of the full underlying state space variables that govern the behavior of the system of interest. In many real-world applications, some governing variables may be completely unobserved or combined into mixed observations. For example, in the present case, we only observe the tracked order vibration and gas-path performance variables which can be thought of as measurements taken from the complex dynamical system whose internal state-variables are unknown.

Classically, researchers have taken inspiration from Taken's embedding theorem for dealing with partial state information [94, 95, 96]. Delay embedding augments the information contained in the system state by supplementing it with measurements of the state history. Recently, DMD on delay coordinates has found application for low dimensional dynamical system connected to these linearizing coordinate systems in the Hankel alternative view of Koopman (HAVOK) approach. This method augments a state represented by a single (or a few) measurement function with its future measurement, resulting in a new observable,

$$\tilde{g}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}(t+\Delta t) \\ \mathbf{x}(t+2\Delta t) \\ . \\ \mathbf{x}(t+n\Delta t) \end{bmatrix}, \tag{3.21}$$

where $\tilde{g}(\mathbf{x}(t)) \in \Re^{nd\times 1}$ is the delay embedded version of the state vector $\mathbf{x}(t)$. This entails

writing eq. (3.20) as,

$$\mathbf{g}(\mathbf{X}_1) = \mathbf{H} = \begin{bmatrix} \mathbf{x}(t_1) & \mathbf{x}(t_2) & . & . & \mathbf{x}(t_{m-1}) \\ \mathbf{x}(t_2) & \mathbf{x}(t_3) & . & . & . \\ . & . & . & . & . \\ . & . & . & . & \\ \mathbf{x}(t_n) & \mathbf{x}(t_{n+1}) & . & . & \mathbf{x}(t_{m+n-1}) \end{bmatrix}, \tag{3.22}$$

$$\tilde{\mathbf{g}}(\mathbf{X}_2) = \mathbf{H}' = \begin{bmatrix} \mathbf{x}(t_2) & \mathbf{x}(t_3) & . & . & \mathbf{x}(t_m) \\ \mathbf{x}(t_3) & \mathbf{x}(t_4) & . & . & \mathbf{x}(t_{m+1}) \\ . & . & . & . & . \\ . & . & . & . & \\ \mathbf{x}(t_{n+1}) & \mathbf{x}(t_{n+2}) & . & . & \mathbf{x}(t_{m+n}) \end{bmatrix}.$$

where $\mathbf{g}(\mathbf{X}_1) \in \Re^{nd \times K-n}$, and $\mathbf{g}(\mathbf{X}_2) \in \Re^{nd \times K-n}$ represents the Hankel matrix and the time shifted Hankel matrices respectively. Thus, the original ioDMD solution in eq. (6.14) can be represented in the form of Hankel matrices as,

$$\begin{bmatrix} \mathbf{H}' \\ \mathbf{Y_0} \end{bmatrix} \approx \begin{bmatrix} \mathbf{A}_d & \mathbf{B}_d \\ \mathbf{C}_d & \mathbf{D}_d \end{bmatrix} \begin{bmatrix} \mathbf{H} \\ \mathbf{U_0} \end{bmatrix}, \tag{3.23}$$

where $\mathbf{A}_d \in \Re^{nd \times nd}, \mathbf{B}_d \in \Re^{nd \times M}, \mathbf{C} \in \Re^{P \times nd}, \mathbf{D}_d \in \Re^{P \times M}$, assuming M inputs, N states and P outputs are measured. Similar to the ioDMD algorithm, the optimal state-space matrices are found by solving the following optimization problem,

$$\underset{\begin{bmatrix} \tilde{\mathbf{A}}_d & \tilde{\mathbf{B}}_d \\ \tilde{\mathbf{C}}_d & \tilde{\mathbf{D}}_d \end{bmatrix}}{\arg\min} \left(\left\| \begin{bmatrix} \mathbf{H}' \\ \mathbf{Y_0} \end{bmatrix} - \begin{bmatrix} \tilde{\mathbf{A}}_d & \tilde{\mathbf{B}}_d \\ \tilde{\mathbf{C}}_d & \tilde{\mathbf{D}}_d \end{bmatrix} \begin{bmatrix} \mathbf{H} \\ \mathbf{U_0} \end{bmatrix} \right\|_{\mathrm{F}} \right). \tag{3.24}$$

The least-squares solution to this optimization problem is given as,

$$\begin{bmatrix} \mathbf{A}_d & \mathbf{B}_d \\ \mathbf{C}_d & \mathbf{D}_d \end{bmatrix} = \begin{bmatrix} \mathbf{H}' \\ \mathbf{Y_0} \end{bmatrix} \begin{bmatrix} \mathbf{H} \\ \mathbf{U_0} \end{bmatrix}^{\dagger}, \tag{3.25}$$

where † denotes the Moore-Penrose pseudo-inverse.

3.2.4 Lasso regression

Supervised machine learning in the context of dynamical system can be defined as the means of modeling the input-output relations of the system using conventional supervised methods such as Support Vector Machines, linear regression, neural networks. The complexity of modeling a dynamical system stems from the time element (sequential data) associated with

the measurements/states of the system. This temporal component, which is omnipresent in almost all the dynamical system, makes the whole problem challenging.

In the present section, a brief background about Lasso regression is presented. In regression, the goal is to find relationships between the descriptive (independent) variables $x_1, x_2, ..., x_k$ and the described (dependent) variable y. In linear regression analysis, the general formula of the regression equation that gives the shape of the relationship between variables can be expressed as,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_k x_k + \epsilon, \tag{3.26}$$

where β_i, $i = 0, 1, 2.., k$ represents the regression coefficients and ϵ represents the normally distributed error. One of the major issues with linear regression is that the model is highly sensitive to the predictor variables, irrespective of any physical meaningful intuition. This can lead to overfitting if not properly cross-validated.

The problem of overfitting and noisy data in linear regression can be solved to an extend by including a regularization parameter γ which prevents the regression parameters from taking large values.,

$$\min_{\tilde{\beta}_0, \tilde{\beta}} \left(\frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \tilde{\beta}_0 - x_i^T \tilde{\beta} \right)^2 + \lambda \sum_{j=1}^{p} \left| \tilde{\beta}_j \right| \right), \tag{3.27}$$

where N represents the total number of samples, and λ represents the regularization parameter. While large value of λ parameter promotes sparsity in the model, smaller value of λ promotes higher variance. In practice, λ is tuned using a procedure known as cross-validation. In the present study, we utilize the same procedure to find the optimal value of λ that results in a model with the lowest testing error.

3.3 Operational conditions and data set

In the present study pass-off test data from a commercial turbofan engine is considered. The turbofan engine is composed of a three independent coaxial shafts with a single annular combustor. The Low Pressure (LP) compressor or the outer fan is driven by a 6 stage axial turbines. Meanwhile the Intermediate Pressure (IP) shaft, driven by a single stage turbine, powers the 8 stage axial compressor. The six-stage High Pressure (HP) Compressor is driven by a single stage turbine, turning in the opposite direction of the two others shafts [97]. We restrict our focus towards order corresponding to the second stage of the turbofan engine under study. The others stages will be considered in a future study.

All turbofan engines undergo a series of pass-off tests before entering into service, during which sensors onboard the engine collect critical information about the performance and safety of the engine. The data recorded by these sensors provide a suitable representative data set to carry out the present study. During a typical pass-off test, the engine under test passes through a series of operating speeds, thereby exciting various engine parts.

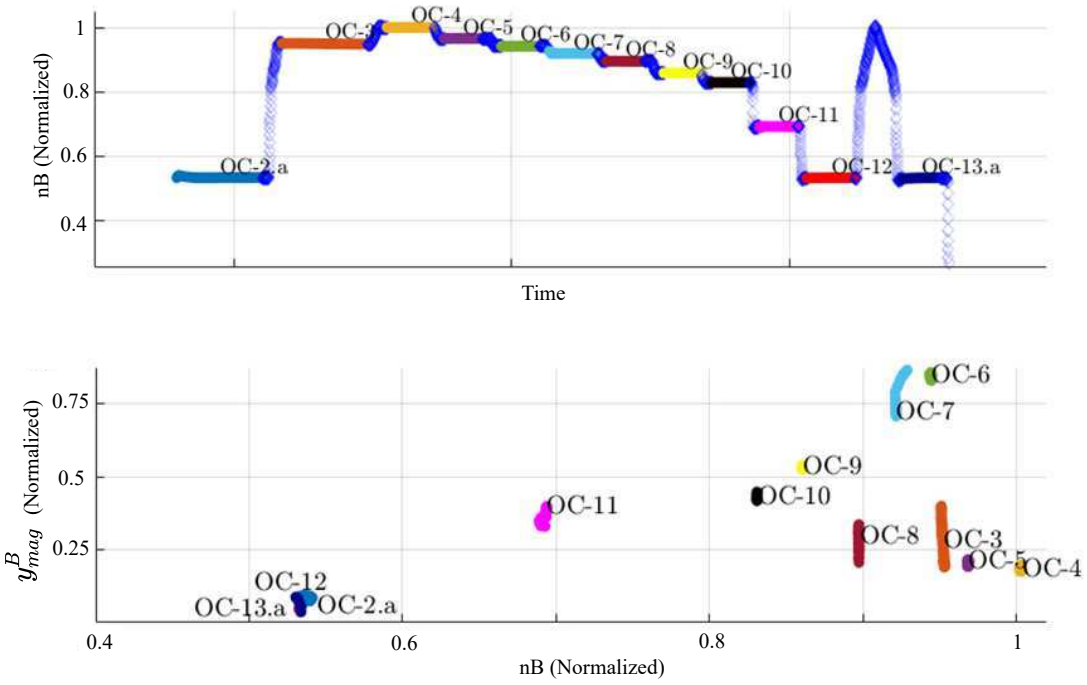


Figure 3.3: Pass-off test curves with operating conditions labelled

Accelerometers attached to the intercase panel record these vibrations in both principal directions, and the Engine Monitoring Unit (EMU) converts these raw acceleration signals into tracked order vibration signals. If the pass-off vibration data exceeds the industry standards, multiple test runs are conducted to zero in on the cause of higher vibrations. This process is time-consuming and results in higher testbed occupancy, thereby translating to loss of revenue. The absence of an accurate and deployable physics-based model for the tracked order vibration causes ambiguity in decision making.

Figure 3.3 illustrates the various operational speeds involved in a typical pass-off test. In the present study, we have normalized the sensor data to a fixed scale and masked several variables to protect the confidentiality of the data. We want to reiterate that the absolute values are inconsequential from a scientific point of view, and hence, most of the scales in the present study are relative. We divide the pass-off test curves based on their operational shaft speed into 13 distinctive regions for analysis purposes, as shown in Figure 3.3. In later sections, the switching linear dynamical system (SLDS) model is based on these divisions known as Operational Conditions (OC). The transition between one OC to another OC is defined as the chirp rate and is identified in the literature as a critical parameter affecting the vibration.

The data set recorded during a typical pass-off test consists of 22 variables encompassing temperatures, pressures, and other control-related data recorded at various locations inside

Sensor type	Sensor variable
Time	time
Shaft speed	nA, nB, nC
Tracked order vibration	y^A_{mag}, y^B_{mag}, y^C_{mag}
Gas-path temperature	tA, tB, tC, tD, tE, tF, tG, tH, tI
Gas-path pressure	pA, pB, pC, pD
Miscellaneous	M1, M2

Table 3.1: Pass-off curve variables and sensor types used in the present study

the engine. Readers are directed to Table 3.1 for a summary of different variables included in the data set. The data is sampled at a frequency resolution of $40Hz$, with discontinuity at engine shutdown.

3.4 Analysis of tracked order vibration

The tracked order vibration is primarily a function of the excitation forces and the system properties [7]. The dynamic operating conditions, particularly the variation in temperature of the structural components, are a critical parameter affecting the mechanical properties of the engine. It is pertinent to note that these structural temperatures cannot be measured directly owing to hardware limitations. However, they can be inferred using the available gas-path temperature variables such as tA, tF, and tI. These variables represent the hot air temperature recorded at critical points inside the engine and are often used by practitioners to approximate the thermal state of the metal/disc temperature.

From Figure 3.3, we observe significant dispersion in the tracked order vibration amplitude at almost all OCs, albeit the engine operating at a 'steady' operating speed. This variability in tracked order vibration curve within an OC is attributed to the differential thermal time constants, whose severity varies for individual OC's. From Figure 3.3, the highest variability in TO vibration within an OC is observed in $OC3$. This variability could be attributed to the position of OC3 in the speed profile, which follows the idle. The hyperplane illustrating the evolution of three temperature variables in $OC3$ is as shown in Figure 3.4 a. The temperature variable, tA, is a controlled parameter and hence has a faster response time compared with tI and tF as seen from Figure 3.4 a. This distinction is essential to identify, as the observed tracked order vibration is a function of both the system characteristics and the input excitation from the system operations. Subsequently, this interplay between the controlled temperature variable (tA) and the thermal state of the engine structure, as inferred from tF and tI, is critical in capturing the characteristics of tracked order vibration in a particular OC. Therefore for better understanding, we employ a clustering algorithm to differentiate the regions inside the hyperspace in each of the operating conditions.

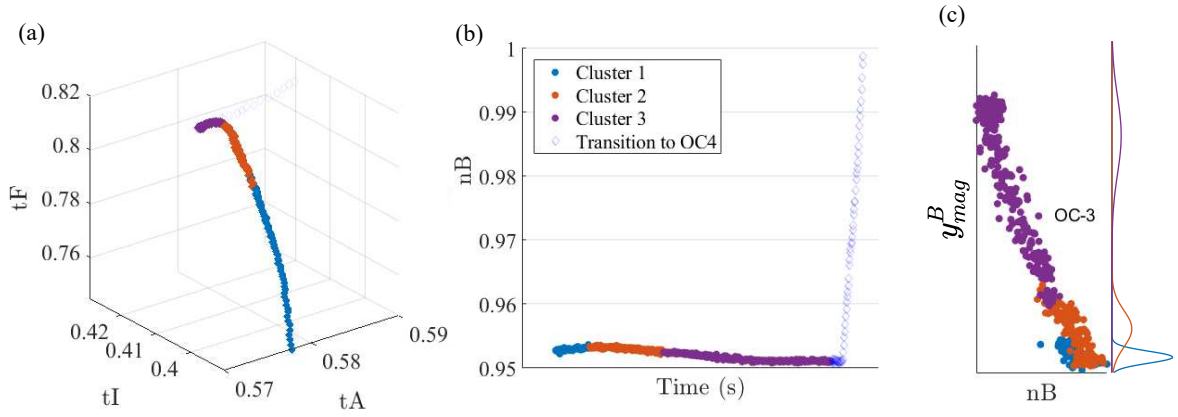


Figure 3.4: Results of GMM on $OC3$: a.) GMM clusters (as shown in blue, orange , and magenta) plotted on the hyperplane on which temperature variables are evolving, (b) carrying forward the cluster assignment onto the shaft speed profile, and (c) cluster assignments from (a) carried forward to the characteristic curve to contextualize the tracked order vibration in each clusters.

We utilize Gaussian mixture models (GMM) to classify the above-mentioned hyperspace. GMM is an unsupervised algorithm for distinguishing between normally distributed subpopulations within an overall population [98]. Mixture models do not require knowledge of which subpopulation a data point belongs to, allowing the model to automatically learn the subpopulations. Since subpopulation assignment is not known apriori, this constitutes a form of unsupervised learning. As the name suggests, GMM tries to fit the data using several Gaussian densities. A gaussian density in d-dimensional space is defined as,

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}(|\boldsymbol{\Sigma}|)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{3.28}$$

where $\mathbf{x} \in \Re^{d\times 1}$ represents the data vector, $\boldsymbol{\mu} \in \Re^{d\times 1}$ is the mean of a Gaussian density and $\boldsymbol{\Sigma} \in \Re^{d\times d}$ is the covariance matrix of the Gaussian density. In GMM, a Gaussian density $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is called a component. Therefore, we can define a K component GMM as,

$$f_K(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{3.29}$$

where w_k is the mixing weight of the k^{th} component, with weights of all components summing up to one. According to eq. (3.29) if the mean, $\boldsymbol{\mu}_k$, the covariance matrix $\boldsymbol{\Sigma}_k$, and the mixing weights w_k of the components are known, then one can easily model the distribution of a data set using GMM. However, w_k, μ_k, Σ_k are unknown in most cases and should be estimated using data learning methods. There exists various classes of methods such as Expectation-

Maximization [82], Bayesian methods [99], and Graphical methods [16] for estimating the the parameters of mixture models. In the present case, we employ dirichlet process GMM toolbox [100] to estimate the parameters of the GMM model. For a complete exhaustive review of the methods and their implementation, interested readers are directed to [98].

We now apply GMM methodology to cluster the hyperplane in which the profiles of tA, tI, and tf evolve. We show in Figure 3.4 (a) the results of applying the GMM methodology to the three temperature profiles from $OC3$. The GMM algorithm divides the hyperplane into three normally distributed subpopulations. Figure 3.4 (b) maps the cluster assignment results into the shaft speed profile. Interestingly, the cluster behavior is sequential, i.e., it follows one after the other. This behavior is expected as there are underlying dynamics behind their evolution, and GMMs successfully cluster them based on their dynamics. Furthermore, these cluster assignments can be carried forward to the characteristic vibration curve and the corresponding cluster assignments markered as shown in Figure 3.4 (c). We observe that the mean amplitude of the tracked order vibration in each of these clusters demonstrates variability. Hence, the variability observed in the characteristic curve in Figure 3.3 (b) for $OC3$ can be partially explained through the present analysis.

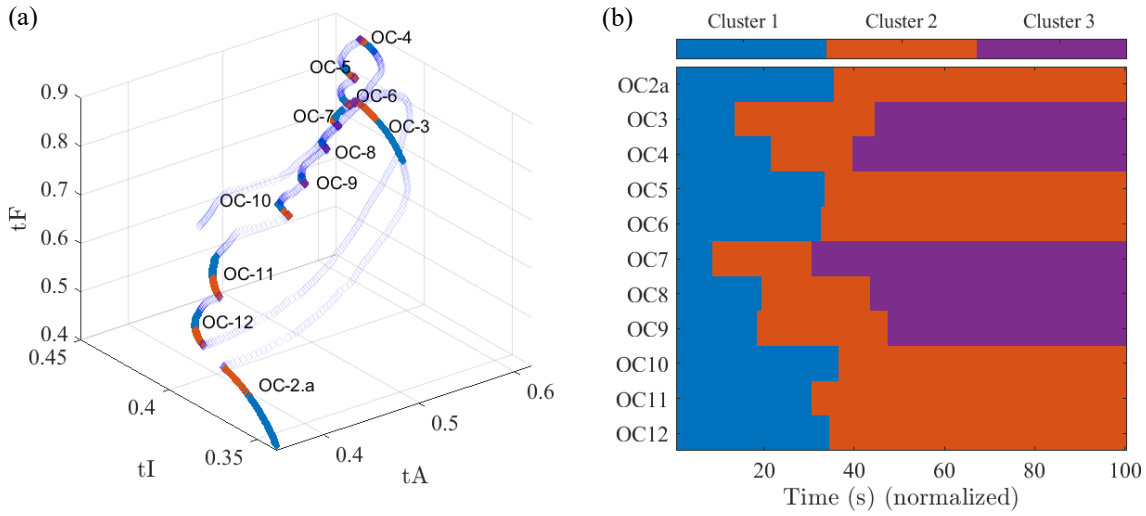


Figure 3.5: (a) Results of applying GMM clusters seperately on all operational conditions from $OC2a$ to $OC12$, and (b) Cluster evolution in time for each OCs.

We repeat the above-mentioned procedure in all the OCs, and results are shown in Figure 3.5. The transparent blue dots in Figure 3.5 (a) corresponds to the transition between each OCs and are neglected in the present analysis. Interestingly, the evolution of the tA, tI, and tF lies in an inclined 2D plane within the hyperplane, thereby demonstrating the controlled nature of the underlying dynamical system. Out of the 11 OC under study, we observe three clusters in five of OCs and two clusters in others. The time axis has been normalized in Figure 3.5 (b) to have a constant reference line, even though the engine's total test time in

each of the *OC*'s is highly variable and differs from run to run. Furthermore, each cluster's time signatures give us a unique platform to evaluate tracked order vibration characteristics in each cluster, as shown in the next section.

3.4.1 Bayesian Gibbs sampling for tracked order vibration mean and dispersion evaluation

In the previous section, GMM clustered the *OCs* into various zones depending on the evolution of the thermals. It was also observed that the system passes through each of the clusters sequentially as shown in Figure 3.4 (b). Thus points appearing in each cluster can be labelled accordingly, and are transferred directly into the n2 shaft speed profile and to the y^B_{mag} tracked order vibration, as seen from Figure 3.4 (c). Hence, for each cluster, there is an associated y^B_{mag} mean as well as standard deviation. There exists an array of methods for estimating the mean and standard deviation within a cluster and in the present work, we use the popular Gibbs sampling methodology [79, 101], which is based on the Bayesian estimating theory. The Bayesian theory provides a framework to evaluate the posterior distribution of the parameter rather than point estimates [16]. Also, bayesian framework can incorporate outliers present in the data by adding in the assumption of data being t-distributed.

The Gibbs sampling is a special case of the Markov Chain Monte Carlo method. MCMC is a statistical sampling procedure to sample from multivariate densities by conditionally sampling from their univariate densities. The basic MCMC procedure provides a framework to: (a) sample from a multivariate posterior distribution, and (b) move through the entire region of support of a posterior distribution. Interested readers are directed to [16, 79, 102] details regarding Gibbs sampling and MCMC procedure .

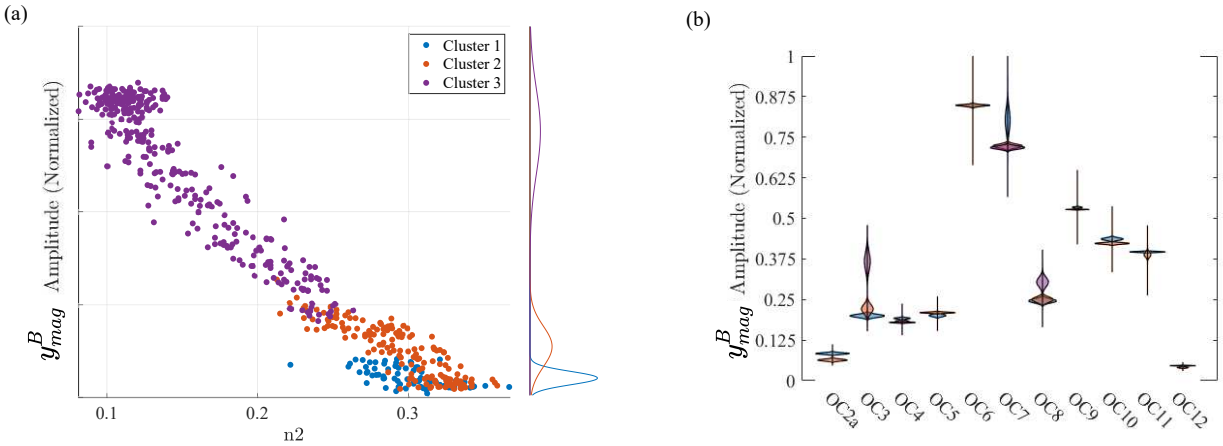


Figure 3.6: (a) Results of applying gibbs sampling on the clustered tracked order vibration as seen from the normal plots (b) Violin plot of y^B_{mag} distributions within each OC.

In the present work, Gibbs sampling is utilized to determine the mean and standard deviation of the tracked order vibration within each cluster. Hence, the parameters to be learned can be represented as $\theta = (\mu, \sigma)$. Outliers within each clusters are handled using the Bayesian framework by building in the gamma outliers assumption. For full derivation of conditionals of μ and σ, please refer to the appendix. As an example, consider $OC3$ in Figure 3.6(a), where the GMM on the temperature hyperplane results in three clusters. The labels when translated to the characteristics curve encompass a spread centered around a mean for all the three clusters. The posterior distribution of mean and standard deviation is evaluated through Gibbs sampling and the results are as shown in Figure 3.6 (a), with all distributions plotted beside the characteristic curve. From Figure 3.6 (a), we observe that the mean of cluster 3 TO vibration is relatively larger compared to cluster 2 and cluster 1 mean's. It is also pertinent to note that standard deviation is different across clusters, partially explaining the dispersion effect seen in characteristic curves.

We repeat the above mentioned procedure for all other OCs, and we plot the y^B_{mag} distributions observed across all clusters within individual OCs in the form of a violin plot in Figure 3.6 (b). We observe interesting patterns from Figure 3.6 (b), in terms of the relative difference in mean of y^B_{mag} within individual OCs. Take for example $OC7$, wherein the total data is comprised of 3 clusters as shown in Figure 3.5 (b). From Figure 3.6 (b) corresponding to $OC7$ violin plot, we observe that the mean of y^B_{mag} in cluster 1 is larger than cluster 2 and cluster 3 mean. This effect is reversed for $OC3$ and $OC8$, wherein the cluster 3 mean is larger compared to cluster 2 and cluster 1 mean. Apart from the OCs mentioned before, the violin plots in Figure 3.6 (b) does not suggest significant variablity within other OCs.

3.4.2 Utilizing the MS-DPCCA for feature selection

In the present section, we analyze the data from seemingly unrelated instrumentation such as temperature, pressure, and shaft speed and compute the degree of correlation with the measured tracked order vibration. We carry out the cross-correlation analysis using the MS-DPCCA procedure, which can track the correlation across multiple time scales, thereby providing a means towards identifying complex interdependencies. In the present work, MSDPCCA is utilized to visualize the complex interdependencies among multiple variables and as a tool for variable selection. Thus, MS-DPCCA can be employed in conjunction with a data-driven modeling approach to build the model for tracked order vibration.

As explained before, the experimental dataset used for the present work consists of data collected from 22 different sensors. These include instrumentation such as temperature, pressure, vibration, and speed transducer, all sampled at the same frequency and operating under nominal conditions. The experimental data set involves five consecutive engine pass-off tests, where the acquisition hardware samples data from various instrumentation across multiple station points inside the engine. This section investigates the de-trended partial cross-correlation between various gas path variables and IP tracked order vibration, which

will be referred to as the y^B_{mag}, in the remainder of the report.

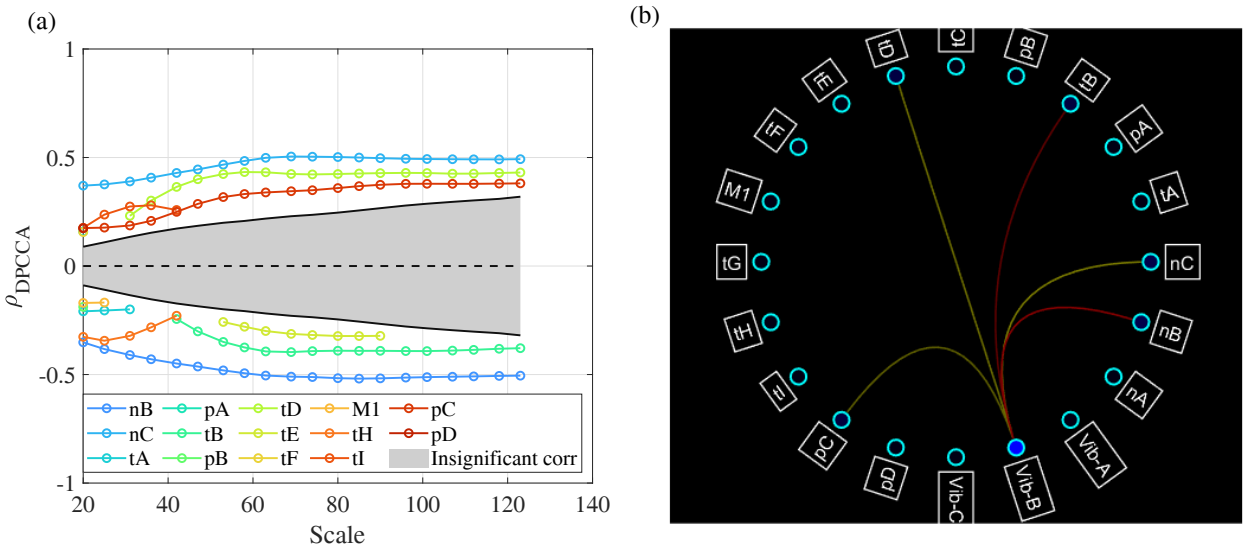


Figure 3.7: Results of applying MS-DPCCA procedure: (a) Evolution of ρ_{DPCCA} between tracked order vibration and other variables specified in table 3.1 for $OC7$, and (b) inter dependency circle showing the variables of importance for modeling y^B_{mag} in $OC7$.

In applying the MS-DPCCA procedure, careful consideration must be given to find only the statistically significant correlations. Towards the same, we determine the 95% significance threshold by applying the Monte Carlo test. We shuffle the time series data and repeat the DPCCA procedure at each scale 10000 times to determine 95% significance threshold [103]. The detrended partial cross-correlation coefficient for each of the sensors mentioned in table 3.1 with y^B_{mag} is computed across 20 scales ranging from 1 to one-fourth length of samples in each operating window.

The analysis results are as depicted in Figure 3.7 for y^B_{mag} corresponding to $OC7$. The shaded area in Figure 3.7 (a) represents the insignificant correlation zone computed using the 95% significance test. The ρ_{DPCCA} clearly shows that corresponding to OC7, y^B_{mag} is correlated significantly to numerous variables at multiple time scales, highly significant amongst which are nB and nC. Since we are interested in cross-correlations at higher time scales, a separate plot consisting of significantly cross-correlated variables is constructed corresponding to the largest time scale (i.e., s = 120) and is shown in Figure 3.7 (b). While the variables nC, tD, and pC are positively correlated with y^B_{mag} at the largest scales, variables tB and nB are negatively correlated to y^B_{mag} as seen from Figure 3.7 (b). Although other variables such as tF and tI are correlated to y^B_{mag} in shorter time scales, for the purpose of building a data-driven model, we select only those variables that are correlated both at larger and smaller time scales.

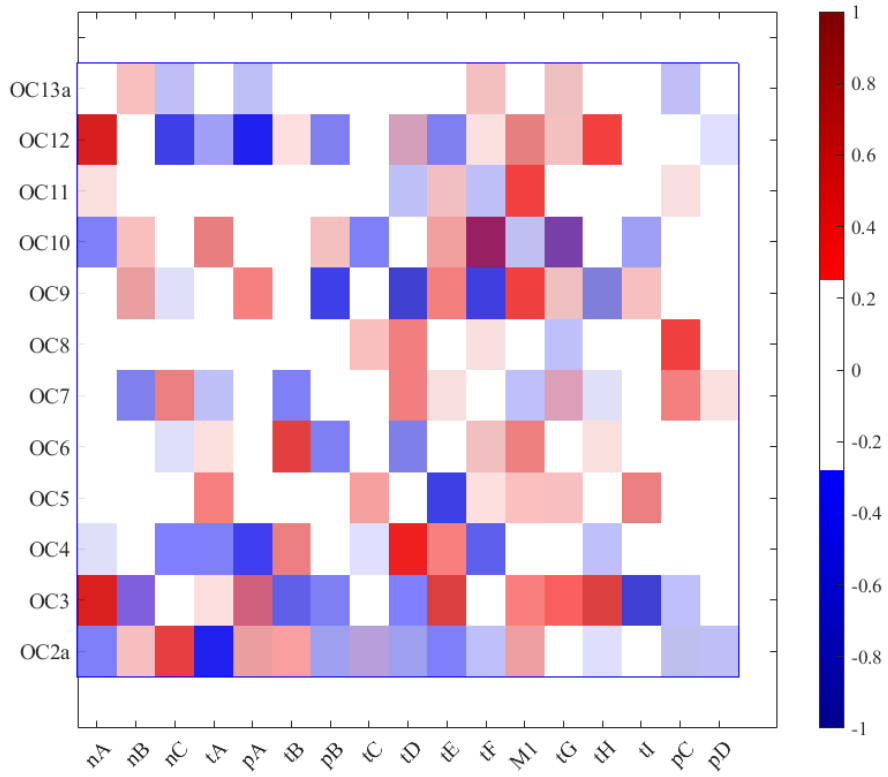


Figure 3.8: ρ_{DPCCA} heatmap for all operating condition for different runs of the engine for y^B_{mag}

We repeat the above mentioned procedure seperately for each of the operating conditions from $OC2a$ to $OC13$ and compute the ρ_{DPCCA} coefficient. It is pertinent to note that across different operating conditions, the correlation structure changes, further necessitating the need for a SLDS model for the tracked order vibration. Finally, this procedure can be repeated across all the different engine pass-off test data and ρ_{DPCCA} values plotted in the form of a heatmap matrix as shown in Figure 3.8. For demonstration, we have masked very low correlations and plotted only $|\rho_{DPCCA}|$ above 0.2. In the present study, the heatmap chart in Figure 3.8 is viewed through the angle of feature selection for building a data-driven model. Thus, to build the data-driven model for y^B_{mag} in the next section, we select the top 4 variables, displaying the highest correlation in each of the OCs.

3.5 Model building

In this section, two different methods of model building, utilizing the Lasso regression and Delay-DMD, are explored. As explained in section 3.3, the engine under test is operated

through a series of speeds (rpm), and tracked order vibration signals are recorded alongside other instrumentation parameters such as temperature and pressure across the gas path. Previous studies had reported the evidence of existing nonlinear vibrations in aircraft engines [97]. Hence in the present study, we use the data from pass-off test curves corresponding to 12 different operating conditions and construct a switching linear dynamical system (SLDS) [104] comprised of 12 linear models where the switch is based on the discrete operational condition classification as shown below,

$$z(t_k) \sim f(z(t_{k-1}), nB(t_k)), \tag{3.30}$$

$$\begin{cases} \mathbf{x}(t_{k+1}) = \mathbf{A}_{z(t_k)}\mathbf{x}(t_k) + \mathbf{B}_{z(t_k)}\mathbf{u}(t_k);\ y(t_k) = \mathbf{C}_{z(t_k)}\mathbf{x}(t_k) + \mathbf{D}_{z(t_k)}\mathbf{u}(t_k), & \textit{Delay DMD} \\ y(t_k) = \boldsymbol{\beta}_{z(t_k)}\mathbf{u}(t_k), & \textit{Lasso regression} \end{cases}$$

where $\mathbf{A}_\zeta, \mathbf{B}_\zeta, \mathbf{C}_\zeta, \mathbf{D}_\zeta$ are the Delay DMD system matrices associated with discrete state $z(t_k) = \zeta$. Similarly $\boldsymbol{\beta}_\zeta$ is the Lasso regression coefficient vector associated with discrete state $z(t_k) = \zeta$. The data-driven models are then used to simulate the tracked order vibration of the structure to a given set of input excitations, and the quality of the fit is estimated. In the present work, the following relative error metric is defined to evaluate the quality of the fit,

$$\epsilon_{td}^{rel} = \sqrt{\frac{\sum_{i=1}^{K} ||y(t_i) - \tilde{y}(t_i)||_2^2}{\sum_{i=1}^{K} ||y(t_i)||_2^2}}, \tag{3.31}$$

where K is the number of time points and ϵ_{td}^{rel} measures the relative error in time domain between the measured responses $y(t)$ and the predicted responses $\tilde{y}(t)$.

The quality of the data-driven model produced by both methods depends on hyper-parameters, which are often problem-specific. In the present study, we utilize the standard cross-validation methodology to choose the hyper-parameters efficiently. Cross-validation is a resampling procedure used to evaluate data-driven models on a limited data sample. This procedure entails iterating over different models, each with a specific hyper-parameter value over different sets of training and testing data to select the hyper-parameter value, resulting in the lowest cross-validated testing error. In the present study, we utilize dataset from four pass-off test runs of the alpha engine, and hence we adopt a 4 fold cross-validation testing strategy for the dual objectives of : (i) selecting the optimal hyper-parameters, and (ii) to evaluate the quality of fit using eq. (6.22). The following subsections demonstrate the model building procedure and testing using Lasso regression and Delay-DMD. Towards the end, we present a rigorous comparative study analyzing each method's pros and cons concerning developing a data-driven model for engine vibrations.

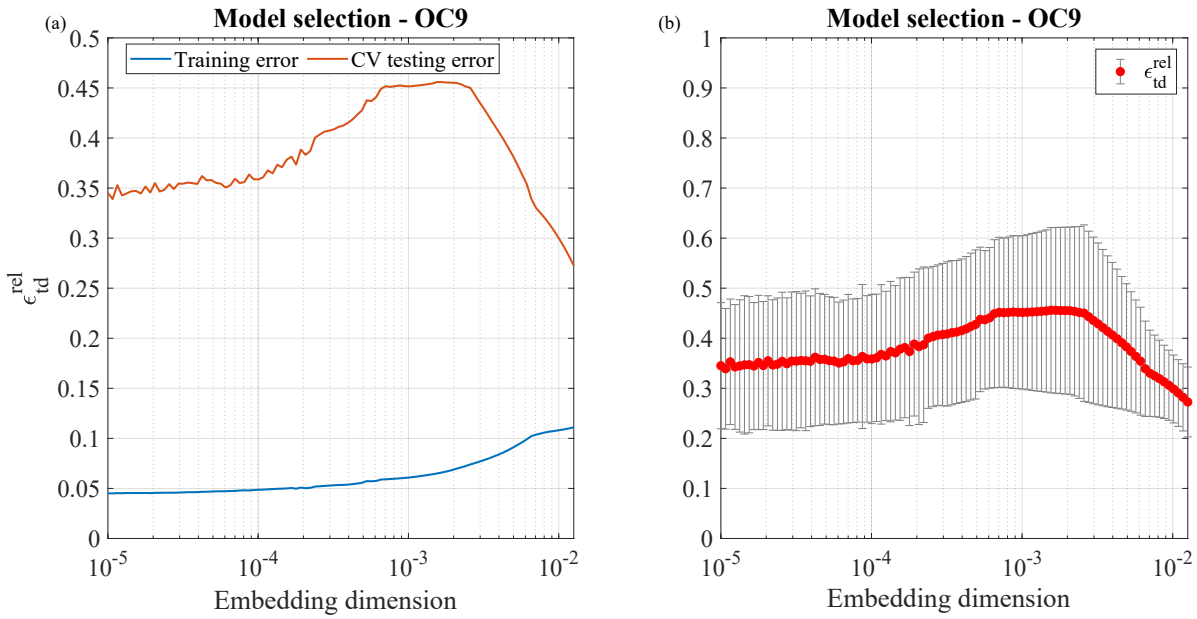


Figure 3.9: Cross validation - Lasso

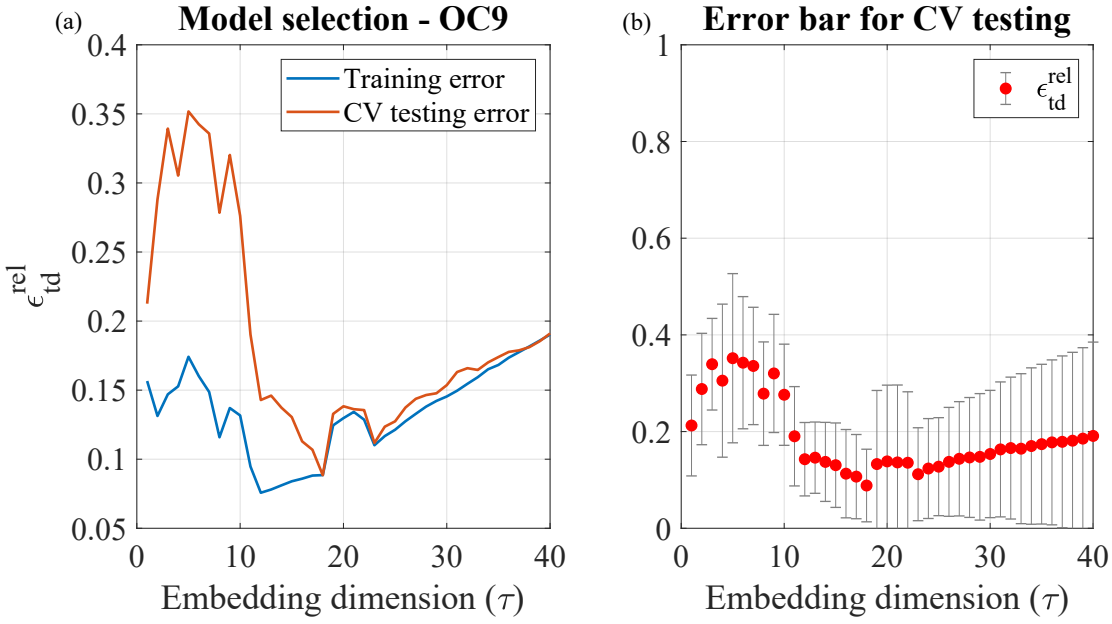


Figure 3.10: Cross validation - Delay-DMD

3.5.1 Data-driven modeling using Lasso regression

As discussed in Section 3.2.4, Lasso regression is a regularized regression method that automatically selects a subset of input features by promoting sparsity in the coefficient vector $\boldsymbol{\beta}_\zeta$. The automatic selection of input features entails circumventing the MS-DPCCA procedure and letting the Lasso regression handle the feature selection based on the lowest CV testing error criteria. As explained before, we employ a four-fold cross-validation strategy and utilize the average ϵ_{td}^{rel} for selecting the regularization coefficient (λ). While large values of λ drives the individual elements of $\boldsymbol{\beta}_\zeta$ to zero, thereby promoting sparsity, smaller values of λ have little or no effect on $\boldsymbol{\beta}_\zeta$. In other words, smaller values of λ result in a model with a higher variance and lower bias, and larger values of λ result in models with higher bias and lower variance. Therefore it is critical to find the optimal value of λ to avoid over-fitting and under-fitting.

Figure 3.9(a) shows the mean value of the cross-validated training and testing error plotted against the value of λ for OC9. While the left half side of the plot with lower training and larger testing results represent the high variance low bias zone, the values of λ larger than 10^{-2} result in low variance high bias model (underfit). Figure 3.9(b) shows the mean and standard deviation of the 4-fold cross-validated testing error. The standard deviation of the model error resulting from 4 testing data sets is uniform across the range of λ. From Figure 3.9(a) and Figure 3.9(b), we select the value of λ that results in the lowest CV testing error, which in the present case is 0.021. The average training and testing error corresponding to $\lambda = 0.021$ from the CV procedure is reported as $\epsilon_{td}^{rel} = 0.11$ and $\epsilon_{td}^{rel} = 0.27$, respectively. This CV procedure ensures that we select λ in such a way that we avoid overfitting as well as underfitting, resulting in the following model for the y_{mag}^B corresponding to $OC9$,

$$y_{Lasso}(t) = \begin{bmatrix} 0.0031, & -4.04e^{-5}, & 0.0204, & -0.0010, & -0.0015, & 0.0033, & -0.0074, & -0.0302 \end{bmatrix} \begin{bmatrix} tA(t) \\ pB(t) \\ tC(t) \\ tE(t) \\ M2(t) \\ tI(t) \\ pC(t) \\ pD(t) \end{bmatrix} \tag{3.32}$$

By setting the value of $\lambda = 0.021$, the Lasso regression automatically selects 8 predictor variables consisting of 4 temperature variables, 3 pressure variables, and throttle position to model y_{mag}^B in $OC9$. Finally, we repeat this procedure for all the OCs, and the quality of the fit analyzed using eq. (6.22) and plotted as shown in Figure 3.12. While the Lasso regression results in good training accuracy in almost all the OCs except $OC2a$ and $OC3$, the testing accuracy is far lower than training accuracy in almost all the OCs. While the testing errors are significant, particularly in $OC2a$,$OC3$, and $OC5$, the errors in $OC6$ to $OC11$ are similar to those of training. Nevertheless, limitations of static methods such as linear regression in modeling the vibration response of a turbofan engine are evident from Figure 3.12.

3.5.2 Data-driven modeling using Delay-DMD

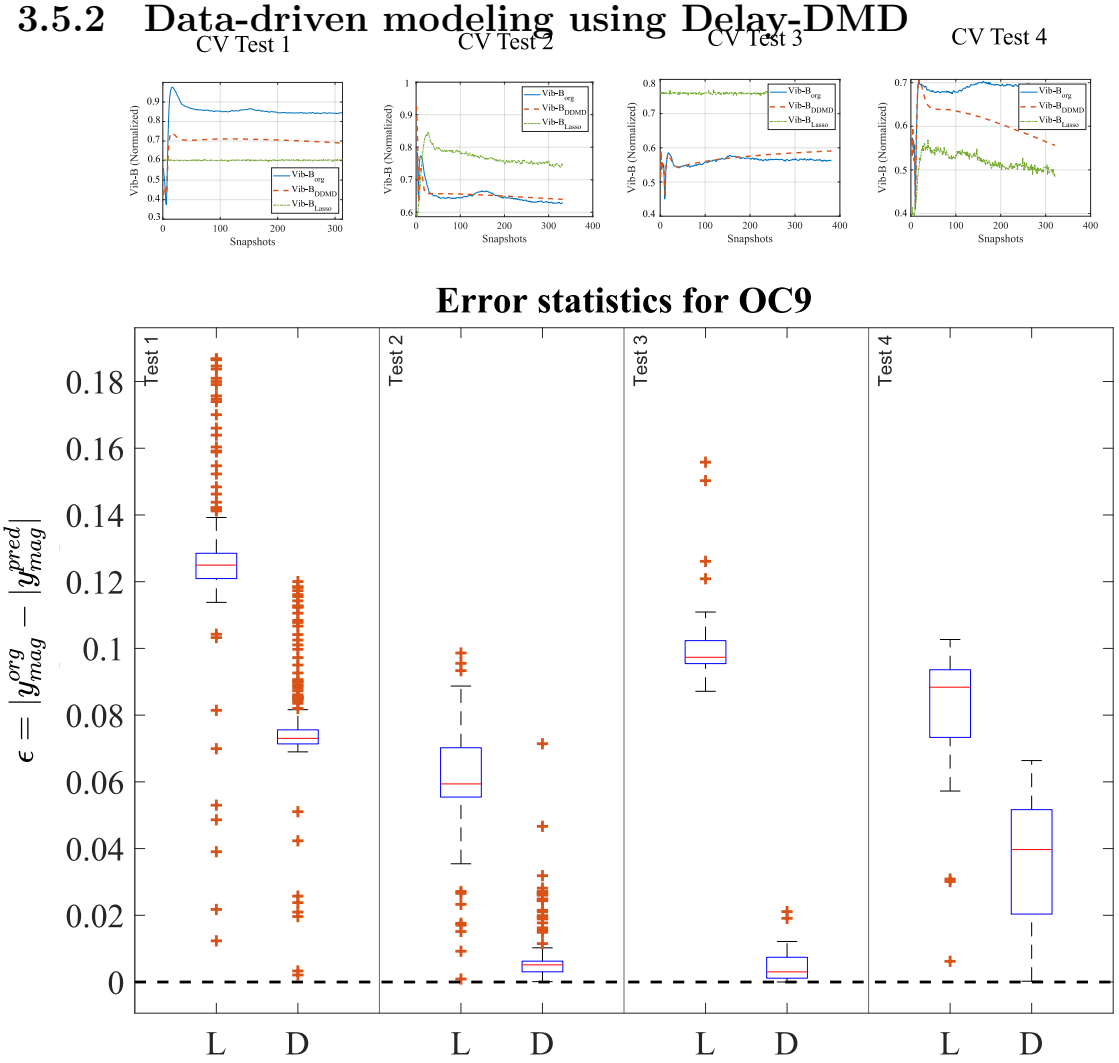


Figure 3.11: Box-whisker plot comparing Lasso regression (L) error distribution with Delay-DMD error (D) in 4 cross-validation testing scenarios, alongside the time-domain prediction.

We utilize the Delay-DMD framework to build a MISO, data-driven, state-space model to predict the HP tracked order vibration for each operating condition of the engine. The Delay-DMD framework utilizes the Hankel matrix of the tracked order vibration measurement and input selected using the MS-DPCCA variable selection to create a discrete state-space model for the tracked order vibration corresponding to an operating condition. At the end of the training phase, Delay-DMD results in 12 separate state-space models, one for each OC, which is tested using a different engine run. From the provided training package, the Delay-DMD algorithm outputs a linear discrete dynamical system of the form valid over a particular

operating condition as shown in eq. (3.30). The state vector's dimension is equal to the embedding dimension in the Hankel matrix and varies per OC. As expected, the optimal embedding dimension is not the same for all the OCs, thus requiring hyper-parameter tuning for each OC similar to Lasso regression.

OC	τ	State Matrices	Training error (ε_{td}^{rel})	Testing error (ε_{td}^{rel})
OC2a	30	$\mathbf{A} \in \Re^{30\times30}$, $\mathbf{B} \in \Re^{30\times4}$, $\mathbf{C} \in \Re^{1\times30}$	0.57	0.57
OC3	40	$\mathbf{A} \in \Re^{40\times40}$, $\mathbf{B} \in \Re^{40\times4}$, $\mathbf{C} \in \Re^{1\times40}$	0.36	0.37
OC4	21	$\mathbf{A} \in \Re^{21\times21}$, $\mathbf{B} \in \Re^{21\times4}$, $\mathbf{C} \in \Re^{1\times21}$	0.13	0.15
OC5	7	$\mathbf{A} \in \Re^{7\times7}$, $\mathbf{B} \in \Re^{7\times4}$, $\mathbf{C} \in \Re^{1\times7}$	0.10	0.12
OC6	17	$\mathbf{A} \in \Re^{17\times17}$, $\mathbf{B} \in \Re^{17\times4}$, $\mathbf{C} \in \Re^{1\times17}$	0.26	0.27
OC7	8	$\mathbf{A} \in \Re^{8\times8}$, $\mathbf{B} \in \Re^{8\times4}$, $\mathbf{C} \in \Re^{1\times8}$	0.16	0.13
OC8	6	$\mathbf{A} \in \Re^{6\times6}$, $\mathbf{B} \in \Re^{6\times4}$, $\mathbf{C} \in \Re^{1\times6}$	0.21	0.33
OC9	18	$\mathbf{A} \in \Re^{18\times18}$, $\mathbf{B} \in \Re^{18\times4}$, $\mathbf{C} \in \Re^{1\times18}$	0.08	0.09
OC10	19	$\mathbf{A} \in \Re^{19\times19}$, $\mathbf{B} \in \Re^{19\times4}$, $\mathbf{C} \in \Re^{1\times19}$	0.05	0.05
OC11	19	$\mathbf{A} \in \Re^{19\times19}$, $\mathbf{B} \in \Re^{19\times4}$, $\mathbf{C} \in \Re^{1\times19}$	0.15	0.19
OC12	4	$\mathbf{A} \in \Re^{4\times4}$, $\mathbf{B} \in \Re^{4\times4}$, $\mathbf{C} \in \Re^{1\times4}$	0.54	0.56

Table 3.2: Table outlining the model specifications

In the present study, 4 fold cross-validation is employed since data exists for 4 different runs of the same engine. We train the Delay-DMD model using 3 engine runs and test on the 4^{th} run. We further repeat this procedure 4 different times, shuffling between the training and testing engine runs and evaluating the quality of the fit using eq. (6.22). For demonstration purposes, we show the results of building a Delay-DMD model in $OC9$. Figure 3.10 (a) shows the average CV testing error (plotted in orange) alongside the training error for the $OC9$ model. As expected, the training error decreases with an increase in τ up to 18. However, further increase in τ seems to have a detrimental impact on the quality of the fit. The average CV error also follows a similar pattern with error continuously decreasing till $\tau = 18$, but there is little or no effect beyond that range. Thus, we select a model order of $18(\tau = 18)$ for the $OC9$ model using CV procedure. The dimensions of the state-space matrices are $\mathbf{A} \in \Re^{18\times18}$, $\mathbf{B} \in \Re^{18\times4}$, $\mathbf{C} \in \Re^{1\times18}$, and $\mathbf{D} \in \Re^{1\times4}$.

The average training and testing error from the CV procedure is reported as $\epsilon_{td}^{rel} = 0.08$ and $\epsilon_{td}^{rel} = 0.085$, respectively. The low value of testing error similar to the training error is in stark contrast with the Lasso regression, which reported a testing error of 2.5 times the training error. Figure 3.11 presents the absolute error between the predicted y_{mag}^{B} and the observed y_{mag}^{B} in the form of series of box-whisker plots. The box-whisker plots show the absolute error distribution in each of the 4 testing cases. In all 4 cases, the mean of the error produced by Delay-DMD is lower than its Lasso counterpart. It is also important to note that the error standard deviations in the Delay-DMD case are smaller than in Lasso regression in three of the four testing cases. The predicted responses from the data-driven

models are overlaid on top of the original responses for all four testing cases and are shown in the plots above the corresponding error distributions. The Delay-DMD predicted responses closely match the measured original response, thereby demonstrating the efficacy of the Delay-DMD.

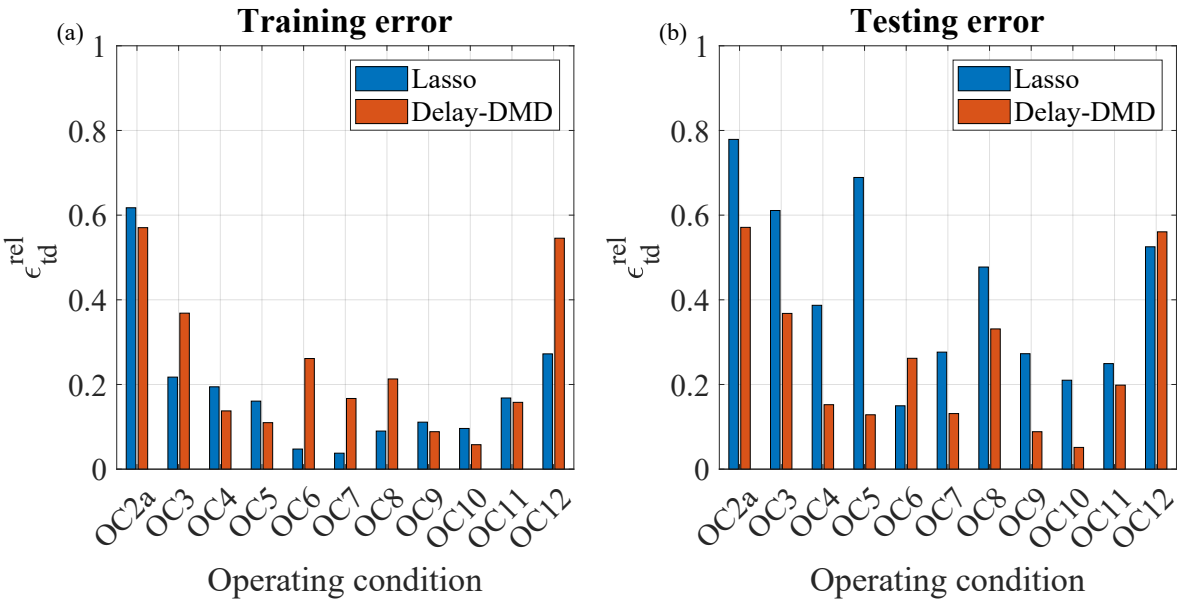


Figure 3.12: (a) Predicted training error (b) Testing error across multiple operating conditions for y_{mag}^{B}.

We repeat the procedure mentioned above for all the OC's and the quality of the training and testing fit assessed using eq. (6.22). This entails finding out the correct embedding dimension for each of the OCs by carrying out the CV procedure. Table 3.2 summarizes the MISO, data-driven, state-space models over different operating conditions, including model order, inputs, and training and testing errors (ϵ_{td}^{rel}). As expected, the model complexity differs for each of the OCs and is highest for OC2a. Figure 3.12 plots the ϵ_{td}^{rel} for the training and testing case across all the OCs and compares the efficacy of both the data-driven methods. Figure 3.12 shows the comparative advantages of Delay DMD over Lasso regression. While the training error is lower for Lasso regression in most OCs, the trend is reversed when the average CV testing error is considered. The Delay-DMD testing error of $\epsilon_{td}^{rel} < 0.20$ across multiple OCs, such as $OC4$, $OC5$, $OC7$, $OC9$, $OC10$, and $OC11$ further highlights the efficacy of the method.

Finally, we excite the SLDS data-driven model using the engine pass-off test data and the predicted tracked order vibration compared with the measured tracked order vibration. Figure 3.13 compares the magnitude of the predicted response from the SLDS with the observed

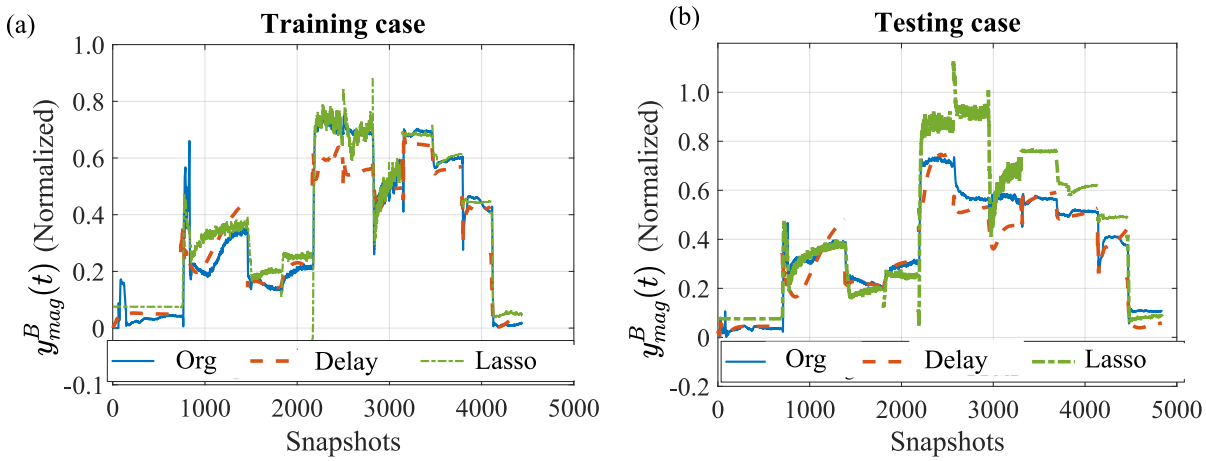


Figure 3.13: Predicted y^B_{mag} across multiple operating conditions overlaid with the measured y^B_{mag} corresponding to : (a) training case, and (b) testing case.

y^B_{mag} across multiple operating conditions. Figure 3.13 also illustrates the instances of the switch based on the rotational speed of the engine, n2. As seen from fig. 3.13, the SLDS comprised of Delay-DMD MISO models, outperforms the SLDS comprised of Lasso models in terms of the quality of the fit.

3.6 Future works and Conclusion

We present preliminary analysis and a data-driven modeling strategy for tracked order vibration in aircraft turbofan engines, valid over multiple operational conditions. We observed the variability and dispersion in tracked order vibration within steady-state operational conditions from the data set collected from a testbed engine. We utilized GMMs to contextualize these dispersions associated with the tracked order vibration in terms of the operational temperatures. GMM analysis revealed various subpopulations within each OC, thereby partially explaining the dispersion of tracked order vibration. By applying the MS-DPCCA approach, we demonstrated the complex interactions and evolution of partial cross-correlation structure over multiple time scales. The ρ_{DPCCA} coefficient values presented in the form of heatmap were subsequently utilized as a threshold for feature selection in building data-driven models.

We first established the data-driven model building using Lasso regression, then followed by the state-of-the-art Delay-DMD. Our current investigation revealed the advantage and efficacy of Delay-DMD over accurately modeling the tracked order vibration in multiple operating conditions of the engine. The low relative testing error of less than $\epsilon^{rel}_{td} = 0.20$ for Delay-DMD in key operational conditions such as $OC4$, $OC5$, $OC7$, and $OC9$, further demonstrated the efficacy of Delay-DMD. The work presented herein demonstrates the fea-

sibility of predicting the tracked order vibration in an aero-engine in a testbed environment. Although Delay-DMD algorithm performed reasonably in most operational conditions, further studies are required to reduce the testing error in OCs such as $OC2a$ and $OC12$. Future studies will also address state variable selection, damping, non-linearity and explore the sensitivity analysis among various parameters.

Acknowledgments

Authors would like to acknowledge the support received through the Rolls-Royce Fellowship and Rolls-Royce Derby, UK for providing the relevant data set for the present work. Authors would also like to thank Mr. Glyn Fox, Rolls-Royce Derby for the helpful suggestions and discussions regarding the engine architecture and data set. Dr. Tarazaga would also like to acknowledge the support received through the John R. Jones III Faculty Fellowship. Mr. Krishnan would also like to acknowledge the support received through the John R. Jones III Graduate Fellowship. The work of Dr. Gugercin was supported in parts by NSF through Grant DMS-1819110 and DMS-1720257.

Chapter 4

A wavelet-based dynamic mode decomposition for modeling mechanical systems from partial observations

Dynamic mode decomposition (DMD) has emerged as a popular data-driven modeling approach to identifying spatio-temporal coherent structures in dynamical systems, owing to its strong relation with the Koopman operator. For dynamical systems with external forcing, the identified model should not only be suitable for a specific forcing function but should generally approximate the input-output behavior of the underlying dynamics. A novel methodology for modeling those classes of dynamical systems is proposed in the present work, using wavelets in conjunction with the input-output dynamic mode decomposition (ioDMD). The wavelet-based dynamic mode decomposition (WDMD) builds on the ioDMD framework without the restrictive assumption of full state measurements. Our non-intrusive approach constructs numerical models directly from trajectories of the full model's inputs and outputs, without requiring the full-model operators. These trajectories are generated by running a simulation of the full model or observing the original dynamical systems' response to inputs. Hence, the present methodology is applicable for dynamical systems whose internal state vector measurements are not available. Instead, data from only a few output locations are only accessible, as often the case in practice. The present methodology's applicability is explained by modeling the input-output response of an Euler-Bernoulli finite element beam model. The WDMD provides a linear state-space representation of the dynamical system using the response measurements and the corresponding input forcing functions. The method is further validated on a real (experimental) data set using modal analysis on a simple free-free beam, demonstrating the efficacy of the proposed methodology as an appropriate candidate for modeling practical dynamical systems despite having no access to internal state measurements.

The contents of this chapter are reproduced from, ***M. Krishnan, S. Gugercin, and P. A. Tarazaga, "A wavelet-based dynamic mode decomposition for modeling mechanical systems from partial observations"****, which is currently under review by Mechanical system and signal processing (MSSP) journal.*

4.1 Introduction

Over the last two decades, data-driven modeling has garnered interest in several research areas, particularly when the involved dynamics are complex and models based on first principles present challenges of varying degrees see, e.g., [20, 22, 23, 105, 106, 107, 108, 109, 110] and the references therein. Moreover, the advances in data processing and sensor capabilities made it much easier to map a system's response to a variety of inputs. In structural dynamics, structures vary in different levels of complexity, and physics-based models may not always be feasible [11, 111, 112, 113, 114, 115, 116, 117, 118]. To gain insight into the aforementioned class of dynamical systems and predict their behaviors in different operational or loading conditions, it is beneficial to create models using the measured input-output response data.

Dynamic mode decomposition (DMD) has become a popular tool in data-driven modeling, owing to its ability to decompose the high dimensional data into its coherent spatio-temporal structures [22, 23]. DMD has its roots in the Koopman theory [19], whose work was later revived by seminal works of Mezić et al. [119, 120]. Koopman theory in essence, associates a nonlinear dynamical system with an infinite-dimensional linear system, allowing tools for linear systems theory to be employed. Different variants of DMD have been proposed to improve the pre-processing or post-processing capabilities of the standard DMD. Optimized DMD [121] and sparsity promoting DMD [122] transforms the approximation of the linear operator into an optimization problem with constraints in the eigenvalues, modes, or mode amplitudes. Multi-resolution DMD [38] and higher-order DMD [123] provide a recursive way to improve the frequency resolution and transient handling capability of the standard DMD. Kernel-based DMD [124] and Extended DMD (EDMD) [92] provide a means to extend the framework towards nonlinear systems by creating meaningful observables. Proctor et al. [36] developed a variant of DMD known as DMD with controls (DMDc) to incorporate the input signals into the DMD framework. Benner et al. [37] developed an extension of DMDc, known as input-output DMD (ioDMD), by providing means to incorporate outputs alongside inputs and states. The ioDMD and DMDc algorithm provide an elegant way to extend the standard DMD to include the effects of external forcing functions.

Although DMD is widely popularised across diverse fields such as fluid flow [37, 42, 120, 125], epidemiology [126], neuroscience [25], video processing [127], to our knowledge it has not been widely employed in the field of structural dynamics. This gap in the literature is partly attributed to the strong dominance of principal component analysis among structural dynamics researchers [128, 129] and also to the requirement of high dimensional data (full state measurements) for DMD [22]. The requirement of full state measurements for the DMD is somewhat restrictive, limiting the methodology's application in fields with high dimensional data. In practice, for mechanical systems, responses can only be measured at specific strategic locations owing to limitations in the acquisition hardware, and internal full model operators are seldom available. For low dimensional systems and systems with limited measurements, researchers have taken inspiration from Taken's embedding theory [96] and

proposed applying the DMD procedure on time-shifted coordinates [24, 25, 123, 125, 130]. These methods require accurate tuning of the time delays (or hyper-parameters), which is often problem-specific, and noise in the observed data may lead to erroneous results [125, 131]. Recent works have proposed techniques to address these problems and the interested reader is referred to, e.g., [42, 130, 132, 133] and the references therein for more details.

In the present study, for systems with a limited number of measurements a novel data-driven methodology called Wavelet-based Dynamic mode Decomposition (WDMD) is proposed. The proposed methodology builds on ioDMD and utilizes the wavelet decomposition of measured responses as observables, thereby enlarging the state dimensions. In other words, wavelet coefficients of the measured outputs serve as the pseudo-states of the dynamical system, and the DMD framework approximates a linear operator that advances the pseudo-states by a time step. The present approach can be thought of as a particular case of EDMD with a special choice of the observables. Hence, the present methodology can be applied to model dynamical systems such as a vibrating mechanical system, whose internal state vectors are not readily available, and only data from a few output locations are accessible, which is often the case in practice.

In the present context, data-driven modeling creates numerical models for capturing the input-output characteristics of the underlying structure. The advantage with such models is that it relies completely on measured responses, thereby circumventing the need for the knowledge of any underlying governing dynamics of the structure. In this paper, the data corresponds to time-domain samples of the input-output trajectories, and modeling equates to the best fit linear operator that advances the system states by a time step. In this paper, the data is fixed, only some outputs are observed and one cannot go back to re-query the dynamics. The authors refer the reader to [26, 132], which deals with the partial state observation where it is possible to re-query the system dynamics at each stage. For data-driven techniques that uses frequency domain samples, the authors refer the reader to [11, 108, 109, 118, 134, 135, 136] and the references therein.

The major contributions of this papers are as follows: First, a new data-driven modeling methodology based on WDMD that utilizes only the input-output trajectories of the system is proposed. To the authors' best knowledge, extending ioDMD's applicability through the use of wavelets has not yet been explored. The proposed methodology provides at least the same and in many cases better quality of the fit using only input-output trajectories, compared with the baseline approach, which has access to full state information. In addition, our present work provides a means to apply the DMD algorithm towards modal analysis and structural vibration of a mechanical system. Finally, these numerical results are complemented with experimental tests on a free-free aluminum cantilever beam, in which WDMD methodology is utilized to develop data-driven models from (noisy) real data.

The paper is structured as follows. First, a brief description of DMD, ioDMD, and EDMD are presented in Section 4.2. In Section 4.3, our work's major contribution, WDMD, is derived. The proposed methodology is demonstrated on data from a simulated finite element model

of a cantilever beam in Section 4.4. In Section 4.5, experimental case studies are carried out on a free-free beam to demonstrate the efficiency and robustness of the WDMD in approximating practical mechanical systems. Conclusions and potential future directions are given in Section 4.6.

4.2 Background

4.2.1 Dynamic mode decomposition

In this section, a brief introduction to the classical dynamic mode decomposition (DMD) framework is provided. For details, we refer the reader to [20, 90, 137] and the references therein. Consider the system of time-invariant ordinary differential equations of the form

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)), \tag{4.1}$$

where $x(t) \in \Re^N$ is the state vector and $f : \Re^N \to \Re^N$ is a nonlinear map. Given the sampling times $t_0, t_1, \ldots, t_K$ (equally distanced), let $\{\mathbf{x}(t_0),\ \mathbf{x}(t_1),\ \ldots,\ \mathbf{x}(t_K)\}$ denote the samples of the state $\mathbf{x}(t)$ of dynamical system eq. (6.7). For this data, define two snapshot matrices $\mathbf{X}_0$ and $\mathbf{X}_1$ as

$$\mathbf{X}_0 = \begin{bmatrix}\mathbf{x}(t_0) & \mathbf{x}(t_1) & \cdots & \mathbf{x}(t_{K-1})\end{bmatrix} \in \Re^{N\times K} \quad \text{and} \quad \mathbf{X}_1 = \begin{bmatrix}\mathbf{x}(t_1) & \mathbf{x}(t_2) & \cdots & \mathbf{x}(t_K)\end{bmatrix} \in \Re^{N\times K}, \tag{4.2}$$

where $\mathbf{X}_0$ denotes the snapshot matrix from t_0 to t_{K-1} and $\mathbf{X}_1 \in \Re^{N\times K-1}$ from t_1 to t_K, which advances the $\mathbf{X}_0$ matrix by one time step. The most fundamental form of DMD aims to explain the snapshot data with a linear dynamical system of the form,

$$\mathbf{x}(t_{k+1}) \approx \mathbf{A}\mathbf{x}(t_k), \quad \text{for} \quad k = 0, 1, 2, .., K-1, \ \text{where } \mathbf{A} \in \Re^{N\times N}. \tag{4.3}$$

In terms of $\mathbf{X}_0$ and $\mathbf{X}_1$, the approximation in eq. (6.11) can be written in the matrix form as

$$\mathbf{X}_1 \approx \mathbf{A}\mathbf{X}_0. \tag{4.4}$$

The DMD algorithm finds the best-fit solution $\mathbf{A}$, one that minimizes the least-squares distance in the Frobenius norm, i.e.,

$$\mathbf{A} = \underset{\hat{\mathbf{A}}\in\mathbb{R}^{N\times N}}{\arg\min} \left\| \mathbf{X}_1 - \hat{\mathbf{A}}\mathbf{X}_0 \right\|_F. \tag{4.5}$$

The optimal solution $\mathbf{A}$ in eq. (4.5) is given by

$$\mathbf{A} = \mathbf{X}_1 \mathbf{X}_0{}^{\dagger}, \tag{4.6}$$

where $\mathbf{X_0}^{\dagger} \in \Re^{K \times N}$ denotes the Moore-Penrose inverse of $\mathbf{X_0} \in \Re^{N \times K}$. The practical issues in computing the $\mathbf{A}$ matrix involves algebraic assumptions and singular value decomposition of the $\mathbf{X_0}$ matrix, which are skipped here for brevity. Interested readers are directed to [20, 91, 137, 138, 139] for more rigorous discussion on practical algorithms and computational considerations.

4.2.2 input-output Dynamic mode decomposition (ioDMD)

The DMD method as described in the previous section can only be used for systems that evolve on their own, with no external input and for which all states are assumed to be measured. However, dynamical systems in practice have external inputs and the form as represented by eq. (6.11) will not be sufficient to explain the dynamics [140]. This lead to the development of DMD with controls (DMDc) [36] by including measurements of a control input $\mathbf{u}(t)$. The input-output DMD is a further extension of the DMDc by incorporating the observed outputs [37, 141]. The ioDMD framework constructs a reduced-order model directly from the observed input-output data and the full state vector $\mathbf{x}(t)$. The ioDMD framework models the dynamical systems of the form

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{y}(t) &= g(\mathbf{x}(t), \mathbf{u}(t)), \end{aligned} \tag{4.7}$$

where $\mathbf{u}(t) \in \Re^m$ denotes the inputs driving the system and $\mathbf{y}(t) \in \Re^d$ is the measured output.

The ioDMD method approximates the evolution of eq. (6.12) with a linear dymnamical system of the form

$$\begin{aligned} \mathbf{x}(t_{k+1}) &\approx \mathbf{A}\mathbf{x}(t_k) + \mathbf{B}\mathbf{u}(t_k), \\ \mathbf{y}(t_k) &\approx \mathbf{C}\mathbf{x}(t_k) + \mathbf{D}\mathbf{u}(t_k), \end{aligned} \tag{4.8}$$

where $\mathbf{A} \in \Re^{N \times N}, \mathbf{B} \in \Re^{N \times m}, \mathbf{C} \in \Re^{d \times N}, \mathbf{D} \in \Re^{d \times m}$. In addition to the state snapshots $\mathbf{X_0}$ and $\mathbf{X}_1$ in eq. (4.2), define the input and output snapshot matrices as

$$\mathbf{U_0} = \begin{bmatrix} \mathbf{u}(t_0) & \mathbf{u}(t_1) & \dots & \mathbf{u}(t_{K-1}) \end{bmatrix} \in \Re^{m \times K} \text{ and } \mathbf{Y}_0 = \begin{bmatrix} \mathbf{y}(t_0) & \mathbf{y}(t_1) & \dots & \mathbf{y}(t_{K-1}) \end{bmatrix} \in \Re^{d \times K}. \tag{4.9}$$

This would entail writing eq. (6.13) in terms of its matrix counterpart as

$$\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{Y_0} \end{bmatrix} \approx \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X_0} \\ \mathbf{U_0} \end{bmatrix}. \tag{4.10}$$

Let

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \in \Re^{(N+d) \times (N+m)} \tag{4.11}$$

denote the optimal solution to eq. (6.14). Similar to the original DMD framework, the

ioDMD method finds the optimal solution $\boldsymbol{\Upsilon}$ by solving a least-squares problem, namely

$$\boldsymbol{\Upsilon} = \underset{\hat{\boldsymbol{\Upsilon}} \in \Re^{(N+d)\times(N+m)}}{\arg\min} \left\| \begin{bmatrix} \mathbf{X_1} \\ \mathbf{Y_0} \end{bmatrix} - \hat{\boldsymbol{\Upsilon}} \begin{bmatrix} \mathbf{X_0} \\ \mathbf{U_0} \end{bmatrix} \right\|_{\mathrm{F}}. \tag{4.12}$$

The optimal $\boldsymbol{\Upsilon}$ in ioDMD is given by

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{X_1} \\ \mathbf{Y_0} \end{bmatrix} \begin{bmatrix} \mathbf{X_0} \\ \mathbf{U_0} \end{bmatrix}^{\dagger}. \tag{4.13}$$

Practical issues that arise in computing eq. (4.6) arise here as well. Computing the pseudo-inverse in eq. (4.13) often involves inverting small non-zero singular values, thereby leading to numerical instabilities. Therefore, in practice, singular values below a relative tolerance $\beta \in \Re^{+}$are truncated during the pseudo-inverse computation or other regularization techniques could be employed. One can also perform model reduction on the state snapshot matrix $\mathbf{X}$ to further reduce the state-space dimension of the output linear dynamical system. This is carried out by performing an additional SVD-based projection step before solving the least squares problem in eq. (4.12). We refer the reader to [20, 37] for details.

4.2.3 Extended DMD and the Koopman operator

Koopman theory [19] has received considerable attention recently due to the pioneering work of Mezić et al. [142]. It has been shown that the DMD algorithm is a special case of Koopman theory applied to linearly consistent data [120, 142] and the DMD modes approximate the Koopman eigenvalues if the set of observable is sufficiently large (i.e., it spans the eigenvectors of the Koopman operator) and the data has to be sufficiently rich (i.e., it covers the dynamics of interest) [20, 92].

In the case of a linear system with full measurements, linear observable or full state measurements are sufficient as they span the eigenvectors of the Koopman operator. However, in the case of partially observed state measurement or non-linear systems, direct application of the DMD algorithm falls short of recovering the underlying dynamics. This observation has lead to the development of extended DMD (EDMD) by Williams et al. [92], which creates new observables, $\boldsymbol{\varphi}$, from the state vector. To give context to the proposed algorithm, WDMD, it is pertinent to introduce EDMD, and the present section serves to do so. The Koopman operator $\mathcal{K}$ acts directly on observables $\boldsymbol{\varphi}$ rather than on state-space [19, 92], i.e.,

$$\mathcal{K}\boldsymbol{\varphi} \triangleq \boldsymbol{\varphi} \circ \mathcal{F} \quad \Rightarrow \quad \mathcal{K}\boldsymbol{\varphi}\left(\mathbf{x}(t_k)\right) = \boldsymbol{\varphi}\left(\mathbf{x}(t_{k+1})\right), \tag{4.14}$$

where $\mathcal{F} : \mathcal{M} \to \mathcal{M}$ is the evolution operator, and $\circ$ denotes the composition operator. Intuitively, the linear Koopman operator takes a scalar function φ and returns a new function $\mathcal{K}\varphi$ that predicts the value of φ, one step ahead in future. It is to be noted that the dynamical system defined by $\mathcal{F}$ and the one defined by $\mathcal{K}$ are two different parameterizations of the

same fundamental behavior.

EDMD aims to approximate the Koopman operator using a suitable choice of dictionary of observables, $\mathcal{D} = [\varphi_1, \varphi_2, ..., \varphi_{N_K}]$. The vector valued function $\mathbf{\Phi} : \mathcal{M} \to \mathbb{C}^{1\times N_K}$, where

$$\mathbf{\Phi}(\boldsymbol{x}) = [\varphi_1(\boldsymbol{x}), \varphi_2(\boldsymbol{x}), ..., \varphi_{N_K}(\boldsymbol{x})], \tag{4.15}$$

can now be defined for the snapshot of the system, $\mathbf{x}(t_k)$, and for $\mathbf{z}(t_k) = \mathcal{F}(\mathbf{x}(t_k))$. Then, one proceeds by defining two snapshot matrices using the samples $\mathbf{\Phi}(\mathbf{x}(t_k))$, and $\mathbf{\Phi}(\mathcal{F}(\mathbf{x}(t_k)))$ for $k = 0, 1, \ldots, K$ and solves a least-squares problem to form the best fit matrix $\mathbf{K} \in \Re^{N_K\times N_K}$. The eigenvalues and eigenvectors of the finite dimensional representation $\mathbf{K}$ are then used to compute an approximation to the Koopman modes and Koopman eigenfunctions. Thus, EDMD provides a mean to approximate the infinite dimensional Koopman operator and Koopman eigenfunctions through the selection of observables. For details, the reader is referred to [92].

4.3 Wavelet-based dynamic mode decomposition

The main idea behind the present methodology is to create observables using the stationary wavelet coefficients of all the available output measurements, and thereby to approximate the Koopman operator that advances these observables by a time step. Central to this idea is the wavelet transform. The next subsection, which mainly follows [143], presents a brief overview of the necessary background on wavelet transform. We refer the reader to [143, 144, 145] for details.

4.3.1 Maximal overlap discrete wavelet transform (MODWT)

The wavelet transform convolves a signal with a function called the mother wavelet, and the transform is computed across several scales representing different frequency bands for different segments of the signal. The wavelet transform provides a multi-resolution analysis (in contrast to the Fourier transform, which has a uniform time-frequency distribution.) The orthogonal wavelet decomposition of a signal $y(t)$ is given by,

$$y(t) = \sum_j \sum_k w_k^j \psi_k^j, \tag{4.16}$$

with the wavelet coefficient w_k^j given by the inner product

$$w_k^j(y) = \langle y, \psi_k^j \rangle = \left\langle y(t), \frac{1}{2^{j/2}} \psi\left(\frac{t}{2^j} - k\right) \right\rangle = \frac{1}{2^{j/2}} \int_{-\infty}^{\infty} y(t) \psi^*\left(\frac{t}{2^j} - k\right) dt, \tag{4.17}$$

where the function ψ represents the mother wavelet, $\psi_k^j = \frac{1}{2^{j/2}}\psi\left(\frac{t}{2^j} - k\right)$ is the scaled and translated mother wavelet, and $(\cdot)^*$ denotes the complex conjugation. The transform is usually computed at discrete values in a grid corresponding to dyadic values of 2^j and translations of k, where both j, k are integers, yielding the discrete wavelet transform (DWT).

In practice, successive high and low-pass filtering replaces the integration procedures in eq. (5.3). This is followed by down-sampling at each level [143]. The coefficients resulting from these operations are called approximation and detail coefficients. The details of DWT implementation are not mentioned here for brevity, and interested readers are referred to seminal works such as [143]. Due to down sampling at each level, DWT wavelet coefficients do not have the property of time invariance.

The aforementioned issue can be addressed through a special type of wavelet transform known as the maximal overlap discrete wavelet transform (MODWT) [145]. MODWT has the advantage that it can eliminate down-sampling, thereby resulting in wavelet detail and scale coefficients at each level of the same length as the original time series, thereby facilitating a ready comparison between the series and its decomposition. Decomposing the time-series $\{y(t_0), y(t_1), ..., y(t_{K-1})\}$ using MODWT to J levels involves the application of J pairs of filters. The filtering procedure at j^{th} level entails applying a high-pass filter $(\widetilde{h}_{j,l}^0)$ known as wavelet filter, and low-pass filter $(\widetilde{g}_{j,l}^0)$ known as scaling filter, where $l = 1, 2, ..L_j$ is the length of the filter. This procedure yields a set of wavelet and scaling coefficients at each level j as

$$\widetilde{W}_{j,t_k} = \sum_{l=0}^{K-1} \widetilde{h}_{j,l} y(t_{k-l \ mod \ K}), \qquad \widetilde{V}_{j,t_k} = \sum_{l=0}^{K-1} \widetilde{g}_{j,l} y(t_{k-l \ mod \ K}), \tag{4.18}$$

where $\widetilde{h}_{j,l}$ is $\widetilde{h}_{j,l}^0$ periodized to length K and $\widetilde{g}_{j,l}$ follows analogously using $\widetilde{g}_{j,l}^0$ values, and mod represents the modular operator; see [145] for details. The equivalent wavelet filter $(\widetilde{h}_{j,l})$ and scaling filter $(\widetilde{g}_{j,l})$ for the j^{th} level are a set of scale-dependent localized differencing and averaging operators, respectively, and can be regarded as stretched versions of the base filter $(j = 1)$. The MODWT wavelet coefficients at each scale will have the same length as the original signal $y(t)$ as seen from eq. (5.5). Define the time-series vector

$$\boldsymbol{y} = [y(t_0), y(t_1), ..., y(t_{K-1})]^T \in \Re^K. \tag{4.19}$$

Then eq. (5.5) can be expressed in matrix form as

$$\widetilde{\mathbf{W}}_j = \widetilde{\mathcal{W}}_j \boldsymbol{y} \ \text{ and } \ \widetilde{\mathbf{V}}_j = \widetilde{\mathcal{V}}_j \boldsymbol{y}, \tag{4.20}$$

where $\widetilde{\mathbf{W}}_j = [\widetilde{W}_{j,t_0}, \ \widetilde{W}_{j,t_1}, \ \ldots, \widetilde{W}_{j,t_{K-1}}]^T \in \Re^K$ and $\widetilde{\mathbf{V}}_j = [\widetilde{V}_{j,t_0}, \ \widetilde{V}_{j,t_1}, \ \ldots, \widetilde{V}_{j,t_{K-1}}]^T \in \Re^K$ represent the j^{th} level MODWT wavelet and scaling coefficients, respectively. The $K \times K$

matrix $\widetilde{\mathcal{W}}_j$ is defined as

$$\widetilde{\mathcal{W}}_j = \frac{1}{2^k}\begin{bmatrix} \widetilde{h}_{j,0} & \widetilde{h}_{j,K-1} & \widetilde{h}_{j,K-2} & \dots & \widetilde{h}_{j,3} & \widetilde{h}_{j,2} & \widetilde{h}_{j,1} \\ \widetilde{h}_{j,1} & \widetilde{h}_{j,0} & \widetilde{h}_{j,K-1} & \dots & \widetilde{h}_{j,4} & \widetilde{h}_{j,3} & \widetilde{h}_{j,2} \\ \widetilde{h}_{j,2} & \widetilde{h}_{j,1} & \widetilde{h}_{j,0} & \dots & \widetilde{h}_{j,5} & \widetilde{h}_{j,4} & \widetilde{h}_{j,3} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ \widetilde{h}_{j,K-2} & \widetilde{h}_{j,K-3} & \widetilde{h}_{j,K-4} & \dots & \widetilde{h}_{j,1} & \widetilde{h}_{j,0} & \widetilde{h}_{j,K-1} \\ \widetilde{h}_{j,K-1} & \widetilde{h}_{j,K-2} & \widetilde{h}_{j,K-3} & \dots & \widetilde{h}_{j,2} & \widetilde{h}_{j,1} & \widetilde{h}_{j,0} \end{bmatrix}, \tag{4.21}$$

and the $K \times K$ matrix $\widetilde{\mathcal{V}}_j$ is defined analogously using $\widetilde{g}_{j,l}$ values; see [145] for details. The original time series y can be recovered from its MODWT via

$$\boldsymbol{y} = \sum_{j=1}^{J} \widetilde{\mathcal{W}}_j^{\mathrm{T}}\widetilde{\mathbf{W}}_j + \widetilde{\mathcal{V}}_j^{\mathrm{T}}\widetilde{\mathbf{V}}_j = \sum_{j=1}^{J} \widetilde{\mathbf{D}}_j + \widetilde{\mathbf{S}}_J \quad \text{where} \quad \widetilde{\mathbf{D}}_j := \widetilde{\mathcal{W}}_j^{\mathrm{T}}\widetilde{\mathbf{W}}_j \in \Re^K \quad \text{and} \quad \widetilde{\mathbf{S}}_j = \widetilde{\mathcal{V}}_j^{\mathrm{T}}\widetilde{\mathbf{V}}_j \in \Re^K. \tag{4.22}$$

The last equality defines a MODWT-based multi-resolution analysis (MRA) of the original time series $\boldsymbol{y}$ in terms of j^{th} level MODWT detail coefficients $\widetilde{\mathbf{D}}_j$ and j^{th} level MODWT smooth coefficients $\widetilde{\mathbf{S}}_j$.

4.3.2 Main approach

Consider an underlying dynamical system evolving in an N-dimensional state-space, i.e.,

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \tag{4.23}$$

where $\mathbf{x}(t) \in \Re^N$ is the state, $\mathbf{u}(t) \in \Re^m$ is the input, and $f : \Re^N \to \Re^N$ is a nonlinear mapping. Assume that, unlike in DMD or ioDMD, we do not have access to the full-state samples $\mathbf{x}(t_k)$. Instead, we have only access to a measurement vector (output) $\mathbf{y}(t) \in \Re^d$ via an observation (output) matrix $\mathbf{C} \in \Re^{d\times N}$, i.e., we have access to the output

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t). \tag{4.24}$$

Assume that dynamics are sampled at time instances $t_0, t_1, \dots, t_K$, yielding the measurement samples

$$\mathbf{y}(t_k) = \mathbf{C}\mathbf{x}(t_k), \quad \text{for} \quad k = 0, 1, \dots, K-1. \tag{4.25}$$

Based on MODWT analysis of the previous section, our goal is now to create new auxiliary state variables (and an observation matrix) so that the new dynamics with the auxiliary state still corresponds to the true output samples in eq. (4.25). Then we can apply the ioDMD using the trajectories of the original input and outputs, and the trajectories of the auxiliary states.

Towards this goal, let $y_i(t)$, for $i = 1, 2, ..., d$, denote the ith component (row) of the measurement vector $\mathbf{y}(t)$, i.e., $y_i(t)$ is the ith output. Decompose $y_i(t)$ using MODWT as in eq. (5.7):

$$\boldsymbol{y}_i = \sum_{j=1}^{J} \widetilde{\mathbf{D}}_j^{(i)} + \widetilde{\mathbf{S}}_J^{(i)} \qquad \text{where} \qquad \boldsymbol{y}_i = [y_i(t_0), y_i(t_1), ..., y_i(t_{K-1})]^T, \tag{4.26}$$

and $\mathbf{D}_j^{(i)} \in \Re^K$ and $\mathbf{S}_j^{(i)} \in \Re^K$ are the corresponding jth level detail and smooth coefficients corresponding to $y_i(t)$. Let $\mathbf{e}_k \in \Re^K$ denote the kth canonical vector and $\mathbf{e} = [1\ 1 ...\ 1]^T \in \Re^{J+1}$ denote the vector of ones. Then, using eq. (5.11), $y_i(t_k)$ (the $(K+1)$st row of $\boldsymbol{y}_i = [y_i(t_0), y_i(t_1), ..., y_i(t_{K-1})]^T$) can be written as

$$y_i(t_k) = \mathbf{e}_{k+1}^T \boldsymbol{y}_i = \sum_{j=1}^{J} \mathbf{e}_{k+1}^T \widetilde{\mathbf{D}}_j^{(i)} + \mathbf{e}_{k+1}^T \widetilde{\mathbf{S}}_J^{(i)} = \mathbf{e}^T \mathbf{w}_i(t_k) \tag{4.27}$$

where

$$\mathbf{w}_i(t_k) = \begin{bmatrix} \mathbf{e}_{k+1}^T \widetilde{\mathbf{D}}_1^{(i)} \\ \mathbf{e}_{k+1}^T \widetilde{\mathbf{D}}_2^{(i)} \\ \vdots \\ \mathbf{e}_{k+1}^T \widetilde{\mathbf{D}}_J^{(i)} \\ \mathbf{e}_{k+1}^T \widetilde{\mathbf{S}}_J^{(i)} \end{bmatrix} \in \Re^{J+1}. \tag{4.28}$$

Then, the full output vector at time t_k, i.e., $\mathbf{y}(t_k)$ in eq. (4.25), can be rewritten as

$$\mathbf{y}(t_k) = \begin{bmatrix} y_1(t_k) \\ y_2(t_k) \\ \vdots \\ y_d(t_k) \end{bmatrix} = \begin{bmatrix} \mathbf{e}^T \mathbf{w}_1(t_k) \\ \mathbf{e}^T \mathbf{w}_2(t_k) \\ \vdots \\ \mathbf{e}^T \mathbf{w}_d(t_k) \end{bmatrix}. \tag{4.29}$$

Using the last formula, we define the new auxiliary state $\mathbf{z}(t)$ and the observation matrix $\mathbf{C_w}$

$$\mathbf{z}(t_k) = \begin{bmatrix} \mathbf{w}_1(t_k) \\ \mathbf{w}_2(t_k) \\ \vdots \\ \mathbf{w}_d(t_k) \end{bmatrix} \in \Re^{d(J+1)} \quad \text{and} \quad \mathbf{C_w} = \begin{bmatrix} \mathbf{e}^T & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{e}^T & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \mathbf{e}^T & 0 \\ 0 & 0 & \cdots & 0 & \mathbf{e}^T \end{bmatrix} \in \Re^{d \times d(J+1)}, \tag{4.30}$$

so that

$$\mathbf{y}(t_k) = \mathbf{C_w} \mathbf{z}(t_k). \tag{4.31}$$

Note that the new auxiliary state variable $\mathbf{z}(t)$ is composed of the wavelet coefficient observables and thus its samples encodes how the wavelet coefficient observables evolve over time. Moreover, with the observation matrix $\mathbf{C_w}$, the true output/measurement vector $\mathbf{y}(t)$ is written in terms of the new state variable $\mathbf{z}(t)$.

Given *only* the output snapshots $\mathbf{y}(t_k)$ in eq. (4.25) of the underlying dynamical system, use eq. (5.11)-eq. (5.14) to construct the snapshot matrix $\mathbf{Z}$ of the wavelet coefficient observables as

$$\mathbf{Z} = [\mathbf{z}(t_0) \;\; \mathbf{z}(t_1) \;\; \mathbf{z}(t_2) \;\; \ldots \; \mathbf{z}(t_K)] \in \Re^{d(J+1)\times(K+1)}. \tag{4.32}$$

Also construct the input snapshot matrix $\mathbf{U}_0$ and $\mathbf{Y}_0$,

$$\mathbf{U_0} = \begin{bmatrix} \mathbf{u}(t_0) & \mathbf{u}(t_1) & \ldots & \mathbf{u}(t_{K-1}) \end{bmatrix} \in \Re^{M\times K} \;\text{ and }\; \mathbf{Y}_0 = \begin{bmatrix} \mathbf{y}(t_0) & \mathbf{y}(t_1) & \ldots & \mathbf{y}(t_{K-1}) \end{bmatrix} \in \Re^{d\times K}. \tag{4.33}$$

Note that while the input snapshot matrix $\mathbf{U}_0$ and the output snapshot matrix $\mathbf{Y}_0$ in eq. (5.16) correspond to the true inputs and outputs of the underlying dynamical system eq. (4.23) and eq. (4.24), the state snapshot matrix $\mathbf{Z}$ in eq. (6.21) are obtained via the wavelet coefficient observables (as the original state measurements $\mathbf{x}(t_k)$ are not available). Then, WDMD represents the snapshot triplets $\mathbf{Z}, \mathbf{U}_0$ and $\mathbf{Y}_0$ with the dynamical system

$$\mathbf{z}(t_{k+1}) \approx \mathbf{A_w}(t_k)\mathbf{z}(t_k) + \mathbf{B_w}\mathbf{u}(t_k), \quad \mathbf{y}(t_k) \approx \mathbf{C_w}\mathbf{z}(t_k) + \mathbf{D_w}\mathbf{u}(t_k). \tag{4.34}$$

This reformulation of the input/output data via wavelet coefficient observables to input/state/output data allows to apply ioDMD to construct the matrices $\mathbf{A_w}$, $\mathbf{B_w}$, $\mathbf{C_w}$, and $\mathbf{D_w}$ via a least-squares fit as in Section 4.2.2. Towards this goal, define the two matrices

$$\mathbf{Z}_0 = [\mathbf{z}(t_0),\; \mathbf{z}(t_1), \ldots, \mathbf{z}(t_{K-1})] \in \Re^{d.(J+1)\times K} \;\text{ and }\; \mathbf{Z}_1 = [\mathbf{z}(t_1),\; \mathbf{z}(t_1), \ldots, \mathbf{z}(t_K)] \in \Re^{d.(J+1)\times K}. \tag{4.35}$$

Then, the dynamical system coefficients in eq. (5.18) are given by

$$\begin{bmatrix} \mathbf{A}_w & \mathbf{B}_w \\ \mathbf{C}_w & \mathbf{D}_w \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Y}_0 \end{bmatrix} \begin{bmatrix} \mathbf{Z}_0 \\ \mathbf{U}_0 \end{bmatrix}^{\dagger}. \tag{4.36}$$

A brief algorithmic sketch of WDMD is given in Algorithm 1.

Given only the output samples, the present work's major contribution is to enlarge the original subspace via wavelet decomposition of the response measurement $\mathbf{y}(t)$ and creating new states of the system using the wavelet coefficients. Therefore, the WDMD methodology can be considered a special case of EDMD with choice of observables in eq. (4.15) resulting from the wavelet coefficients $w_k^j(\boldsymbol{y}) = \langle \boldsymbol{y}, \psi_k^j \rangle$. Thus, WDMD provides a set of basis functions or a kernel operator, which lifts the output measurements to wavelet states, thereby potentially spanning the eigenvectors of the Koopman operator and approximating the Koopman operator through ioDMD.

Algorithm 1 WDMD algorithm

Input: Output measurements $\{\mathbf{y}(t_i)\} \in \Re^d$ and input measurements $\{\mathbf{u}(t_i)\} \in \Re^m$ for $i = 0, 1, \ldots, K$.

Output: State-space model: $\mathbf{A_w} \in \Re^{d.(J+1)\times d.(J+1)}$, $\mathbf{B_w} \in \Re^{d.(J+1)\times m}$, $\mathbf{C_w} \in \Re^{d\times d.(J+1)}$, and $\mathbf{D_w} \in \Re^{d\times m}$.

1: Using $\{\mathbf{y}(t_i)\}$, construct the wavelet observable snapshots $\{\mathbf{z}(t_i)\}$ using eq. (5.11) – eq. (5.14).
2: Form $\mathbf{Z} = [\mathbf{z}(t_0) \;\; \mathbf{z}(t_1) \;\; \mathbf{z}(t_2) \;\; \ldots \mathbf{z}(t_K)]$ as in eq. (6.21)
3: Assemble $\mathbf{Z_0}$ and $\mathbf{Z_1}$ as in eq. (4.35), and $\mathbf{Y_0}$ and $\mathbf{U_0}$ as in (5.16).
4: Compute the approximate discrete linear state-space matrices as in eq. (5.19)

4.4 A numerical case study using a finite element beam

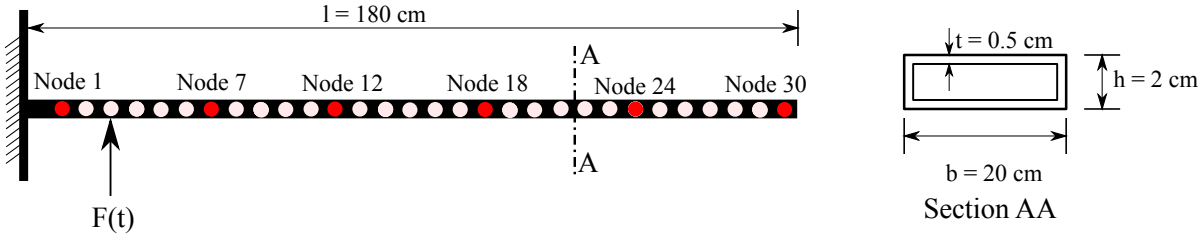


Figure 4.1: Cantilever beam model used for the Finite Element simulations

Numerical simulations are carried out on a hollow cantilever beam model with the dimensions shown in Figure 4.1. The beam under study is a finite element representation of an Euler-Bernoulli beam with 30 nodal points representing 60 degrees of freedom (DOF). Taking the displacement and velocity of each DOF as the states yields a state-space representation in the first-order form

$$\dot{\mathbf{x}}(t) \;=\; \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{G} \end{bmatrix}}_{:=\mathbf{A}} \mathbf{x}(t) \;+\; \underbrace{\begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^{-1}\mathbf{F} \end{bmatrix}}_{:=\mathbf{B}} \mathbf{u}(t), \qquad \mathbf{y}(t) \;=\; \mathbf{C}\mathbf{x}(t), \tag{4.37}$$

where $\mathbf{M}, \mathbf{K}, \mathbf{G} \in \Re^{60\times 60}$ are, respectively, the mass, stiffness, and damping matrices and $\mathbf{F} \in \Re^{60\times 1}$ is the loading vector; $\mathbf{x} \in \Re^{120\times 1}$ is the state vector; $\mathbf{u}(t) \in \Re$ is the scalar input; and $\mathbf{y} \in \Re^d$ is the d-dimensional output vector. This yields the first-order state-space quantities $\mathbf{A} \in \Re^{120\times 120}$, $\mathbf{B} \in \Re^{120}$, and $\mathbf{C} \in \Re^{d\times 120}$. The outputs, observed in $\mathbf{y}$, can be either the displacement or velocity of the observed nodal points. The choice of output will be further clarified below.

The proposed WDMD approach will be utilized to generate a single-input/multiple-output (SIMO), data-driven approximation to the the beam model in eq. (4.37) using *only* the

simulated input-output response of the beam *without* access to its state-space matrices. This model will then be used to simulate the transient dynamic response of the structure to a given excitation (a testing signal) to illustrate the quality of the fit. In addition to this time-domain error measure, the input-output mapping of the data-driven model can also be assessed in the frequency domain by computing the frequency response function (FRF) of the learned model and comparing it with the original FRF. The FRF of the beam model eq. (4.37), denoted by $\mathcal{H}(\omega)$, is given by

$$\mathcal{H}(\omega) = \mathbf{C}(\imath\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}, \tag{4.38}$$

where $\imath^2 = -1$. Let $\tilde{\mathcal{H}}$ denote the FRF of the learned model and $\tilde{\mathbf{y}}(t)$ the output of the learned. Then, the following two relative error metrics are defined to evaluate the quality of the fit,

$$\epsilon_{fd}^{rel} = \sqrt{\frac{\sum_{j=1}^{L_\omega} \|\mathcal{H}(\omega_j) - \tilde{\mathcal{H}}(\omega_j)\|_2^2}{\sum_{j=1}^{L_\omega} \|\mathcal{H}(\omega_j)\|_2^2}} \quad \text{and} \quad \epsilon_{td}^{rel} = \sqrt{\frac{\sum_{i=1}^{K} ||\mathbf{y}(t_i) - \tilde{\mathbf{y}}(t_i)||_2^2}{\sum_{i=1}^{K} ||\mathbf{y}(t_i)||_2^2}}, \tag{4.39}$$

where L_ω and K are the number of frequency samples and time points respectively. While ϵ_{fd}^{rel} is the relative error between the original FRF ($\mathcal{H}(\omega)$) and the fitted FRF ($\tilde{\mathcal{H}}(\omega)$), ϵ_{td}^{rel} measures the relative error in time domain between the measured responses $\mathbf{y}(t)$ and the predicted responses $\tilde{\mathbf{y}}(t)$.

In the present numerical case study, the FEM beam is excited using a chirp input over the frequency range $10-800$ Hz. The responses are collected at a sampling frequency of 5000 Hz. The application of the ioDMD methodology, which assumes access to full state observation, for modeling the input-output response of the FEM beam, is demonstrated in Section 4.4.1. Next, in Section 6.3.1 we compare these results with WDMD. In Section 4.4.4 we present a brief comparative study between WDMD and Delay-DMD. As mentioned earlier, one can perform an additional model reduction via an SVD-based projection on the state data to further reduce the learned system dynamics [20, 37] . In the present study, this additional step has yielded negligible changes to the final data-driven model and thus is skipped in all the results.

4.4.1 Data-driven modeling using ioDMD

As discussed in Section 4.2.2, the ioDMD methodology assumes knowledge about the system's full internal states $\mathbf{x}(t)$. Hence, the ioDMD is ideally suited towards grey box modeling wherein the internal states of the system are also sampled. For the beam's finite element model, the internal states represent the displacement and velocity at each degree of freedom. The training package provided to the algorithm consists of: i) the input forcing signal used to excite the structure (chirp signal), ii) the internal state measurements, and iii) the mea-

sured output responses. In the current example, the ioDMD has access to all the internal states of the system and the output is assumed to be measured at the 6 nodal points shown in Figure 4.1. The measured displacements at nodes $1, 7, 12, 18, 24$, and 30 are designated as the output responses in the present section. From the provided training package, the ioDMD algorithm, as presented in Section 4.2.2, generates a linear discrete dynamical system of the form,

$$\begin{aligned} \mathbf{x}(t_{k+1}) &= \mathbf{A}\mathbf{x}(t_k) + \mathbf{B}\mathbf{u}(t_k), \\ \mathbf{y}(t_k) &= \mathbf{C}\mathbf{x}(t_k) + \mathbf{D}\mathbf{u}(t_k), \end{aligned} \tag{4.40}$$

to approximate the original beam dynamics in eq. (4.37). The singular value truncation tolerance in the computation of the pseudoinverse in eq. (4.13) is set to $\beta = 10^{-12}$. Since no model reduction step is applied to further reduce the system dimension, the learned model's state dimension is equal to the total number of degrees of freedom in the finite element model, i.e., 120. Since, the output is measured at 6 nodal points, the SIMO ioDMD state-space model in eq. (5.21) has 120 internal states, single input, and six output, and thus the state-space matrices are $\mathbf{A} \in \Re^{120\times120}$, $\mathbf{B} \in \Re^{120\times1}$, $\mathbf{C} \in \Re^{6\times120}$, and $\mathbf{D} \in \Re^{6\times1}$.

ioDMD model training and testing results

The results of modeling the dynamic response of the FEM beam using ioDMD are summarized in Figure 4.2. The data-driven ioDMD model is excited using the same chirp signal used for training the model. For demonstration purposes, among the six outputs, the predicted response output at node 18 ($\hat{y}$) is compared with the measured output from the FEM simulations (y) and is shown in Figure 4.2(a). The low value of the time domain error ($\epsilon_y = y - \hat{y}$), shown in green, in Figure 4.2(a) demonstrates the good quality of the fit. The relative error of $\epsilon_{td}^{rel} = 2.3 \times 10^{-2}$ further substantiates the good quality of the fit across all the 6 outputs in the time domain. For validation purposes, a sine burst at 165.1 Hz is used to excite the ioDMD model and quality of the fit analyzed. The Figure 4.2(b) contrasts the simulated response from the ioDMD methodology with the original response measurement from node 18 in the testing case. The low value of the error plot in Figure 4.2(b) clearly illustrates the validity of the model over the frequency ranges of interest. Figure 4.2(c) and Figure 4.2(d) presents zoomed versions of the training and testing case respectively. The relative time domain error for ioDMD is $\epsilon_{td}^{rel} = 1.62 \times 10^{-2}$ for the testing case.

It is straightforward to recover the dynamic modes of the system under consideration using the developed state-space model. The finite element model is setup in such a way that the dynamic modes of the beam corresponds to the modes of vibration of the system [117]. The recovered modes (ϕ_{ioDMD}), as shown in Figure 4.3(a), closely resemble the modes of vibration of a cantilever beam (ϕ_{FEM}). The quality of the modes recovered using the ioDMD methodology is evaluated using the model assurance criteria (MAC) [146]. If individual columns of ϕ_{ioDMD}, representing the DMD modes, are a close match with that of ϕ_{FEM}, then the MAC value will be close to 1. The value of 1 in the diagonal term in Figure 4.3(b) shows

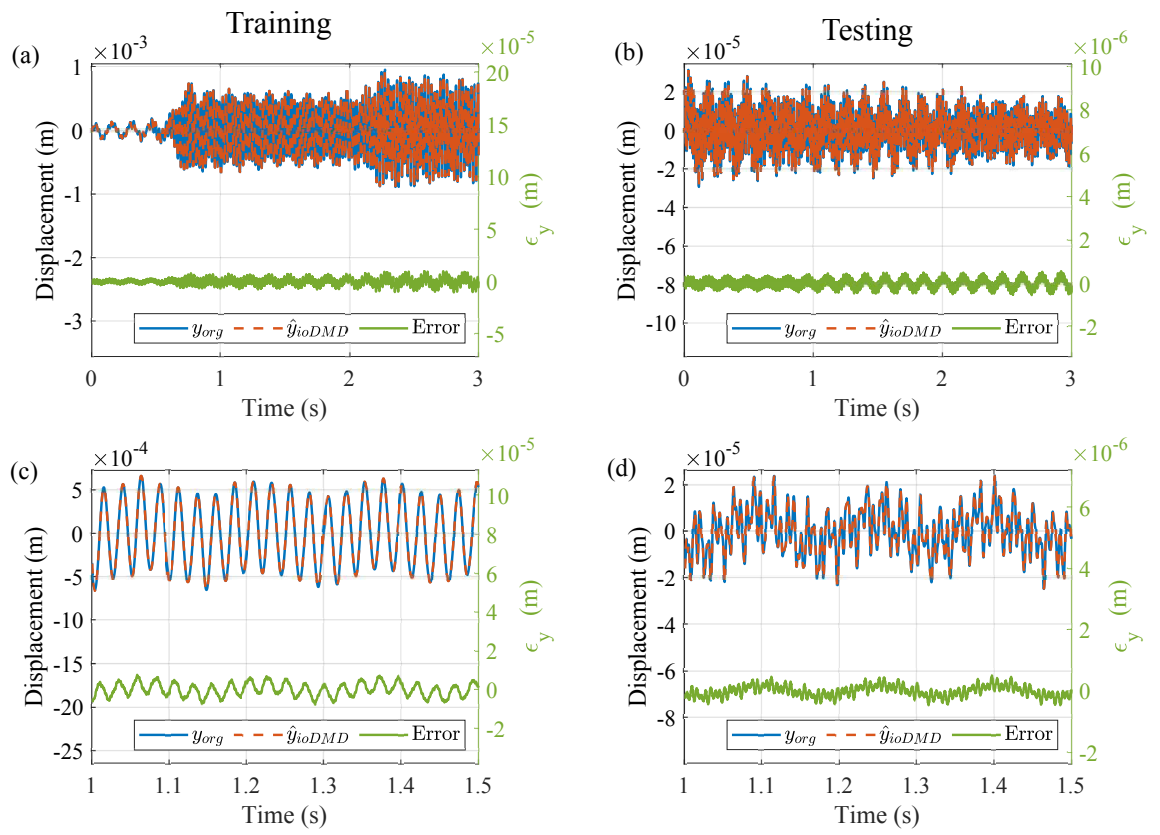


Figure 4.2: (a) Comparison of the predicted response and the original response measured at node 18 when excited with the same training signal (chirp); (b) Comparison of the predicted response and the original response measured at node 18 when excited with a sine burst at 165 Hz (testing data); (c) Zoomed up version of (a) demonstrating a high-fidelity approximation in the training phase (d) Zoomed up version of (b) demonstrating a high-fidelity approximation in the testing phase

that the recovered DMD correlates well with the actual modes of vibration of the system. The zeros in the off-diagonal position further validate the orthogonality of the DMD modes (also a property of the physical modes), thus further demonstrating the efficacy of the ioDMD model in accurately capturing the hidden dynamics of the system under consideration.

4.4.2 Data-driven modeling using WDMD

We now apply the WDMD methodology to model the input-output dynamic responses of the simulated beam. WDMD is used to obtain a data-driven model, using the snapshots matrices of only the measured outputs at $d = 6$ locations and the chirp input, recorded in training phase. It is important to note that WDMD develops a SIMO, data-driven, state-

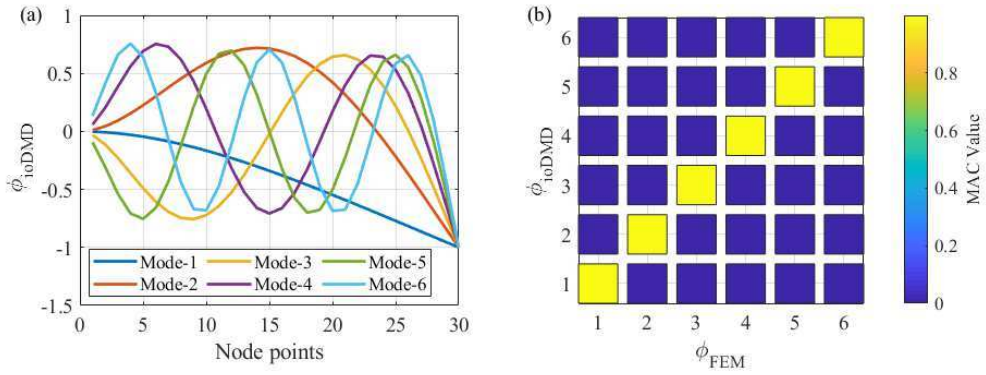


Figure 4.3: (a) Mode shapes extracted by ioDMD (b) Comparison of the ioDMD modeshapes with the analytical mode shapes using modal assurance criterion (MAC)

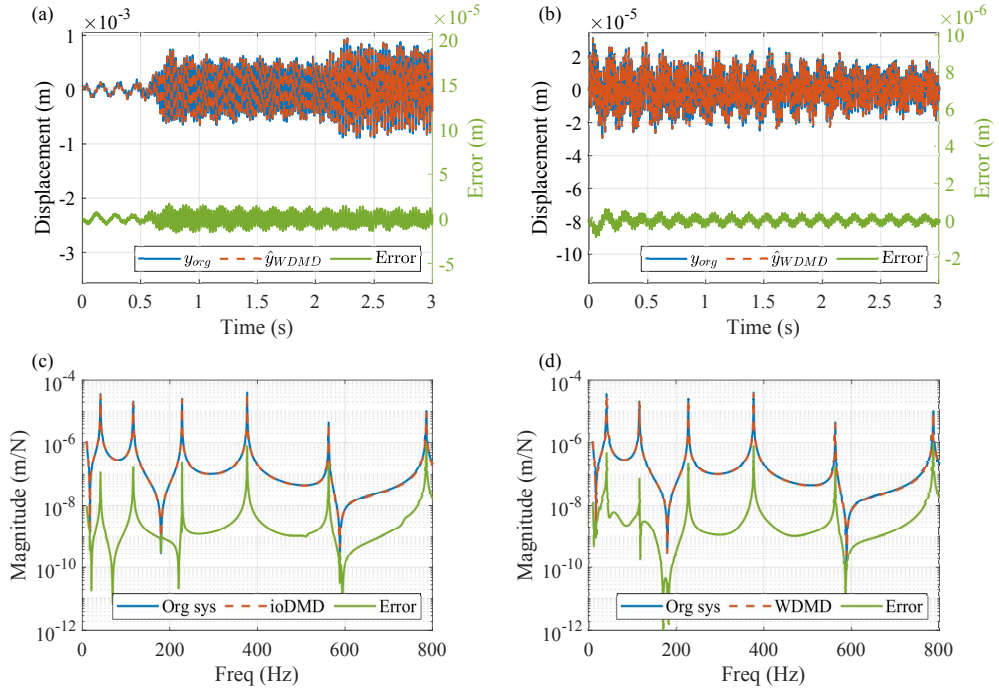


Figure 4.4: (a) Comparison of the predicted response using WDMD and the original response measured at node 18 when excited with the training input, plotted alongside with the error in green; (b) Comparison of the predicted response using WDMD and the original response measured at node 18 when excited with the sine burst input at 165.1 Hz (testing phase), plotted alongside with the error in green; (c) Comparison of the magnitude of analytical and predicted FRF at node 18 using ioDMD alongside with error magnitudes; (d) Comparison of the magnitude of analytical and predicted FRF at node 18 using WDMD alongside with error magnitudes.

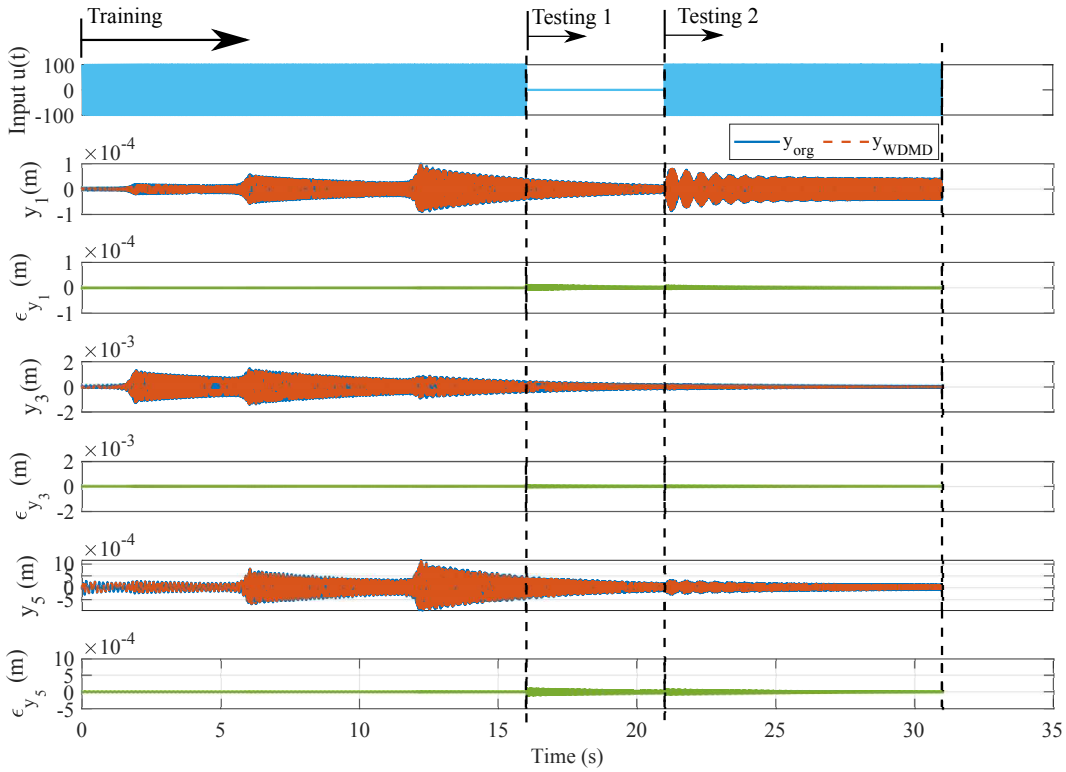


Figure 4.5: Training phase followed by testing phases, comparing the predicted responses from WDMD model with the FEM simulation results.

space model by *only* utilizing the input-output trajectories of the measured nodal points, thus circumventing the restrictive assumption to require the samples of all the latent states of the system. It will be shown in later sections that WDMD yields high-fidelity approximates even with a small number of outputs and the quality of the fit further improves with an increase in number of measured outputs. The parameters controlling the WDMD algorithm are i) the type of wavelet and ii) the level of wavelet decomposition. In the current study, the Haar wavelet [143] is selected as the default setting throughout and the level of decomposition in this section is $J = 13$. As in the ioDMD case, the singular value truncation tolerance in computing the pseudoinverse is set to $\beta = 10^{-12}$. Based on these parameters, the number of auxiliary states in the resulting WDMD model in eq. (5.18) is given by $d \times (J + 1) = 84$. Finally, this results in a state-space matrices with the following dimensions $\mathbf{A_w} \in \Re^{84\times84}$, $\mathbf{B_w} \in \Re^{84\times1}$, $\mathbf{C_w} \in \Re^{6\times84}$, and $\mathbf{D_w} \in \Re^{6\times1}$.

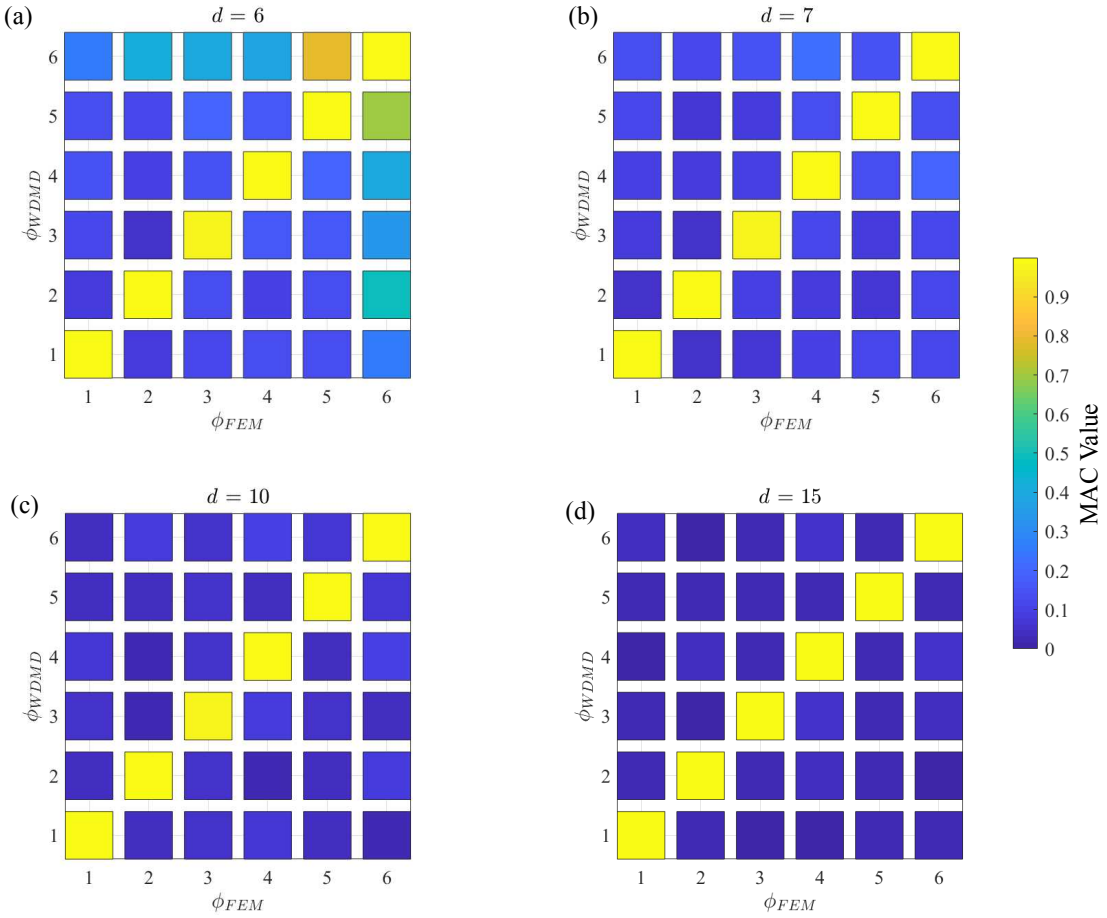


Figure 4.6: MAC comparison plots for WDMD for (a) $d = 6$; (b) $d = 7$; (c) $d = 10$; (d) $d = 15$

WDMD model training and testing results

The results of modeling the dynamic response of the FEM beam using WDMD are summarized in Figure 4.4. For a first comparison, the data-driven model produced by WDMD is used to reproduce the behavior of the system during the training phase. To be consistent, we excite the WDMD model with the same training (chirp) and testing signal (sine burst) as in the ioDMD case. Figure 4.4(a) shows the predicted response at node 18 alongside with the measured output from the FEM simulations, illustrating a high-fidelity match. The low value of the time domain error (green) in Figure 4.4(a) further validates the good quality of the fit. The relative time-domain error of $\epsilon_{td}^{rel} = 1.59 \times 10^{-2}$ across the 6 predicted response illustrates a better performance over ioDMD in this case. The sine burst at 165.1 Hz excites the SIMO WDMD model and the predicted response at node 18 is compared with that of FEM simulation results in Figure 4.4(b). The testing phase results in a relative error of $\epsilon_{td}^{rel} = 7.84 \times 10^{-3}$.

To better illustrate the frequency domain performance, Figure 4.4(c) and Figure 4.4(d) depict, respectively, the magnitude of the predicted FRF ($\widetilde{\mathcal{H}}_{iodmd}(\omega)$) due to ioDMD and predicted FRF ($\widetilde{\mathcal{H}}_{wdmd}(\omega)$) due to WDMD, as compared to the original FRF ($H(\omega)$) measured at node 18. While ioDMD results in a frequency domain relative error of $\epsilon_{fd}^{rel} = 1.51 \times 10^{-2}$, WDMD results in a slightly higher error value of $\epsilon_{fd}^{rel} = 1.63 \times 10^{-2}$. Nevertheless, both WDMD and ioDMD demonstrates excellent capability to capture the frequency domain characteristics of the system. And more importantly WDMD achieves this accuracy by measuring only 6 of the state variables out of the total 120.

To further test the capabilities of the WDMD model, two different testing phases are performed. The first test (Test 1) consists of a time interval with no excitation, thereby allowing the system to be driven by the initial conditions at the end of the training phase. The second test (Test 2) consists of a sine burst at 230.4 Hz. It is pertinent to observe that at no point during the beginning of a phase, the input conditions are corrected. This is particularly challenging for Test 1 where there is no input and thus small deviations in initial conditions can result in high errors. Figure 4.5 shows 3 out of the 6 predicted responses alongside with the error between the WDMD model and the original FEM model. WDMD produces a high-quality fit for the all phases of these tests as can be seen from the low value of the errors, thus illustrating the efficacy of the algorithm.

Similar to the ioDMD case, the quality of the WDMD modes (ϕ_{WDMD}) and their agreement to the physical modes of the beam (ϕ_{FEM}) can be examined by using the MAC plots as shown in Figure 4.6, where d represents the number of outputs measured. The quality of the extracted dynamic modes improves with the increase in total number of outputs measured as seen from Figure 4.6. Nevertheless, even with a small number of measured output ($d = 6$), WDMD resulted in a data-driven model that was able to meaningfully extract modal characteristics of the leading six modes as seen from the diagonal terms in Figure 4.6 (a). At $d = 10$, we see that WDMD results almost converge to the ioDMD MAC plots.

Since WDMD only relies on input-output trajectories at the observed nodes, the major factor affecting the WDMD methodology's efficacy is the total number of outputs measured across the system. This necessitates an error convergence study in the time domain as well as in the frequency domain. By increasing the number of outputs measured, the quality of the fit improves as seen from Figure 4.7. The figure shows the relative error as a function of the total number of measured outputs (d). The figure also provides the ioDMD model error values for comparison purposes. Even with fewer measurements (as few as 6), for this example, the WDMD methodology outperforms the ioDMD methodology. This figure demonstrates the advantage of applying the WDMD in the practical situation wherein only a handful of the output trajectories can be measured. The same is true for the frequency domain representation. The relative error, ϵ_{fd}^{rel} also drops as a function of the number of measurements available but eventually converges to the ioDMD model error of around 0.015.

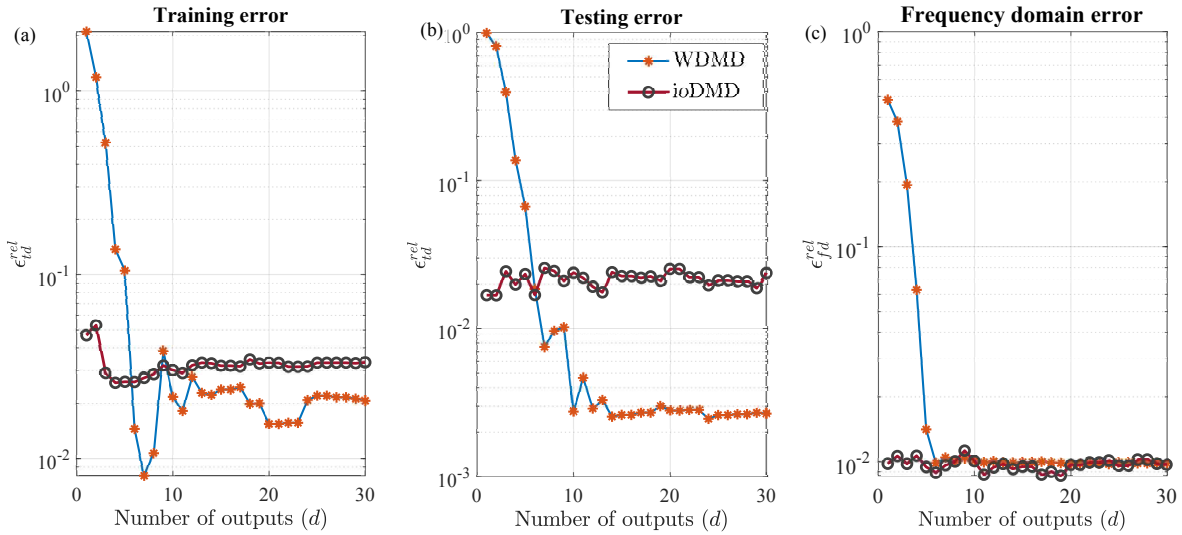


Figure 4.7: Error convergence study results for the SIMO case: (a) Relative time domain error for training phase; (b) Relative time domain error for testing phase; (c) Relative frequency domain error

4.4.3 Results for the multiple input case - WDMD

In this section, a multiple-input multiple-output (MIMO), state-space model using WDMD is developed. Towards this goal, the FEM beam is subjected to uncorrelated input excitations at node 3 and node 17, thereby simulating multiple (two) input excitations. Similar to the SIMO case, WDMD builds a data-driven model using measured outputs at selected nodal points and chirp inputs that are used for excitation, resulting in a MIMO learned model as in eq. (5.18). WDMD uses $J = 13$ as before. Similar to the SIMO case, error convergence studies for the training and testing cases are performed. Figure 4.8(a) and Figure 4.8(b) shows error convergence plots for the training and testing case respectively as the number of outputs change. The WDMD model error converges towards the full ioDMD baseline error for the training case, when the number of outputs (d) measured are greater than 11. However for the testing case with $d > 11$, the WDMD results in a lower testing error compared to the ioDMD model. These results follow the similar pattern to those of the SISO case. The smaller error for WDMD in the testing case for this specific testing input might be due to the frequency content of the signal. Figure 4.8 depicts the error in the frequency domain. Similar to the time domain error, there is no noticeable reduction in the error beyond $d = 10$ and around this value the WDMD error converges to the ioDMD model error. It is important to emphasize that WDMD provides comparable results to ioDMD (and better in the time-domain testing case for this example) with a much smaller number of observed state. In this example, there are a total of 120 state variables with WDMD observing only 10 of them (less than 10% of the total). Thus, WDMD is able to match the full-state observation accurately.

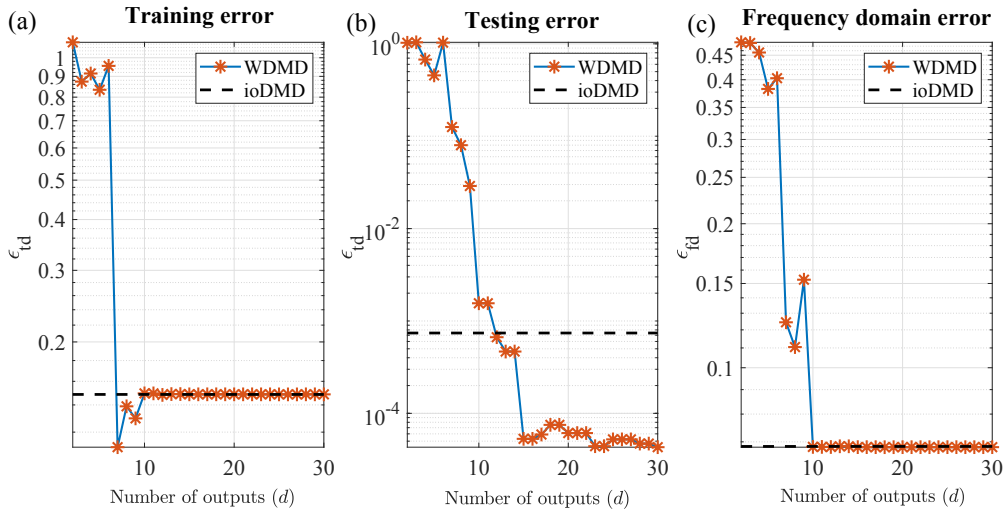


Figure 4.8: Error convergence study results for the MIMO case: (a) Relative time domain error for training phase; (b) Relative time domain error for testing phase; (c) Relative frequency domain error.

4.4.4 Comparison with the Delay-DMD

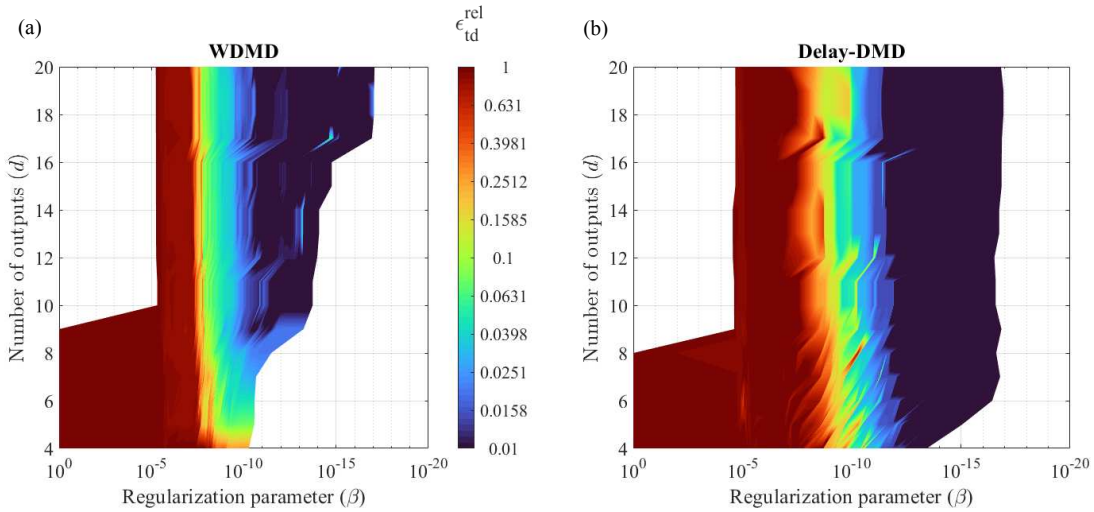


Figure 4.9: Comparison of WDMD with Delay-DMD for the noise-free data using the ϵ_{td}^{rel} measure as β and d vary

In this section, WDMD is compared with Delay-DMD. Delay-DMD can be thought of as a special case of EDMD with the observable vector in eq. (4.15) composed of the time-delayed versions of the measurements as, $\mathbf{\Phi}(\mathbf{x}_k) = [\mathbf{x}(t_k), \mathbf{x}(t_{k-\delta}), ..., \mathbf{x}(t_{k-\delta\tau})]$, where $k = 1, 2, ...K - \delta\tau$ represent the snapshot indices. The integer δ and τ represent the lag-time and the embedding dimension, respectively. As with any algorithm, the performance of the

Delay-DMD depends on various parameter choices, such as the lagtime δ and the embedding dimension τ that are often problem-specific [125, 147]. Discussion regarding the parameter choices and embedding dimensions are beyond the scope of the present study. For comparison purposes with WDMD, δ is chosen as 1 and the embedding dimension is varied on a per case basis [20, 125, 147].

The present comparative study is conducted on the FEM beam model described in Section 4.4 with the same training and testing cases. As discussed in Section 4.4, the performance WDMD depends on the number of available measurements and the decomposition level. Thus, the number of available measurements is varied as one of the parameters.

The input-output trajectories of the FEM beam model are utilized by WDMD and Delay-DMD towards building a SIMO, data-driven, state-space model. For comparison purposes, both WDMD and Delay-DMD model are excited with the same chirp signal to reproduce the system's behaviour during the training phase and the quality of the fit is assessed. Since both of these methods utilize only the measured input-output trajectories, the quality of the fit depends on the total number of measured responses, d. While WDMD results in a state-space model order of dimension $d(J+1)$, Delay-DMD results in a model order of $d \times \tau$. Thus, Delay-DMD can lead to higher model orders even with a small number of measured outputs if the embedding dimension τ is large enough.

As explained in Section 4.2.2, apart from the number of responses measured, the performance of the algorithm also depends on the regularization parameter β. Therefore, for every training data set (i.e., for each value of d), the parameter β is varied in both WDMD and Delay-DMD and the quality of the fit evaluated using ϵ_{td}^{rel}. Therefore, for both WDMD and Delay-DMD the relative error ϵ_{td}^{rel} is plotted in the form of surface contours with d and β being the two axes, as shown in Figure 4.9(a) and Figure 4.9(b), respectively.

In Figure 4.9(a) and Figure 4.9(b), three distinct regions are observed: i) $\beta > 10^{-8}$, ii) $10^{-8} > \beta > 10^{-12}$, and iii) $\beta < 10^{-12}$. For the first region, both the training and testing errors are large for both methodologies, and this is attributed to the higher value of the regularization parameter, thus leading to an oversimplified model with significant singular values being truncated. However, in the second region, both methodologies demonstrate better performance, with WDMD having lower training and testing error compared to Delay-DMD. Finally, in the third region, Delay-DMD shows lower training error compared to the WDMD.

For most practical situations, the signal obtained from the sensors will be corrupted with noise. This warrants repeating the same set of simulations in the presence of added noise. Towards this goal, simulations are realized to study the performance of both methodologies in the presence of added noise. Zero-mean Gaussian white noise with amplitude corresponding to 0.5% of the measured signal is artificially added to all the outputs recorded from the FEM simulations and both methods are repeated. As expected, both methodologies perform poorly for β greater than 10^{-5} as seen in the surface plots in Figure 4.10. However, it is observed from Figure 4.10(a) and Figure 4.10(b) that for lower values of regularization

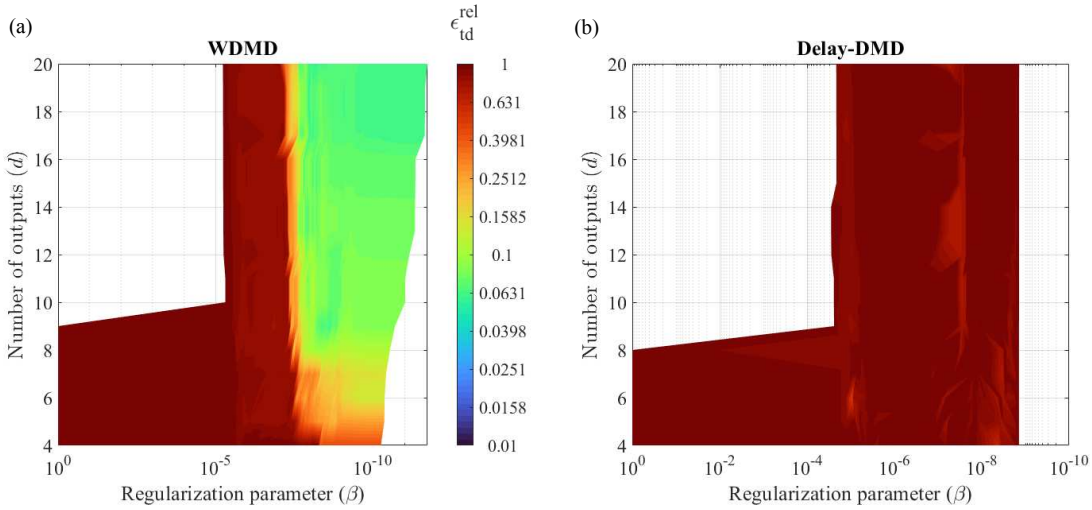


Figure 4.10: Comparison of WDMD with Delay-DMD for the 0.5% additive noise case using the ϵ_{td}^{rel} measure as β and d vary

parameter the training errors in WDMD algorithm are orders of magnitude smaller than Delay-DMD. However, by no means this one numerical example claims to show that in the case of noisy data WDMD performs better in general. Authors acknowledge the fact that the issues observed with Delay-DMD in this one specific example may be resolved by proper tuning of hyper-parameters, even though tuning these parameters for every possible case of noise could be challenging in practice. [42, 125]. The main goal of this numerical study was simply to see how WDMD might behave under noise if it was tuned, i.e., the level of decomposition J, for the noise-free data. It is possible that the smoothing operation inherent to the wavelet observables used in WDMD might naturally help with the noisy data. However, a detailed theoretical analysis of WDMD for the noisy case is beyond the scope of this paper and will be done in a future work.

4.5 Experimental study

We now present an experimental case study to validate the efficacy of the WDMD in modeling the dynamical response of a beam excited by an external forcing. The experimental set up used to measure the time domain response of the beam is depicted in Figure 4.11. A 30 *in.* long aluminum beam with a rectangular cross-section of 1.5 *in.* $\times$ 0.1 *in.* (bxh) has been selected for this study. Free-free boundary conditions are approximated by suspending the beam under test with fishing line wires. Two Macro Fiber Composites (MFCs), model number 29K06-005B, are bonded to either end of the beam for excitation purposes. The MFCs are actuated by supplying a Matlab generated signal delivered using an NI DAQ, and amplified through a power amplifier (Trek PZD350A-2-L). A scanning laser doppler

vibrometer (SLDV), Polytec PSV-400, is used to measure the beam's dynamic response when excited with the MFC's. In the present set of experiments, 67 equally spaced scanning points are defined along the beam's length. The SLDV measures the velocity response of all the scanning points in the beam. The whole assembly has been placed on top of a Newport ST series smart table to isolate the effects of ground vibrations and other random excitations.

Using this setup, two sets of experiments are realized: i) Single-input multiple-output (SIMO) and ii) Multi-input multi-output (MIMO). Similar to the finite element simulations, the beam under study is excited using two sets of inputs for both of these cases: i) Chirp signal over the frequency range 100 - 500 Hz to train the algorithm, and ii) Sine burst to validate the model. The SLDV measures the output response (velocity) at a sampling rate of 5000 Hz from multiple locations along the beam. Measurement locations are densely selected to have enough data to study the effect of number of measurement points on the quality of the fit. In this experimental case study, the input corresponds to the voltage supplied to the MFC, while the output to the measured velocity responses.

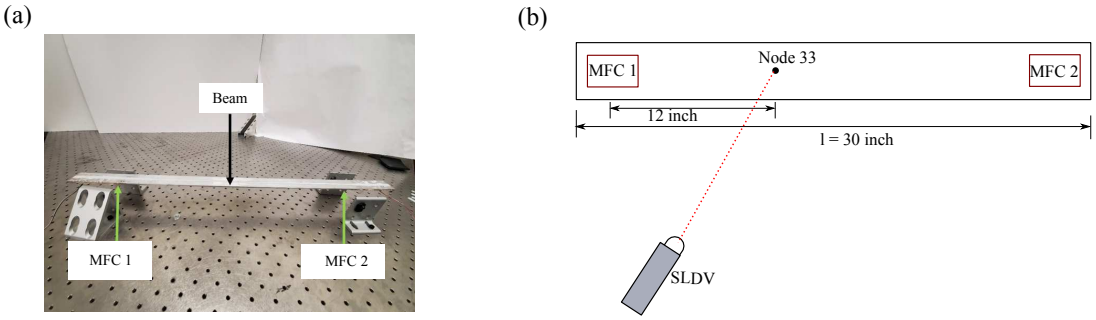


Figure 4.11: Experimental setup used for the study (a) the picture of the free-free beam attached with MFC-1 and MFC-2, (b) schematic displaying the locations of MFC's on the beam and node 33.

4.5.1 Data-driven SIMO model

In this section, a SIMO, data-driven, state-space model is developed using WDMD based on the measured responses along the beam, valid over the frequency range of 100 to 500 Hz. The WDMD methodology utilizes the measured the velocity responses at 10 equidistant points along the beam and chirp input voltage supplied to the MFCs for building the data-driven SIMO model. It is pertinent to note that, although we measured velocity at 66 scanning points in the beam, we utilize only 10 output responses to develop the data-driven model. The level of wavelet decomposition (J) is set as $J = 13$. Hence, WDMD outputs a linear discrete state-space form with the following dimensions: $\mathbf{A_W} \in \Re^{140\times140}$, $\mathbf{B_w} \in \Re^{140\times1}$, $\mathbf{C_w} \in \Re^{10\times140}$, and $\mathbf{D_w} \in \Re^{10\times1}$. The singular value truncation tolerance β is set to $\beta = 10^{-12}$.

Once the data-driven, SIMO, state-space model is developed, the behaviour of the free-free

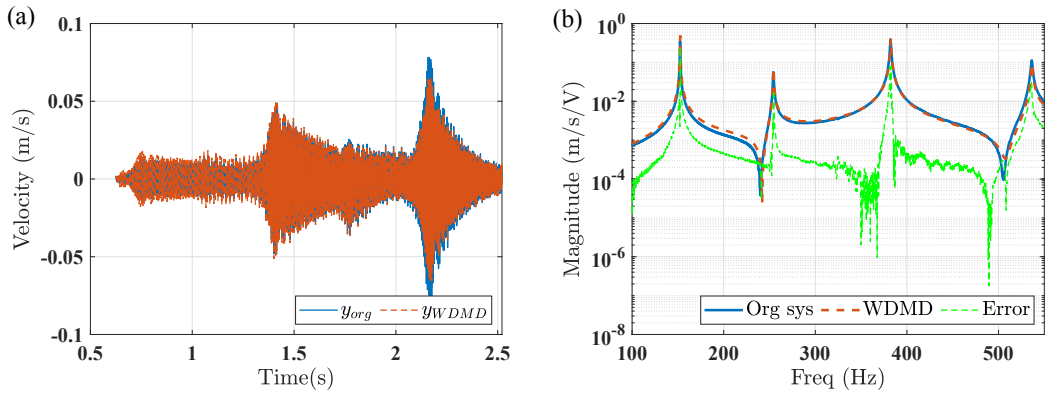


Figure 4.12: Comparison of predicted WDMD response with the actual experimental data in (a) time domain and (b) frequency domain

beam during the training phase is reproduced using the recorded chirp input. For demonstration purpose, the predicted response (dashed orange line) at one of the ten measured locations (nodal point 33) is compared with the SLDV measured velocity (solid blue line) in Figure 4.12(a). Figure 4.12(b) depicts the magnitude of the predicted FRF ($\tilde{\mathcal{H}}_{wdmd}(\omega)$) compared to the experimentally measured FRF ($H(\omega)$) corresponding the same node. The high-fidelity fits in Figure 4.12(a) and Figure 4.12(b) demonstrate the efficacy of the algorithm in accurately reproducing the time domain and frequency domain characteristics of the beam under test. The relative error $\epsilon_{td}^{rel} = 2.17 \times 10^{-1}$ is higher compared to the simulated data cases of the previous section, but this is expected because of the unfiltered experimental noise. Further studies are needed to further improve the robustness of WDMD methodology in cases with high experimental noise. It is important to note that employing ioDMD for the present experimental case study is not feasible since the internal states of the structure under test is unknown.

Similar to Section 4.4, we perform error convergence studies to evaluate the quality of the fit as a function of the number of output responses available to the algorithm. As before, the data-driven model's quality of the fit is evaluated using ϵ_{td}^{rel} and ϵ_{fd}^{rel} in time and frequency domain, respectively. The error convergence study is carried out by sequentially varying the number of outputs provided to the WDMD from 2 to 30. Figure 4.13 shows the error convergence in both time and frequency domains. The relative error metrics ϵ_{td}^{rel} and ϵ_{fd}^{rel} converges at 2.17×10^{-1} and 1.64×10^{-1}, respectively, at $d = 9$ and no further significant improvement in the quality of the fit is observed for $d \geq 9$. Nevertheless, the experimental studies clearly show the efficacy of WDMD methodology in accurately modeling the dynamic response of a beam.

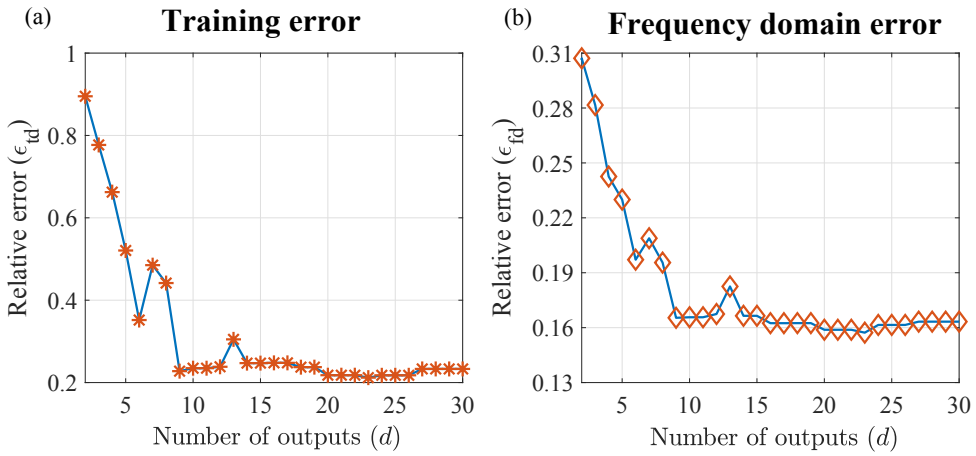


Figure 4.13: Error convergence results for the experimental SIMO case study: (a) Relative time domain error for training phase and (b) Relative frequency domain error

4.5.2 Data-driven MIMO model

The algorithm is now experimentally tested for a MIMO case study using the same free-free beam excited by applying an input voltage to both MFC's simultaneously. The experimental setup and the procedure adopted follows a similar approach to the SIMO case with the only difference being multiple excitations.

Uncorrelated input chirp voltage signals provided to MFC 1 and MFC 2 excite the beam simultaneously. The chirp is designed to have different cycles of frequency sweeping to remove the mode cancellation arising due to correlated input signals. The WDMD methodology builds a MIMO, data-driven, state-space model using the collected response measurements. The learned model has state-space dimensions $\mathbf{A_w} \in \Re^{140\times140}, \mathbf{B_w} \in \Re^{140\times2}, \mathbf{C_w} \in \Re^{10\times140}$ and $\mathbf{D_w} \in \Re^{10\times2}$. The number of columns in matrix $\mathbf{B}$ matrix represents the number of inputs in the system, which in the present MIMO example is two.

As in Section 4.5.1, the MIMO state-space model is excited with the training inputs to reproduce the results of the training phase. For brevity, the time domain and frequency domain fitting results for the model are not shown. Figure 4.14 shows the time domain error ϵ_{td}^{rel}, and the frequency domain ϵ_{fd}^{rel} as a function of the number of outputs recorded and made available to the WDMD methodology. The lowest relative error value recorded is around 0.25, which is slightly higher than the SIMO case. Similar is the case for ϵ_{fd}^{rel}, which has a lowest recorded value of 0.19, which is slightly higher than the SIMO case.

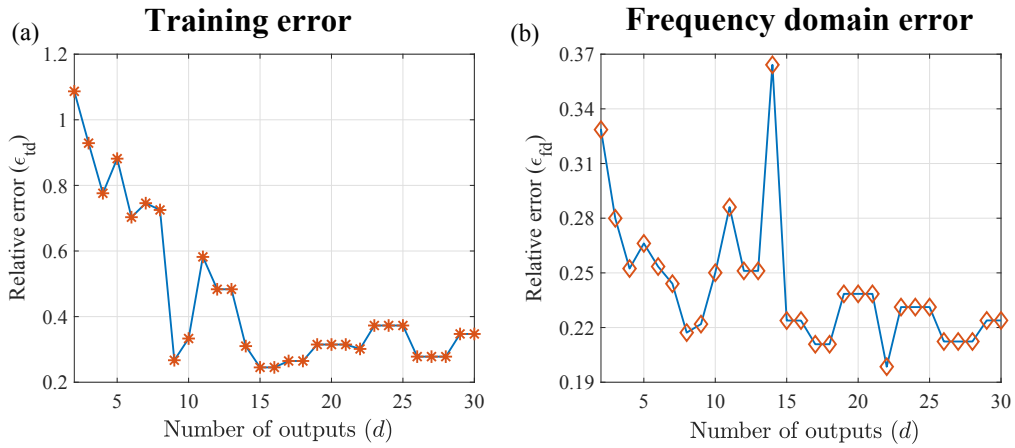


Figure 4.14: Error convergence results for the experimental MIMO case study: (a) Relative time domain error for training phase and (b) Relative frequency domain error

4.6 Conclusions and future work

The current study presented a novel data-driven methodology to model dynamical systems from its input-output trajectories, without having access to governing physical equations or full internal state dynamics. This was achieved using wavelets in conjunction with the ioDMD approach, leading to the proposed methodology, wavelet based DMD (WDMD). The numerical case study involving the dynamical response of a finite element cantilever beam was performed to demonstrate the effectiveness of WDMD. WDMD was utilized to develop a data-driven SIMO state-space dynamical model of the FEM beam based on measured input-output response. The WDMD methodology utilizes a subset of these measurements and approximates the underlying dynamics via a linear model using the maximal overlap discrete wavelet transform (MODWT) coefficients of the measured outputs as the auxiliary state-vector. The error convergence studies illustrated that even with a few measured outputs, WDMD was able to model the underlying dynamical system accurately. The experimental case study on a simple free-free beam, demonstrated the efficacy of WDMD methodology as an appropriate candidate for modeling practical dynamical systems despite having no access to internal state measurements.

The work presented herein demonstrates the feasibility of approximating the input-output dynamics of a vibrating beam based on measured input-output data using this new data-driven modeling approach. Although the WDMD algorithm performed reasonably in presence of noise, as demonstrated in Section 4.5, further studies into sensor placement, input excitation requirements and hyper-parameter selection are required to improve the robustness of the WDMD algorithm in the presence of higher noise level.

Acknowledgements

The authors would like to thank Drs. Benner, Himpe, and Mitchell for providing access to their input-output DMD code that has provided the foundation for building all the codes used for the current research. The authors acknowledge the support received through the Rolls Royce Fellowship and John R. Jones III Graduate Fellowship. Dr. Tarazaga acknowledges the support received through the John R. Jones III Faculty Fellowship. Gugercin was supported in parts by the National Science Foundation under Grant No. DMS-1819110.

Chapter 5

Wavelet-based Dynamic mode decomposition for tracked order vibration modeling in turbofan engine

The current study explores a recently developed data-driven modeling technique known as wavelet based dynamic mode decomposition (WDMD) for modeling tracked order vibration in turbofan engines. valid over various operating conditions. Due to the absence of a representative physics-based model tying together various operational conditions, significant challenges exist in interpreting the tracked order vibrations. This causes significant downtime and loss of revenue for the engine manufacturer. We demonstrate the efficacy of the proposed methodology by modeling the 1^{st} order amplitudes of a finite element beam with an imbalanced rotor. We also validate the performance of the WDMD algorithm in the presence of noise. In the latter sections, we present a data-driven strategy to model tracked order amplitude of a turbofan engine using a switching linear dynamical system comprising of various locally valid WDMD models. Finally, we compare the performance of the WDMD models with that of the state-of-the-art time-delay embedded dynamic mode decomposition (Delay-DMD).

The contents of this chapter are reproduced from, ***M. Krishnan, I. Sever, S. Gugercin, and P. A. Tarazaga, "Wavelet-based dynamic mode decomposition for tracked order vibration modeling in turbo-fan engine"****, which is currently under internal intellectual property review by Rolls-Royce.*

5.1 Introduction

As with all turbo machinery, aircraft turbofan engines are subjected to periodic excitation when they are in operation [2, 148]. The rotational components inside the engine, such as shafts, are the primary source of these periodic excitations. Vibrations due to these periodic excitations are of interest because they can become large without resonance amplifications. Frequency components of this vibration signal which are integer or non-integer multiples of rotational speed, are defined as orders [27, 148, 149]. During an engine's operation, these frequency components or harmonics are continuously monitored and analyzed. Depending upon the operating regime and loading conditions, the amplitude of these frequency components demonstrates high variability, making their interpretation challenging and time-consuming [8, 9, 10, 97]. The lack of suitable physics-based models for tracked order amplitudes, unifying the concepts of operational and loading conditions, further exacerbates the problem [7].

A rotatory machine's vibration response is shaped by : (i) the rotational frequency of the machine, and (ii) the system parameters such as natural frequency and damping [7]. Conventional modal analysis techniques such as experimental modal analysis (EMA) [4] and operational modal analysis (OMA) [3] are not directly applicable for rotating machines because of the presence of harmonics in the output vibration signal that are tied to the fairly consistent inputs. This led to the development of the order tracking (OT) method, which became the industry standard to analyze rotatory machine vibration signals [1, 148, 149, 150, 150]. The OT methods decompose the measured vibration signal into harmonic components related to the rotational frequency of the machine, thereby determining the critical rotational speeds that excite the structural resonances. The vibration signals for order analysis are recorded during the machine run-up or coast-down operation. When plotted against their corresponding rotations per minute (RPM), the response curves of different engine orders are known as order spectrum [148]. The order spectrum is widely adopted in the rotating machinery industry for vibration monitoring, system identification [6], and baselines for anomaly detection [6]. However, vibration monitoring of aircraft turbofan engines using OT methodology has an inherent disadvantage. It focuses only on shaft orders without explicitly considering other operational variables such as temperature, pressure, and altitude [8, 9]. Thus, relying on extracted order amplitudes without accounting for operational and environmental conditions can lead to erroneous judgment and higher testing times. If there were accurate physics-based models to predict orders given any operating conditions, these issues can be circumvented. However, developing accurate physics-based models is not always feasible or straightforward. In such situations, are data-driven models a viable alternative? Could they provide important information to develop physics-based models.

Data-driven modeling provides an effective alternative approach for capturing the dynamic response of a turbofan engine. Such models rely entirely on measured input-output responses; thus, any knowledge of the underlying physical characteristics of the engine is not required. Data-driven modeling literature can be broadly divided into two categories based on the type

of training data: (i) time-domain methods and (ii) frequency-domain methods. In the context of frequency-domain data-driven models, the training data amounts to a set of frequency response measurements, and the modeling amounts to constructing a low-dimension rational function to fit these data: see, e.g., [11, 12, 136, 151, 152]. Machine learning techniques such as Neural networks [153], recurrent neural networks (RNN) [154], and Long short-term memory (LSTM) [65, 155] are at the forefront of time-domain methods. However, they lack physical interpretability and have the problem of vanishing gradients. In the previous decade, the operator theoretic framework based on the Koopman operator [19] has garnered interest in the dynamical system community [20, 21, 22, 42]. The most popular of which is the Dynamical Mode Decomposition (DMD) [20], owing to its ability to decompose high dimensional fluid flow data into its spatio-temporal structures [22, 23]. Over the past few years, various extensions of DMD have been proposed to adapt the DMD algorithm towards various applications. For example, DMD with control (DMDc) [36] and input-output DMD (ioDMD) [37] extends the classical DMD approach for externally forced systems. Optimized DMD [121] and sparsity promoting DMD [122] improves the performance of DMD in the presence of noise. Similarly, Kernel based DMD, and Extended DMD (EDMD) [92] extends the DMD algorithm towards fitting a nonlinear system. For an exhaustive review, we refer the readers to [20]. Most variants of DMD require high-dimensional state data and are primarily used for fluid flow problems where such data is plentiful. However, for systems evolving on a low dimension manifold such as mechanical systems, variants of DMD known as time-delay DMD have found successful application [24, 25, 26].

In prior work, we proposed a novel wavelet-based dynamic mode decomposition (WDMD) methodology that utilizes wavelets in conjunction with input-output DMD (ioDMD) to circumvent the high dimensional state requirement of DMD [21]. We demonstrated the algorithm using a numerical and experimental case study and illustrated the efficacy of WDMD in approximating a linear dynamical system that can accurately predict the dynamic response of a structure, given an excitation input. One of the significant findings of the previous effort was that the quality of the data-driven model generated by WDMD with just 5% of the state information is almost comparable to the ioDMD results. WDMD also functioned satisfactorily in the presence of low noise. Therefore in the present work, we utilize the WDMD framework to model the first-order response of a turbofan engine for a range of operating environments. To build the data-driven framework, we will use the pass-off tests sensor data that all commercial turbofan engines must undergo before entering into service. The significant contributions of this study are as follows: First, we demonstrate the WDMD approach through a numerical case and evaluate the quality of the data-driven model at various levels of signal-to-noise ratio. Secondly, we introduce a data-driven framework to predict tracked orders for a wide range of operational conditions for a real turbofan engine. Finally, we apply the novel WDMD framework to predict the tracked orders at each operational condition and compare and validate its performance with the state-of-the-art time-delay DMD.

The paper is organized as follows. First, we present a brief background of the traditional order tracking methodology and stress the importance of a data-driven model using case scenarios

from typical pass-off tests. Second, we provide an overview of the important concepts used in the present work, such as DMD, ioDMD, and WDMD, to better help readers. The proposed WDMD methodology is demonstrated with a numerical example of an imbalanced rotor on top of a finite element beam. The final section applies the WDMD on the engine data-set and compares its performance with delay-DMD.

5.2 Engine model and order analysis

Rotatory machines produce repetitive vibrations while operated. When subjected to changes in rotational speed, these vibrations signals are composed of non-stationary components and applying standard vibration analysis is challenging. Hence, for rotatory machines, a measurement technique called order analysis is commonly utilized to decompose the vibration signal into its various components, which significantly simplifies the analysis. Order tracking refers to a family of signal processing techniques to convert the measured time-domain signals to revolution (or angular domain). Consecutively, the order tracking methodology extracts frequency components related to the rotational speed of the machine (or harmonics) from the measured vibration signal. The outcome of order analysis is the amplitude of the Fourier transform corresponding to those frequency components. Theoretically, we can write a single order in the following form,

$$\psi(t) = Z_p(t)e^{i2\pi\frac{p}{T}t}, \tag{5.1}$$

where $\psi(t)$ is the measured single order signal, $Z_p(t)$ is a complex number representing the amplitude and phase of the p^{th} order, and T is the time period of the primary order (in sec). For a rotatory machine, several orders are present due to several mechanical components and their harmonics, and hence, the measured vibration signal, $x(t)$ can be represented as the sum of several orders as,

$$x(t) = \sum_{n=0}^{n=\infty} Z_p(t)e^{i2\pi\frac{p}{T}t}. \tag{5.2}$$

The main objective of order tracking methods is to split the response $x(t)$ into its sinusoidal components and estimate $Z_p(t)$. If the rotational speed is kept constant, then these components also appear at a constant frequency. However, during run-ups or coast down, when the rotational speed of the machine is continuously changing, these orders span a specific frequency band, depending on the rotational speed of the machine [7, 148]. Hence, the order time series signals finds applications ranging from imbalance detection [156] to bearing fault detection [157] and system identification [27]. In terms of fault detection, specific harmonic components or orders are associated with certain machine faults [158, 159, 160]. For example, the first order (harmonic) usually relates to imbalance of the machine, and if there are n blades attached to a shaft, higher energy in n^{th} harmonic relates to errors on the blades. In most industrial turbines, the orders of interest are the integer multiples of the rotational speed of the shaft [97]. Non-integer orders can also be of interest, especially in the context

of identifying bearings and gearbox fault signatures [161].

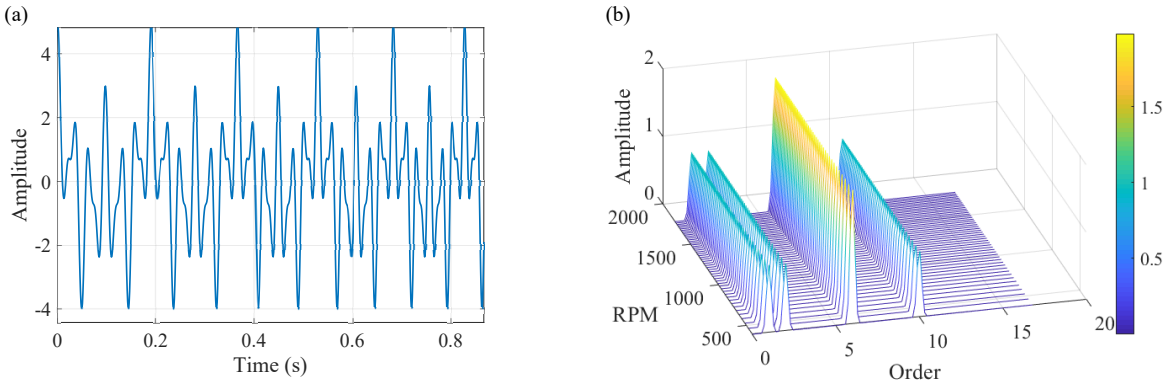


Figure 5.1: A simple order analysis example : (a) Synthetic signal comprising of orders $1, 2, 6$, and 10, (b) RPM-order map of the synthetic signal.

To provide a simple example of the order analysis procedure, we generate a synthetic signal $x(t)$ consisting of orders $1, 2, 6$, and 10. We illustrate the synthetic signal time series in Figure 5.1 (a). Waterfall plots, as shown in Figure 5.1 (b), are the most common way to visualize the order spectrum. The waterfall plot has frequency along the x-axis, time or RPM along the y-axis, and amplitude in the z-axis [7]. For the present example of the synthetic signal, all order amplitudes are kept constant as seen from Figure 5.1 (b). However, for a practical system such as a turbofan engine these amplitudes continuously vary due to the system natural responses and deviation in the operational environment. There are several methods in the literature for estimating the order in terms of amplitude and phase, such as time-domain sampling-based FFT [162], resampling-based order tracking [163], and Vold-Kalman filter-based order tracking [149]. All these methods rely on the accurate estimation of instantaneous rotational speeds. There are several advantages and disadvantages associated with each, and we refer interested readers to [27] for further information.

In the present case study, we focus on the fundamental shaft order (i.e., order = 1). Figure 5.2 (a) shows the typical pass-off RPM curve associated with a turbofan engine. Using the techniques above, we obtain the order 1 amplitude corresponding to Figure 5.2 (a). Figure 5.2 (b) shows the order 1 amplitude alongside the temperature measured at one of the station points inside the engine. To protect the confidentiality of the data, we provided pseudo-names to all the sensors mentioned in the study and normalized all the axes. We note from Figure 5.2 (b) that tracked order amplitude's variation at a steady RPM value has temperature dependency. Since the main goal is to build a data-driven model for 1^{st} order amplitudes, capturing this interdependency and dynamics is critical to have a good quality of the fit for the model. We direct interested readers to [8, 164, 165] for more details regarding variable interdependency. As mentioned before, the tracked order curves plotted against its corresponding shaft RPM (herein named as characteristic curves) functions parallel to the standard frequency response curve. The characteristic curves find application in the

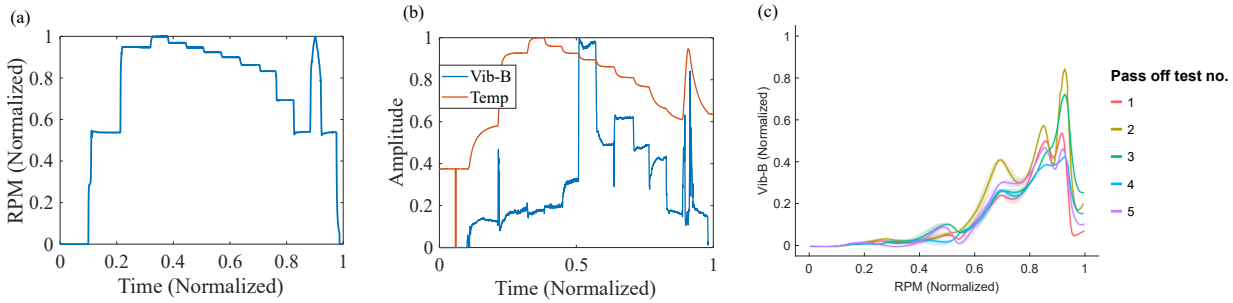


Figure 5.2: Traditional pass-off tests performed on turbofan engines. (a) Set of maneuvers performed on the engine illustrated in the form of RPM changes. (b) Time series of the Vib-B (tracked order) plotted alongside evolution of the temperature inside second stage of the engine during the pass-off tests, and (c) Characteristic curve (tracked order vs RPM) for different runs of the same engine.

domain of system identification, anomaly detection, and modal analysis. Figure 5.2 (c) shows the tracked order values plotted against its corresponding shaft RPM values for an engine undergoing several pass-off tests over a few days. The plot further illustrates the variability in tracked order amplitudes from one run to another. This variability can be attributed to the changes in the initial operating conditions and the loading [10, 97]. The characteristic curves also point to resonance peaks in the normalized speed ranges of $0.8-0.85$ and $0.94-0.965$. In our current effort, we provide a data-driven methodology involving our recently developed WDMD [21] to model the order 1 amplitudes of a turbofan engine under various operating conditions.

5.3 Methodology

In this section, we provide a brief background about the recently developed WDMD [21]. One of the major limitations of the classical DMD approach is that it requires full-state measurements. In a prior work by the authors, this limitation in DMD was circumvented by utilizing the maximal overlap discrete wavelet transform (MODWT) coefficients of the measured output responses as the states in ioDMD, thereby enriching the states.

5.3.1 Wavelet based dynamic mode decomposition

In the present section, we briefly discuss the WDMD methodology, which was previously developed in a prior study [21]. In the same study, we provided the mathematical background and connected it to the Extended DMD theory. We also demonstrated the efficacy of the WDMD methodology by modeling the input-output behaviour of a simple FEM beam and

presented an experimental case study involving a free-free beam, with excellent accuracy. The present subsection aims to provide a brief background about discrete wavelet transform (DWT) and maximal overlap discrete wavelet transform (MODWT), and we direct interested readers to the seminal works of [143, 144, 145] for further details.

Wavelet theory and the maximal overlap discrete wavelet transform

The wavelet transform is a time-frequency signal decomposition technique that decomposes a measured signal into different frequency ranges by dilating (scaling) and translating a single function named the mother wavelet. Mathematically, the wavelet transform is represented as the convolution of a signal $y(t)$ with a mother wavelet function $\psi(t)$, as follows,

$$w_k^j(y) = \langle y, \psi_k^j \rangle = \left\langle y(t), \frac{1}{2^{j/2}} \psi \left(\frac{t}{2^j} - k \right) \right\rangle = \frac{1}{2^{j/2}} \int_{-\infty}^{\infty} y(t) \psi^* \left(\frac{t}{2^j} - k \right) dt, \tag{5.3}$$

where w_k^j represents the wavelet coefficient corresponding to j^{th} scaling and k^{th} translational level, and $\psi_k^j = \frac{1}{2^{j/2}} \psi \left(\frac{t}{2^j} - k \right)$ is the dialated and translated mother wavelet. The discrete wavelet transform (DWT) is a special case of (5.3), where w_k^j is computed at each dyadic value of 2^j and translations of k.

According to the algorithm proposed by Mallat [143], DWT replaces the integration procedure in (5.3) by high (h) and low-pass (g) filters and a down-sampling step [143]. This operation divides the frequency-band of the original signal into high and low-frequency components known as wavelet detail and approximation coefficients, respectively. The recursive adoption of these filters by feeding the down-sampled, low-pass filter output into another pair of identical high and low-pass filter pair, decomposes the signal into wavelet detail and approximation coefficients corresponding to various scales.

The maximal overlap discrete wavelet transform (MODWT) is a variant of the DWT that performs the same operations as that of DWT with the exception of eliminating the down-sampling step. Thus, MODWT has the advantage that the resulting details and scale coefficient at each level has the same length as that of the original signal. This is highly beneficial for multi-resolution applications, where the time-invariance property is desired. Hence, application of MODWT on an original signal provides a decomposition of the original signal based on frequency-bands, where individual decomposed components have the same length as that of the original signal. Defining a time-series vector,

$$\boldsymbol{y} = \{y(t_0), y(t_1), \ldots, y(t_{K-1})\}^T \in \Re^K, \tag{5.4}$$

and decomposing $\boldsymbol{y}$ into J levels using MODWT involves successive application of J pairs of high-pass ($\widetilde{h}_{j,l}$) and low-pass filters ($\widetilde{g}_{j,l}$), where $j = 1, 2, \ldots, J$. This procedure yields a

set of wavelet and scaling coefficients at each level j as,

$$\widetilde{W}_{j,t_k} = \sum_{l=0}^{K-1} \widetilde{h}_{j,l} y(t_{k-l \bmod K}), \qquad \widetilde{V}_{j,t_k} = \sum_{l=0}^{K-1} \widetilde{g}_{j,l} y(t_{k-l \bmod K}). \tag{5.5}$$

One can write (5.5) in matrix notation as,

$$\widetilde{\mathbf{W}}_j = \widetilde{\mathcal{W}}_j \boldsymbol{y} \quad \text{and} \quad \widetilde{\mathbf{V}}_j = \widetilde{\mathcal{V}}_j \boldsymbol{y}, \tag{5.6}$$

where the matrix $\widetilde{\mathcal{W}}_j \in \Re^{K\times K}$ and $\widetilde{\mathcal{V}}_j \in \Re^{K\times K}$ is composed of $\widetilde{h}_{j,l}$ and $\widetilde{g}_{j,l}$, respectively. We refer the interested reader to [21] for further details. The original time series $\boldsymbol{y}$ can be written in terms of its MODWT coefficients as,

$$\boldsymbol{y} = \sum_{j=1}^{J} \widetilde{\mathcal{W}}_j^{\mathrm{T}} \widetilde{\mathbf{W}}_j + \widetilde{\mathcal{V}}_j^{\mathrm{T}} \widetilde{\mathbf{V}}_j = \sum_{j=1}^{J} \widetilde{\mathbf{D}}_j + \widetilde{\mathbf{S}}_J \quad \text{, where } \widetilde{\mathbf{D}}_j := \widetilde{\mathcal{W}}_j^{\mathrm{T}} \widetilde{\mathbf{W}}_j \in \Re^K \text{ and } \widetilde{\mathbf{S}}_j = \widetilde{\mathcal{V}}_j^{\mathrm{T}} \widetilde{\mathbf{V}}_j \in \Re^K. \tag{5.7}$$

The last equation provides a means to represent the original signal as a summation of its high frequency and low frequency component. This is utilized in the concept of observables, which will be explained in the next subsection.

Main approach

In this section, we borrow some discussions from our previous work [21] to briefly demonstrate the idea behind WDMD methodology. Let us consider a dynamical system of the form,

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \tag{5.8}$$

where $\mathbf{x} \in \Re^N$ is the underlying state, $\mathbf{u} \in \Re^m$ is the input driving the system, and $f : \Re^N \to \Re^N$ is the evolution function. We consider cases where we do not have access to the full-state samples $\mathbf{x}(t_k)$. This is in sharp contrast to the ioDMD and DMD methods which assume full-state measurements [20, 21]. Rather, the assumption is that we only have access to sparse measurements $\mathbf{y}(t) \in \Re^p$ of the state-vector via an observation matrix $\mathbf{C} \in \Re^{p\times N}$ as follows,

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t). \tag{5.9}$$

We assume that snapshots of this measurement vector is available as,

$$\mathbf{Y}_0 = [\mathbf{y}(t_0), \mathbf{y}(t_1), \ldots, \mathbf{y}(t_{K-1})] \in \Re^{p \times K}. \tag{5.10}$$

As explained in our previous work [21], the main aim of WDMD is to create new pseudo-state variables using the MODWT analysis explained in Section 5.3.1. Therefore, using (5.7) one can still relate the new auxillary state vectors created using the MODWT analysis to the true

output samples measured. We can then apply ioDMD using the snapshots of the original inputs and outputs and the new pseduo-state vectors formed using MODWT coefficients.

Assuming p measurements are available, let $y_i(t)$, where $j = 1, 2, \ldots, p$ denote the i^{th} component of the measurement vector $\mathbf{y}(t)$. We can define the snapshot vector of $y_i(t)$ as $\boldsymbol{y}_i = [y_i(t_0), y_i(t_1), ..., y_i(t_{K-1})]^T$, and further write the decomposition of $\boldsymbol{y}_i$ using MODWT as shown in (5.7),

$$\boldsymbol{y}_i = \sum_{j=1}^{J} \widetilde{\mathbf{D}}_j^{(i)} + \widetilde{\mathbf{S}}_J^{(i)}, \tag{5.11}$$

where $\widetilde{\mathbf{D}}_j^{(i)}$, $\widetilde{\mathbf{S}}_j^{(i)} \in \Re^K$ represents the j^{th} level wavelet detail and smooth coefficients corresponding to $y_i(t)$, respectively. We define $\mathbf{e}_{k+1} \in \Re^K$ as the $(k+1)$st canonical vector and $\mathbf{e} = [1\ 1 ...\ 1]^T \in \Re^{J+1}$ as the vector of ones. This would entail writing $y_i(t_k)$ (the $(K+1)$st row of $\boldsymbol{y}_i$) using (5.11) as,

$$y_i(t_k) = \mathbf{e}_{k+1}^T \boldsymbol{y}_i = \sum_{j=1}^{J} \mathbf{e}_{k+1}^T \widetilde{\mathbf{D}}_j^{(i)} + \mathbf{e}_{k+1}^T \widetilde{\mathbf{S}}_J^{(i)} = \mathbf{e}^T \mathbf{w}_i(t_k), \qquad \text{where } \mathbf{w}_i(t_k) = \begin{bmatrix} \mathbf{e}_{k+1}^T \widetilde{\mathbf{D}}_1^{(i)} \\ \mathbf{e}_{k+1}^T \widetilde{\mathbf{D}}_2^{(i)} \\ \vdots \\ \mathbf{e}_{k+1}^T \widetilde{\mathbf{D}}_J^{(i)} \\ \mathbf{e}_{k+1}^T \widetilde{\mathbf{S}}_J^{(i)} \end{bmatrix} \in \Re^{J+1}. \tag{5.12}$$

Using (5.12) we can write the full measurement vector $\mathbf{y}(t_k)$ in (5.9) as,

$$\mathbf{y}(t_k) = \begin{bmatrix} y_1(t_k) \\ y_2(t_k) \\ \vdots \\ y_p(t_k) \end{bmatrix} = \begin{bmatrix} \mathbf{e}^T \mathbf{w}_1(t_k) \\ \mathbf{e}^T \mathbf{w}_2(t_k) \\ \vdots \\ \mathbf{e}^T \mathbf{w}_p(t_k) \end{bmatrix}. \tag{5.13}$$

Let us define a new pseudo-state vector $\boldsymbol{\phi}$(t) and the observation matrix $\mathbf{C}_\phi$ as,

$$\boldsymbol{\phi}(t_k) = \begin{bmatrix} \mathbf{w}_1(t_k) \\ \mathbf{w}_2(t_k) \\ \vdots \\ \mathbf{w}_p(t_k) \end{bmatrix} \in \Re^{p(J+1)\times 1} \quad \text{and} \quad \mathbf{C}_\phi = \begin{bmatrix} \mathbf{e}^T & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{e}^T & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \mathbf{e}^T & 0 \\ 0 & 0 & \cdots & 0 & \mathbf{e}^T \end{bmatrix} \in \Re^{p\times p(J+1)}. \tag{5.14}$$

Hence using (5.14), we write (5.9) as follows,

$$\mathbf{y}(t_k) = \mathbf{C}_\phi \boldsymbol{\phi}(t_k). \tag{5.15}$$

In (5.15), the actual measurement vector $\mathbf{y}$(t) is written in terms of its state variable $\boldsymbol{\phi}(t)$,

which is composed of its MODWT coefficients. In other words, our task is reduced to finding the evolution operator of the new observables $\boldsymbol{\phi}(t)$. This can be achieved using the ioDMD explained in Section 4.4.1. We provide a brief summary of the WDMD algorithm in the next paragraph.

Assuming, from the underlying dynamical system defined in (5.8), we only have access to input $\mathbf{u}(t_k)$ and output $\mathbf{y}(t_k)$ snapshots. We define,

$$\mathbf{U_0} = \begin{bmatrix} \mathbf{u}(t_0) & \mathbf{u}(t_1) & \dots & \mathbf{u}(t_{K-1}) \end{bmatrix} \in \Re^{m\times K} \text{ and } \mathbf{Y}_0 = \begin{bmatrix} \mathbf{y}(t_0) & \mathbf{y}(t_1) & \dots & \mathbf{y}(t_{K-1}) \end{bmatrix} \in \Re^{p\times K}. \tag{5.16}$$

Using (5.11) - (5.14), we can construct the snapshot matrix of the observables $\boldsymbol{\Phi}$ as follows,

$$\boldsymbol{\Phi}_0 = [\boldsymbol{\phi}(t_0) \;\; \boldsymbol{\phi}(t_1) \;\; \dots \boldsymbol{\phi}(t_{K-1})] \in \Re^{d(J+1)\times(K)} \text{ and } \boldsymbol{\Phi}_1 = [\boldsymbol{\phi}(t_1) \;\; \boldsymbol{\phi}(t_2) \;\; \dots \boldsymbol{\phi}(t_K)] \in \Re^{d(J+1)\times(K)} \tag{5.17}$$

It is pertinent to note that the input and output snapshot matrices in (5.16) corresponds to the actual inputs and outputs of the underlying dynamical system in (5.8). However, the pseudo-state snapshots $\boldsymbol{\Phi}$ in (6.21) are obtained via the MODWT coefficient vector. Thus, WDMD represents the dynamical system of the form,

$$\boldsymbol{\phi}(t_{k+1}) \approx \mathbf{A}_\phi(t_k)\boldsymbol{\phi}(t_k) + \mathbf{B}_\phi\mathbf{u}(t_k), \quad \mathbf{y}(t_k) \approx \mathbf{C}_\phi\boldsymbol{\phi}(t_k) + \mathbf{D}_\phi\mathbf{u}(t_k). \tag{5.18}$$

The above mentioned equation is a reformulated form of the input-output mapping via a MODWT coefficient observable vector, and this allows to apply the ioDMD in Section 4.4.1 to construct the state-space matrices $\mathbf{A}_\phi$, $\mathbf{B}_\phi$, $\mathbf{C}_\phi$, and $\mathbf{D}_\phi$ via a least-squares approximation. These matrices are given by,

$$\begin{bmatrix} \mathbf{A}_\phi & \mathbf{B}_\phi \\ \mathbf{C}_\phi & \mathbf{D}_\phi \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \mathbf{Y}_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_0 \\ \mathbf{U}_0 \end{bmatrix}^+, \tag{5.19}$$

where "+" denotes the pseudo-inverse. For further information and algorithmic implementation, we refer interested readers to [21].

5.4 Numerical case study using switching linear dynamical system (SLDS)

This section presents a numerical example to illustrate the applicability of WDMD in modeling tracked order vibration response of a finite element (FEM) beam excited by an imbalanced rotor. The FE beam under study is an Euler-Bernoulli beam with 30 nodal points, corresponding to 60 degrees of freedom. An imbalanced rotor placed on the 13^{th} nodal point provides the required excitation when operated under the RPM profile shown in Figure 5.3 (b). We chose this profile to closely emulate the RPM profiles of the pass-off tests in Sec-

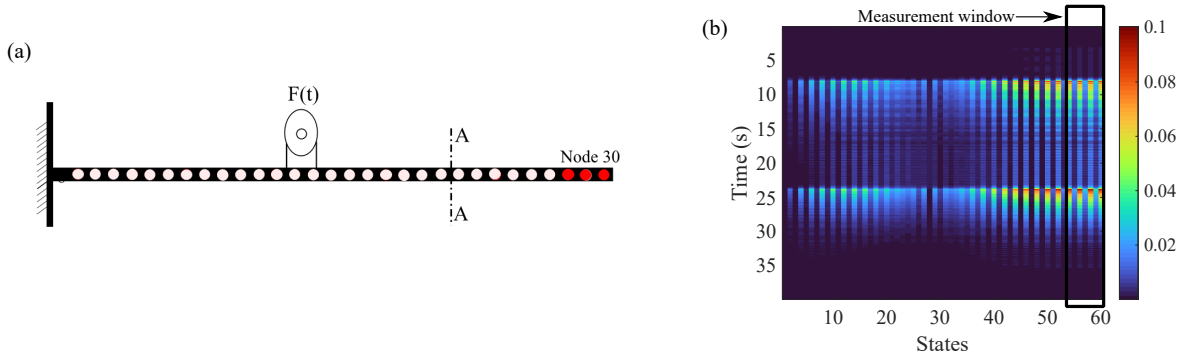


Figure 5.3: (a) Cantilever beam model with the imbalanced rotor positioned above node 13 (b) Evolution of order 1 amplitudes of the nodal points alongside the measurement window.

tion 5.5. The imbalanced rotor provides an excitation force of $F(t) = me\omega^2 sin(\omega t)$, where m and e are the mass and eccentricity of the imbalance, and ω corresponds to the RPM in Hz. The time-domain response of the FEM beam corresponding to the imbalanced rotor is simulated for the duration of the input speed profile. Figure 5.3 (c) shows the evolution of the order 1 amplitude corresponding to the first 60 states. We assume that the transverse responses corresponding to nodes $28, 29$, and 30 are available to the WDMD algorithm to build the data-driven model.

The proposed approach is utilized to generate a SIMO, data-driven, state-space model using the simulated input-output data. This model is then used to simulate the transient dynamic response of the structure to a given excitation, and the quality of the fit is estimated. The following relative error metric is defined to evaluate the quality of the fit [11, 12], namely,

$$\epsilon_{td}^{rel} = \sqrt{\frac{\sum_{i=1}^{K} ||\mathbf{y}(t_i) - \tilde{\mathbf{y}}(t_i)||_2^2}{\sum_{i=1}^{K} ||\mathbf{y}(t_i)||_2^2}}, \tag{5.20}$$

where K denotes the total number of time snapshots. ϵ_{td}^{rel} measures the relative error in time domain between the measured tracked order responses $\mathbf{y}(t)$ and the predicted tracked order responses $\tilde{\mathbf{y}}(t)$. In this section, we also investigate the robustness of the algorithm with respect to measurement noise.

5.4.1 Data-driven modeling using WDMD

We now apply WDMD towards building a SIMO, data-driven, state-space model for the order 1 amplitude of the system. As mentioned before, we carry out a numerical simulation by operating the imbalanced rotor with a pre-defined RPM profile as shown in Figure 5.3 (b). The output responses from the three nodal points and the input profile are assumed to be known and are made available to the WDMD algorithm to build a data-driven model. In the

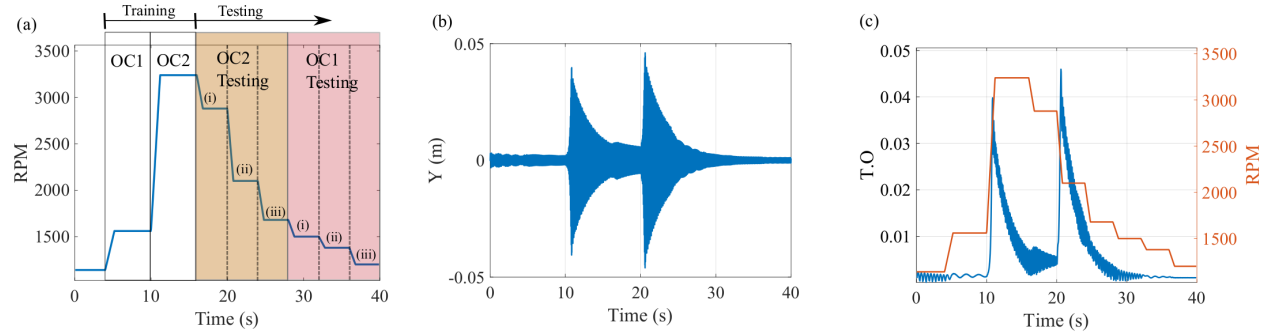


Figure 5.4: (a) Rotor input RPM profile, two separate training phases followed by testing phases, (b) raw time domain response of node 28, and (c) evolution of order 1 amplitude alongside the RPM profile.

present example, for a given rotor RPM profile, the WDMD model is trained to recreate the raw output responses of the measured nodal points and not the tracked order 1 amplitudes. Rather the order 1 amplitude is extracted from the WDMD model-simulated response using (6.23). We take this approach to have a physically consistent data-driven model capable of recreating the accurate dynamics of the system.

We divide the input RPM profile into the training and testing stage. Within the training phase, we further divide the total RPM range into two operating conditions (OC) - $OC1$ and $OC2$, as shown in Figure 5.4 (a). While $OC1$ is valid between RPM ranges $1150-1560$, the $OC2$ is valid between $1561-3200$. We then build separate WDMD reduced order models (ROMs) for each of these operating conditions, using the snapshot matrices of the output responses and the input RPM profile. Figure 5.4 (b) shows the measured transverse response of nodal point 28 during the duration of the simulation. The extracted order 1 amplitude corresponding to Figure 5.4 (b) is shown in Figure 5.4 (c) alongside the input RPM profile. By plotting tracked order amplitudes alongside the RPM profile, one can easily discern the resonance regions of the system as seen from Figure 5.4 (c). From the perspective of a turbofan engine, this greatly simplifies the analysis and hence order tracking methods have been largely adopted in the industry as the method of choice for vibration monitoring.

The two parameters controlling the WDMD algorithm are: (i) the type of wavelet, and (ii) the level of wavelet decomposition. In the current study, the Haar wavelet [143] is selected as the default setting throughout, and the level of decomposition is varied based on case. The default setting for the level of decomposition (J) is set at 4 for the current study. For each of the OC, the WDMD outputs a state-space ROM of the form,

$$\begin{aligned} \boldsymbol{\phi}(t_{k+1}) &= \mathbf{A}_\phi \boldsymbol{\phi}(t_k) + \mathbf{B}_\phi \mathbf{u}(t_k), \\ \mathbf{y}(t_k) &= \mathbf{C}_\phi \boldsymbol{\phi}(t_k) + \mathbf{D}_\phi \mathbf{u}(t_k), \end{aligned} \tag{5.21}$$

where the first equation describes the internal dynamics, illustrating how the pseudo-state vector (MODWT coefficients) $\boldsymbol{\phi}(t) \in \Re^{p\times(J+1)}$ changes in response to the input $\mathbf{u}(t) \in \Re^{m}$ applied to the system. The state vector's dimension is equal to product of the total number of output measurements and the level of decomposition, which in the present case results to 12. Since the output is assumed to be measured at 3 nodal points, the WDMD results in a SIMO state-space model. The regularization parameter, β is set as 10^{-12}. Finally, the dimensions of the state-space matrices are $\mathbf{A}_{\phi} \in \Re^{12\times12}$, $\mathbf{B}_{\phi} \in \Re^{12\times1}$, $\mathbf{C}_{\phi} \in \Re^{3\times12}$, and $\mathbf{D}_{\phi} \in \Re^{3\times1}$. For a first comparison, this WDMD ROM is used to recreate the tracked order response of the system during the training phase. Afterwards, three different testing phases are performed for each of these models and the quality of the fit is assessed using (6.22).

5.4.2 Results of applying the WDMD model

In this section, we present the results of applying WDMD to model the 1^{st} order amplitude, herein defined as $y^{mag}(t)$, for the two OCs, $OC1$ and $OC2$. As mentioned before, this results in two separate SIMO state-space data-driven models that are valid for RPM ranges $1150-1560$ and $1561-3200$, respectively. For a first comparison, we simulate the individual data-driven model with the same training input, and the quality of the fit is evaluated. We then proceed to three levels of the testing phase for both the data-driven models. A low value of ϵ_{td}^{rel} indicates that the errors are low and that WDMD models can accurately predict the dynamics of the system. In the latter part of this section, we also investigate the performance of WDMD for different levels of measurement noise.

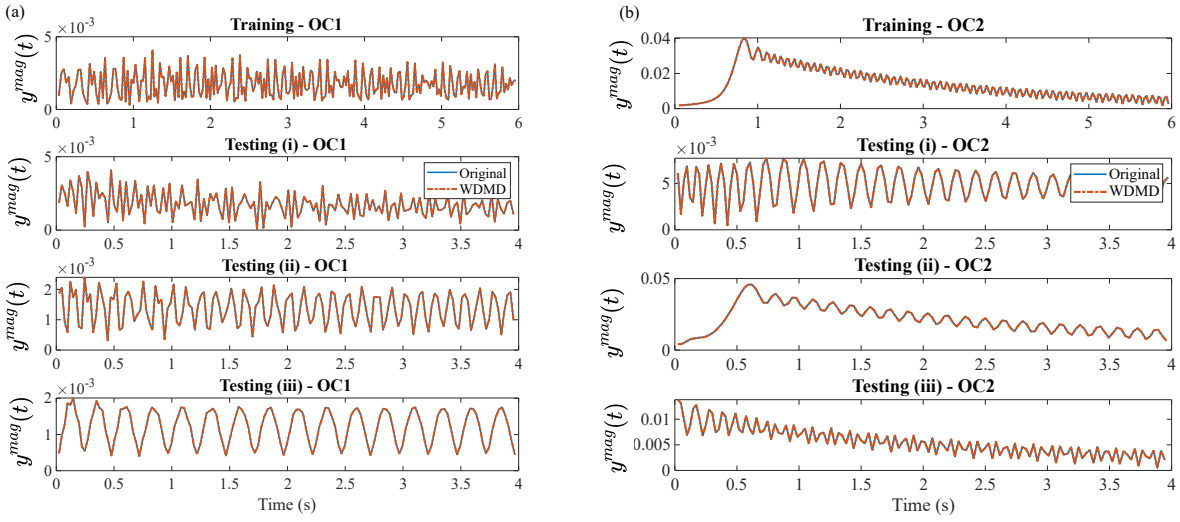


Figure 5.5: Comparing the measured tracked order response alongside the model predicted response for (a) $OC1$, and (b) $OC2$, in training and testing phases.

Figure 5.5 (a) and (b) illustrates the performance of the $OC1$ and $OC2$ WDMD models by comparing the WDMD predicted tracked order responses (y_{pred}^{mag}) with the measured tracked

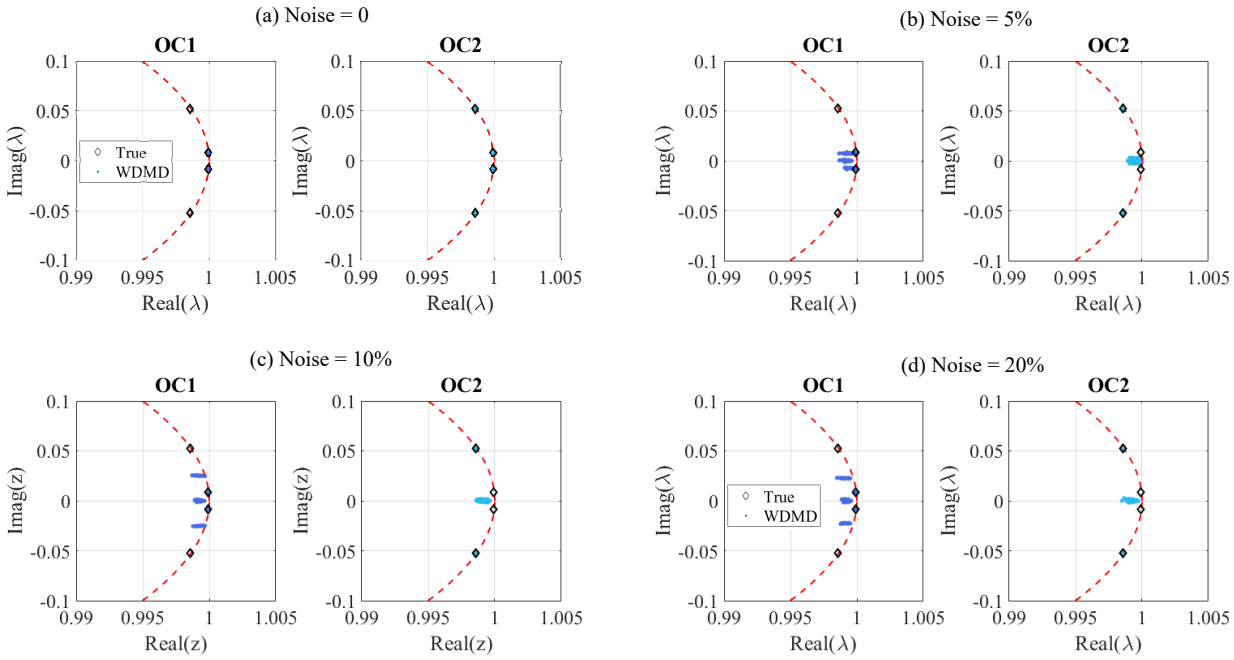


Figure 5.6: Noise dependency of WDMD models based on 50 noise realizations. The figures corresponds to noise percentages of (a) 0%, (b) 5%, (c) 10%, and (d) 20%.

order responses(y_{org}^{mag}). While $OC1$ WDMD results in $\epsilon_{td}^{rel} = 0.0006$ and $\epsilon_{td}^{rel} = 0.0046$ for the training and testing cases, the $OC2$ WDMD model produces an $\epsilon_{td}^{rel} = 0.0406$ and $\epsilon_{td}^{rel} = 0.0283$ for its training and testing cases. The model's efficacy can be discerned from the fact that in the training case for $OC2$, the tracked order vibration levels first increase and then stabilizes to a mean value. However, in the testing (i) for OC2, we see that the values are stable and oscillates about an order amplitude value of 2.52×10^{-3}. Similarly, we observe that the WDMD model accurately captures the non-stationary behavior of the tracked order vibration in testing (iii) The higher value of ϵ_{td}^{rel} corresponding to $OC2$ is warranted, as the $OC2$ passes through system resonance and hence, is a far greater challenge to model than $OC1$.

WDMD can uncover the underlying dynamics and accurately predict the tracked order vibration with few measurements, thereby relaxing the need for high-dimensional measurements. However, in realistic experimental conditions, measurement noise is always present, and it has been shown in several studies that noise adversely affects the DMD algorithm. A prior study by the authors showed that WDMD acts as a smoothening operator, thereby offsetting some level of measurement noise. Hence, in the present study, we study the effect of measurement noise on the performance of the WDMD model by comparing the ability of the algorithm to unearth the true dynamics and evaluate the quality of the fit. For comparison, we use the first 4 eigenvalues of the true model and compare it with the first 4 poles of the WDMD model.

	OC1		OC2	
Noise(%)	Training	Testing	Training	Testing
0	0.0006	0.0046	0.0406	0.0283
5	0.0438	0.2831	0.1887	0.1645
10	0.0783	0.3351	0.1294	0.1023
15	0.0975	0.3104	0.2186	0.1338
20	0.0861	0.2955	0.3076	0.1948
40	0.1703	0.2926	0.5702	0.4203

Table 5.1: ϵ_{td}^{rel} for varying levels of noise corresponding to training and testing cases in $OC1$ and $OC2$.

Figure 5.6 shows the performance of the WDMD for varying levels of measurement noise, averaged over 50 different noise realizations in each case. For the no noise case, while the $OC1$ WDMD model accurately captures the first 2 poles of the system, the $OC2$ WDMD model manages to capture the first 4 poles of the system. For moderate noise levels (5%), we observe that the OC1 WDMD model provides reasonably accurate eigenvalues, at least the first 2. Interestingly for moderate noise levels, $OC2$ WDMD accurately captures the second pair of eigenvalues but not the first pair. For the case of higher noise levels ($10\%-20\%$), while the discrepancy between the true poles and the WDMD poles increases further for $OC1$, the $OC2$ WDMD ROM performance is similar to the moderate level noise scenario. Readers should keep in mind that the WDMD has access to only a few measurement locations. We also investigate the quality of the fit using (6.22) for varying noise levels, which are reported in Table 5.1. The training errors for $OC1$ and $OC2$ increases with increase in noise percentage as reported in Table 5.1. While the lowest value of testing error is reported as $\epsilon_{td}^{rel} = 4.62 \times 10^{-3}$, for $OC1$, for the no noise case, the highest value of testing error is $\epsilon_{td}^{rel} = 5.70 \times 10^{-1}$, for $OC2$, for 40% noise case. From [21], the performance of the model increases with an increase in the number of measurements made. However, such investigation is beyond the purview of the current study.

5.5 Engine model building and results

In this example, we demonstrate the efficacy of the WDMD algorithm in modeling the tracked order vibration time series in a turbofan engine. The data for this study is generated by an industrial turbofan engine commonly used in commercial passenger airplanes. The engine consists of three independent coaxial shafts with a single annular combustor. Since three independent coaxial shafts rotate at different speeds, there are three fundamental first orders of interest. For this study, we will focus our attention on the order 1 amplitudes corresponding to the engine's second stage.

Engine manufacturers require that all manufactured engines undergo a series of pass-off tests before entering into service. During these tests, sensors onboard the engine collect critical information regarding the mechanical and thermal characteristics of the engine. The accelerometer attached to the intercase panel records the vibration responses. Using the measured RPMs of the three fundamental shafts, the vibration monitoring unit on the engine transforms the vibration response into the corresponding tracked order vibrations. Figure 5.7 (a) shows the different operational speeds or OCs in a typical pass-off test. In this case, there are 12 different operational conditions starting from $OC2a$ and going all the way up to $OC13$. Each OC consists of a transitional region and a steady-state region as shown in Figure 5.7 (b). It has been observed that the rate of change of RPM in this transient region affects the vibration in the steady-state region, especially near the resonant regions of the engine. Finally, we plot the tracked order amplitude 1 alongside the pass-off test curve in Figure 5.7 (c). Figure 5.7 (c) demonstrates the non-stationarity nature of the order 1 amplitude and the complexity associated with building a data-driven model. Even at steady RPMs, the influence of the 1^{st} order magnitude does not stay constant. This is believed to be a consequence of the temperature profile of the engine.

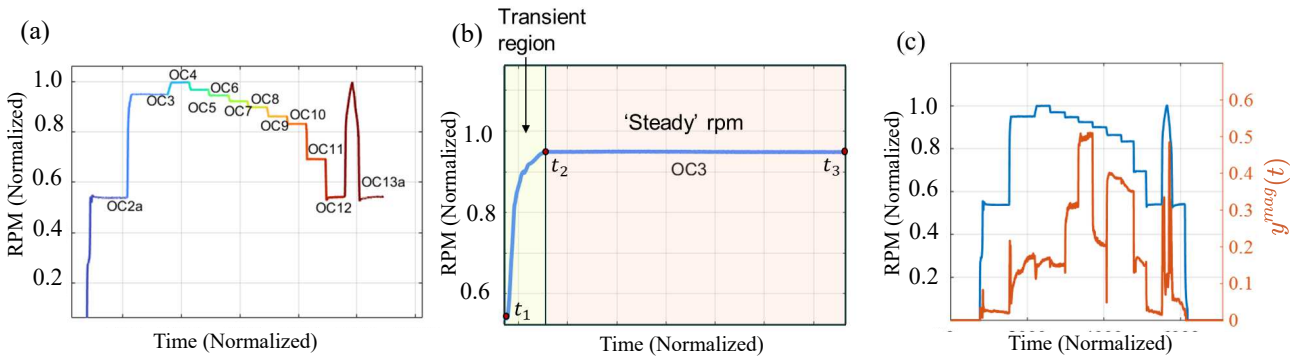


Figure 5.7: (a) Pass-off tests with labelled OC's, (b) zoomed up version of the $OC3$, and (c) order 1 amplitude corresponding to the second stage of the engine, plotted alongside the RPM profile.

Any attempt to build a data-driven model for the order 1 amplitude poses challenges of varying degrees as there are variable dependencies. We refer the readers to [164] for further details regarding the inter-dependency and cross-correlation. For the purpose of the present study, we utilize two inputs, RPM and one of the temperature readings closer to the turbine, to capture the temperature-related changes. Significant changes in operating conditions generally pose a challenge for data-driven models. Indeed, when the operating condition changes enough, the ROM loses its validity, and another training phase must be performed. Hence, in this present example, we built switching dynamical linear system models (SLDS), where the switch variable is the operational condition variable. An SLDS is a natural extension of a linear dynamical system, where we assume the existence of n distinct linear models,

$M = M_l | 1 < l < n$, where each model M_l, is a data-driven WDMD model as described in Section 5.3.1. Hence, to cover the 12 OC's described above, we train 12 different WDMD models valid for each operational condition. This results in a highly flexible SLDS model that seamlessly covers many flight conditions and results in high performance for continuously varying operating conditions.

As mentioned in Section 5.3.1, the quality of the WDMD model depends on the level of decomposition, which is often problem-specific and also depends on the measurement noise. Since the measurement noise is not known in advance, we employ cross-validation to efficiently choose the decomposition level. Cross-validation is a resampling procedure to evaluate data-driven models on a limited data supply. This involves iterating over different models, each with a hyper-parameter value over different training and testing data sets. We evaluate the quality of the fit using (6.22), and the hyper-parameter value that results in lowest testing error across multiple validation tests is selected for the final model. In the present case study, we utilize the data from **41** pass-off tests and employ a 4-fold cross-validation (CV).

5.5.1 Data-driven modeling using WDMD

This section demonstrates the model-building procedure using the WDMD algorithm and compares the results with the state-of-the-art delay-DMD algorithm. As mentioned before, we divide the pass-off data into multiple operating conditions based on the RPM profile. For demonstration purposes, we take the example of $OC9$. The dataset consists of a set of 50 engine pass-off tests. We utilize 40 engine pass-off tests for training the WDMD model and the remaining 10 for testing the model. The data from the 40 pass-off tests can be assumed as 40 different trajectories of the same dynamical system. To handle multiple trajectories, we define,

$$\begin{aligned}\mathbf{Y}_0 &= \left[\mathbf{y}^{(1)}(t_0),\ \mathbf{y}^{(1)}(t_1),\ldots,\ \mathbf{y}^{(1)}(t_{K_1-1}),\mathbf{y}^{(2)}(t_0),\ \mathbf{y}^{(2)}(t_1),\ldots,\ \ \mathbf{y}^{(2)}(t_{K_2-1}),\ldots,\ \mathbf{y}^{(R)}(t_{K_R-1})\right]\\ \mathbf{Y}_1 &= \left[\mathbf{y}^{(1)}(t_1),\ \mathbf{y}^{(1)}(t_2),\ldots,\ \mathbf{y}^{(1)}(t_{K_1}),\mathbf{y}^{(2)}(t_1),\ \mathbf{y}^{(2)}(t_2),\ldots,\ \ \mathbf{y}^{(2)}(t_{K_2}),\ldots,\ \mathbf{y}^{(R)}(t_{K_R})\right],\end{aligned} \tag{5.22}$$

where $\mathbf{Y}_0, \mathbf{Y}_1 \in \Re^{\sum_{i=1}^{R} K_i}$; $\mathbf{y}^{(i)}$ is the measured tracked order vibration corresponding to the i^{th} trajectory, and $i = 1, 2, ..R$; R being the total number of trajectories used for training the algorithm. Further steps follow suit as explained in Section 5.3.1. Thus, the WDMD algorithm results in a MISO, data-driven, state-space model that outputs the 1^{st} order amplitudes, $y^{mag}(t)$, for a given input excitation for each of the OC. Since there are 12 different OC, this procedure has to be repeated for each of the OC, thus resulting in 12 separate linear state-space models pertaining to each of the OC.

The main hyper-parameter controlling the WDMD algorithm is the level of the wavelet decomposition. The higher levels of decomposition (J) effectively smooths out the signal,

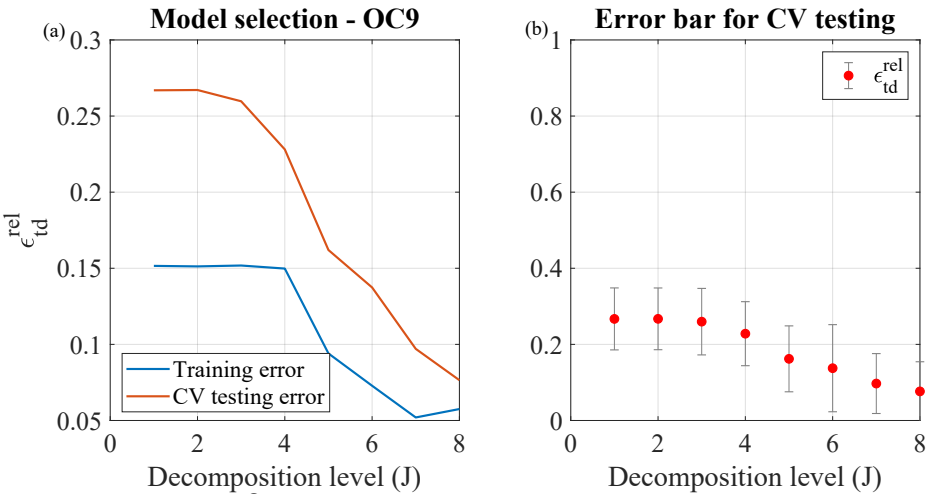


Figure 5.8: Cross-validation results for WDMD : (a) Evolution of the $\epsilon_t^{rel}d$ corresponding to different values of J and (b): Same as (a) but with mean and dispersion of $\epsilon_t^{rel}d$ plotted for demonstrating variability within the cross-validation regimes.

thereby finding application for higher noise cases. The lower levels of decomposition are preferred for cases with rapid changes. Since this is often problem-specific, we employ a 4-fold cross-validation strategy to fix the hyper-parameter of the WDMD model. Specifically, we vary the value of J from $1-8$ and train the WDMD model and perform sequential training and testing across five different datasets. Figure 5.8 shows the results of CV for $OC9$ for demonstrative purposes. Figure 5.8 (a) illustrates the mean value of the training and testing error and the lowest CV testing error is recorded as $\epsilon_{td}^{rel} = 7.54 \times 10^{-2}$ for $J = 8$. This results in a state-space model with the matrices : $\mathbf{A} \in \Re^{9\times9}$, $\mathbf{B} \in \Re^{9\times4}$, $\mathbf{C} \in \Re^{1\times9}$, and $\mathbf{D} \in \Re^{1\times4}$.

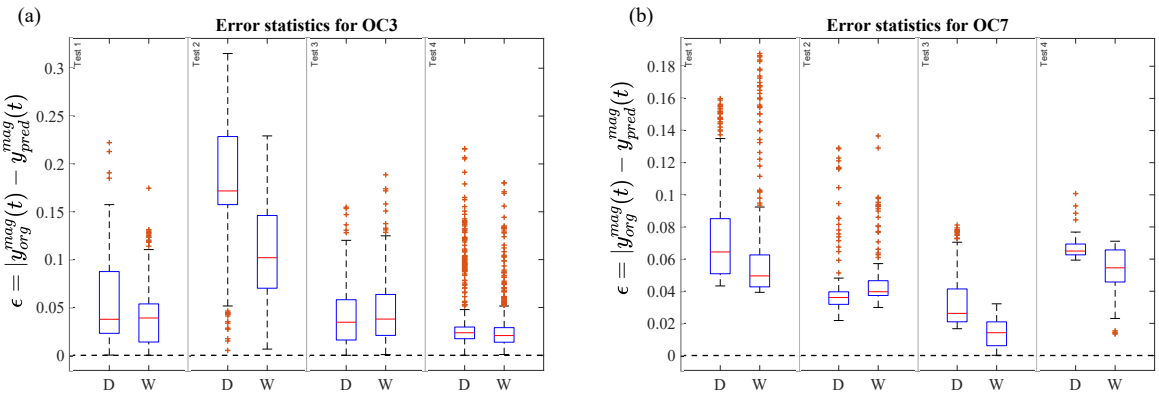


Figure 5.9: Box-whisker plot comparing Delay-DMD error (D) error distribution with WDMD error distribution in 4 cross-validation testing scenarios for $OC3$ and $OC7$.

It is pertinent that any data-driven algorithm must effectively perform satisfactorily across multiple testing regimes by having lower variance ϵ_{td}^{rel}. Figure 5.8 (b) shows the mean value and the standard deviation of the 5 testing errors involved in the 5 fold cross-validation stage. We ascertain from Figure 5.8 (b) that the WDMD algorithm performs satisfactorily during the CV testing stage. We repeat this procedure for each of the WDMD models and select the J value that results in the lowest average cross-validation test errors. The final model specifications for each of the WDMD models are summarized in Table 6.1. For comparison purposes, we compare the results with the time-delay embedded DMD (Delay-DMD) models as reported in [164].

5.5.2 Comparison with Delay-DMD

In this section, we compare our results with the previously developed approach using the time-delay embedded DMD. Interested readers are directed to [164] for a quick summary of the approach. In the present work, we analyze the error time series defined as the absolute of the difference between the original order 1 amplitudes, and the data-driven model predicted tracked order amplitudes (i.e., $\epsilon(t) = |y_{org}^{mag}(t) - y_{pred}^{mag}(t)|$). In Figure 5.9, we present the error time series for the Delay-DMD and WDMD model in the form of box plots. For demonstration purposes, we only illustrate error box plots corresponding to $OC3$ and $OC7$, and we refer the interested readers to the appendix for other OCs. Figure 5.9 consists of 4 box plots representing the 4 cross-validation test cases. In almost all four testing cases, we can see that the box plots corresponding to WDMD (represented as W) has equal or slightly lower mean values compared to the Delay-DMD (represented as D). Figure 5.9 also reveals that the error associated with WDMD follows normality assumption closer than that of Delay-DMD.

Once individual models are built for each operating condition, we calculate the training and testing error associated with each. The training error is computed by taking the average of the training error accrued during the 4-fold cross-validation step and is reported in Table 6.1. In the testing phase, we utilize the 10 pass-off tests data for computing the testing error using the quality of the fit criteria and reports the values in Table 6.1. The WDMD model predicted responses corresponding to all the 12 OC are presented in Figure 5.10 (a) and (b). We utilize one run from the training and testing case for demonstrative purposes. In Figure 5.10, we also plot Delay-DMD predicted responses to have an illustrative comparison. In almost all OC cases, the agreement between the WDMD model-predicted response and the measured response is better than that of the Delay-DMD predicted response, especially for the testing case. Figure 5.10 (c) and (d) shows the training ϵ_{td}^{rel} and testing ϵ_{td}^{rel}, respectively, pertaining to each of the OC. The WDMD model results in an overall lower testing error compared to Delay-DMD in OC's except $OC10$ and $OC12$. We also summarize and present the results in table 6.1. It is interesting to observe that both data-driven models have comparatively higher errors during coast-up. This is attributed to higher variability in the tracked order responses during the initial start of the engine, as can be seen from Figure 5.7 (c) and Figure 5.10.

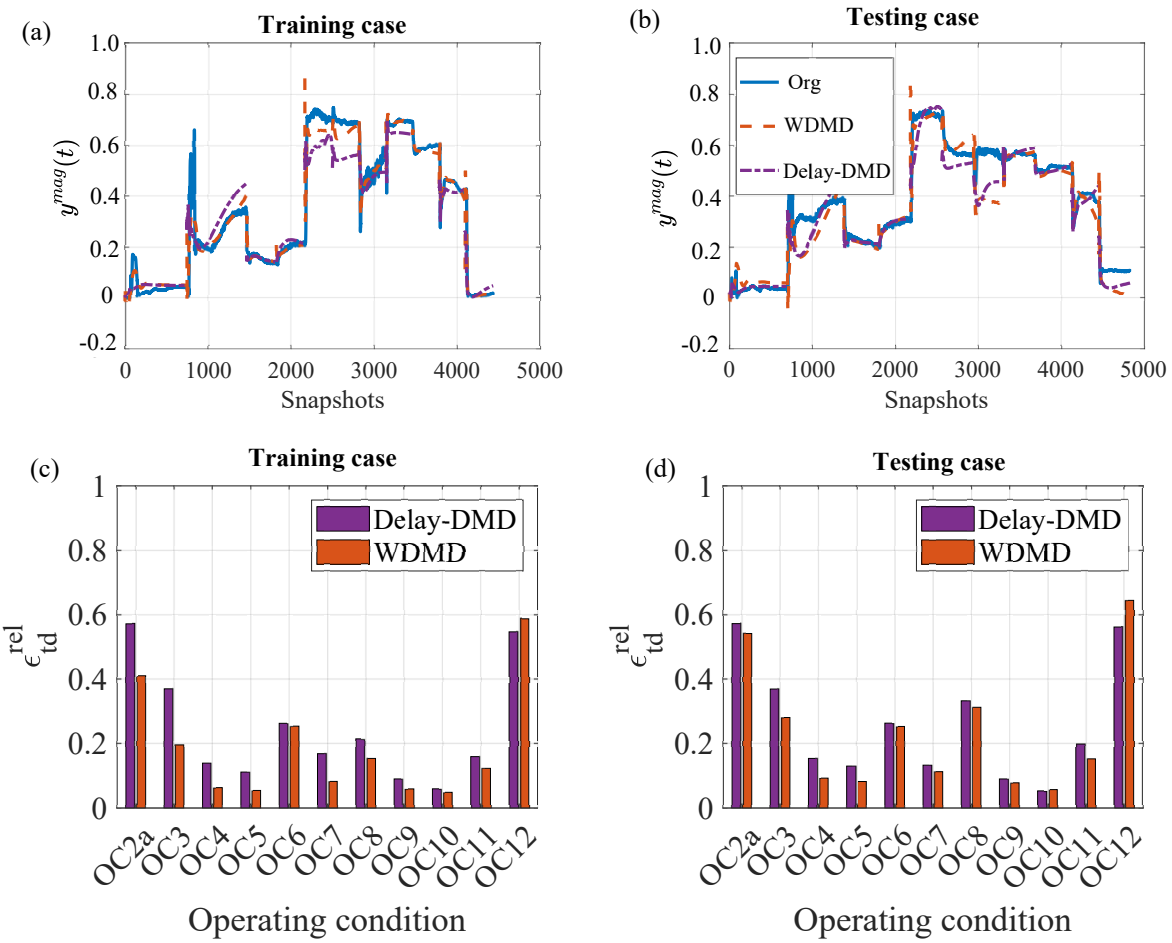


Figure 5.10: Data-driven modeling results for the tracked order vibration for training and testing

5.6 Conclusion

In summary, we presented a novel data-driven methodology, utilizing the recently developed wavelet based dynamic mode decomposition (WDMD) for to model the tracked order vibration in aircraft turbofan engines. First, we provided the basic background into traditional vibration monitoring in aircraft turbofan engine and motivated the need for a data-driven strategy. The paper then described the input-output DMD and the recently developed WDMD in a common mathematical framework. Next, the performance of the WDMD methodology is demonstrated through a numerical simulation involving an imbalanced rotor. The testing and training strategies successfully highlighted the efficacy of the algorithm in capturing the dynamics of the system. The algorithm also performed satisfactorily in the

	WDMD		Delay-DMD	
OC(%)	Model	Testing error	Model	Testing error
OC2a	$\mathbf{A} \in \Re^{4\times4}$, $\mathbf{B} \in \Re^{4\times4}$, $\mathbf{C} \in \Re^{1\times4}$	0.55	$\mathbf{A} \in \Re^{30\times30}$, $\mathbf{B} \in \Re^{30\times4}$, $\mathbf{C} \in \Re^{1\times30}$	0.57
OC3	$\mathbf{A} \in \Re^{5\times5}$, $\mathbf{B} \in \Re^{5\times4}$, $\mathbf{C} \in \Re^{1\times5}$	0.28	$\mathbf{A} \in \Re^{40\times40}$, $\mathbf{B} \in \Re^{40\times4}$, $\mathbf{C} \in \Re^{1\times40}$	0.37
OC4	$\mathbf{A} \in \Re^{4\times4}$, $\mathbf{B} \in \Re^{4\times4}$, $\mathbf{C} \in \Re^{1\times4}$	0.11	$\mathbf{A} \in \Re^{21\times21}$, $\mathbf{B} \in \Re^{21\times4}$, $\mathbf{C} \in \Re^{1\times21}$	0.15
OC5	$\mathbf{A} \in \Re^{6\times6}$, $\mathbf{B} \in \Re^{6\times4}$, $\mathbf{C} \in \Re^{1\times6}$	0.10	$\mathbf{A} \in \Re^{7\times7}$, $\mathbf{B} \in \Re^{7\times4}$, $\mathbf{C} \in \Re^{1\times7}$	0.12
OC6	$\mathbf{A} \in \Re^{6\times6}$, $\mathbf{B} \in \Re^{6\times4}$, $\mathbf{C} \in \Re^{1\times6}$	0.26	$\mathbf{A} \in \Re^{17\times17}$, $\mathbf{B} \in \Re^{17\times2}$, $\mathbf{C} \in \Re^{1\times17}$	0.27
OC7	$\mathbf{A} \in \Re^{5\times5}$, $\mathbf{B} \in \Re^{5\times4}$, $\mathbf{C} \in \Re^{1\times5}$	0.12	$\mathbf{A} \in \Re^{8\times8}$, $\mathbf{B} \in \Re^{8\times4}$, $\mathbf{C} \in \Re^{1\times8}$	0.13
OC8	$\mathbf{A} \in \Re^{8\times8}$, $\mathbf{B} \in \Re^{8\times4}$, $\mathbf{C} \in \Re^{1\times8}$	0.32	$\mathbf{A} \in \Re^{6\times6}$, $\mathbf{B} \in \Re^{6\times4}$, $\mathbf{C} \in \Re^{1\times6}$	0.33
OC9	$\mathbf{A} \in \Re^{8\times8}$, $\mathbf{B} \in \Re^{8\times4}$, $\mathbf{C} \in \Re^{1\times8}$	0.08	$\mathbf{A} \in \Re^{18\times18}$, $\mathbf{B} \in \Re^{18\times4}$, $\mathbf{C} \in \Re^{1\times18}$	0.09
OC10	$\mathbf{A} \in \Re^{7\times7}$, $\mathbf{B} \in \Re^{7\times4}$, $\mathbf{C} \in \Re^{1\times7}$	0.06	$\mathbf{A} \in \Re^{19\times19}$, $\mathbf{B} \in \Re^{19\times4}$, $\mathbf{C} \in \Re^{1\times19}$	0.05
OC11	$\mathbf{A} \in \Re^{8\times8}$, $\mathbf{B} \in \Re^{8\times4}$, $\mathbf{C} \in \Re^{1\times8}$	0.16	$\mathbf{A} \in \Re^{19\times19}$, $\mathbf{B} \in \Re^{19\times4}$, $\mathbf{C} \in \Re^{1\times19}$	0.19
OC12	$\mathbf{A} \in \Re^{4\times4}$, $\mathbf{B} \in \Re^{4\times4}$, $\mathbf{C} \in \Re^{1\times4}$	0.62	$\mathbf{A} \in \Re^{4\times4}$, $\mathbf{B} \in \Re^{4\times4}$, $\mathbf{C} \in \Re^{4\times30}$	0.56

Table 5.2: Model building summary and comparison

presence of added measurement noise. Finally, we presented a data-driven strategy using a switching linear dynamical system built using locally valid reduced order WDMD models to cover the entire range of operational conditions. The ROMs built using WDMD demonstrated higher quality of fits during training and testing phase. There are number of avenues in which this work can be extended. Although WDMD algorithm performed satisfactorily during coast-down stage of operation, further improvements in the modeling during coast-up regions needs to be achieved. Next, future studies will also address non-linearity, damping, and effect of sensor selection on the model.

Acknowledgments

Authors would like to acknowledge the support received through the Rolls-Royce Fellowship and Rolls-Royce Derby, UK for providing the relevant data set for the present work. Authors would also like to thank Mr. Glyn Fox, Rolls-Royce Derby for the helpful suggestions and discussions regarding the engine architecture and data set. Dr. Tarazaga would also like to acknowledge the support received through the John R. Jones III Faculty Fellowship. Mr. Krishnan would also like to acknowledge the support received through the John R. Jones III Graduate Fellowship. The work of Dr. Gugercin was supported in parts by NSF through Grant DMS-1819110 and DMS-1720257.

Chapter 6

Data-driven modeling of tracked order vibrations in turbofan engine using pole-residue interpolation of reduced order surrogates

Vibration monitoring in aircraft engines is traditionally performed using order tracking methodology. Currently, there are no representative efficient physics-based models with adequate level of fidelity for performing vibration predictions in aircraft engines given the existence of various parametric dependencies existing among different attributes such as temperature, pressure, and external conditions. This gap in research is primarily attributed to the limited understanding of the complex nature of multi-disciplinary mutual interactions, variability in operating environment, uncertainties in boundary conditions etc., and the nonlinear nature of engine vibrations. In prior work, we proposed a data-driven framework using wavelet-based dynamic mode decomposition (WDMD) in conjunction with switching linear dynamical system (SLDS) and achieved reasonable accuracy at specific operating regimes. In this work, we leverage the previously developed WDMD and time-delay embedded DMD (Delay-DMD) framework to develop accurate and tractable parametric reduced-order models (p-ROMs) that are valid over specific operating regimes. In particular, we adopt a recently developed pole-residue interpolation scheme to interpolate between several linear data-driven reduced order models (ROMs) build at known parameter samples. We first demonstrate the parametric modeling approach by modeling the transverse vibration of an axially loaded finite element (FE) beam, where the axial loading is the parameter. We then present a parametric modeling strategy for tracked order amplitudes by constructing locally valid Delay-DMD ROMs at different parametric samples corresponding to each pass-off test. We further illustrate the efficacy of the parametric-ROM by comparing it with previously developed techniques such as WDMD.

The contents of this chapter are reproduced from, ***M. Krishnan, S. Gugercin, I. Sever, and P. A. Tarazaga, "Data-driven modeling of tracked order vibrations in turbofan engine using pole-residue interpolation of reduced order surrogates"****, which is currently under internal intellectual property review by Rolls-Royce.*

6.1 Introduction

Vibrations in commercial aircraft engines are of utmost importance as they directly translate to aviation health and safety, and hence are continuously monitored both online and offline for potential abnormalities [2, 8, 9, 10, 97]. As with any rotating machinery, order analysis is the method of choice for monitoring the aircraft engine vibration [2]. Order analysis or order tracking refers to a family of signal processing techniques to extract frequency components related to the rotational speed of the machine (or harmonics) from the measured vibration signal [148, 149, 166]. It is beneficial to track the first order (or harmonics) throughout an engine run. These are directly correlated to the shafts' imbalance and is often referred to as tracked order vibration [7]. The measured tracked order vibration profile of a turbofan engine is affected by variations in flight parameters and conditions [8, 97, 155]. Hence, focusing on 1^{st} order amplitudes in isolation may result in inaccurate judgment when a range of other factors is also known to affect the operation of an aircraft engine. Vibration analysts often face ambiguous situations due to the absence of a representative physics-based model that captures the variations in operating conditions and their effect on vibration. Such ambiguity can lead to a lengthy investigative process and, therefore, an increase in test-bed occupancy cost. In such situations, data-driven modeling has been the method of choice, particularly so if the models from first principles are involved and challenging to build [11, 12, 20, 21, 167, 168]

With the increase in sensing capabilities, data-driven modeling techniques are a compelling family of methods for building models of complex systems [168]. Using measurements made from the system, the data-driven methods extract dynamics without knowing the underlying governing equations. The data-driven modeling techniques applicable to dynamical systems can be broadly divided into two techniques: (i) frequency-domain and (ii) time-domain. The frequency-domain techniques constructs low-dimensional rational functions to fit frequency response measurements: see, e.g., [12, 136, 151, 152, 169, 170, 171, 172, 173, 174, 175, 176]. Likewise, time-domain techniques such as neural-network-based methods [177], operator theoretic framework [20, 36, 37, 42, 178, 179, 180, 181], and time-series methods [182] construct models directly from the measured input-output trajectories. The Dynamical Mode Decomposition (DMD) [20, 21, 22, 42] has become a popular method of choice from the operator theoretic framework for fluid-flow applications, owing to its connection with the Koopman operator [19]. Researchers have proposed various extensions of DMD such as DMD with control (DMDc) [36], input-output DMD (ioDMD) [37], optimized DMD [121], extended DMD (EDMD) [92], time-delay embedded DMD (Delay-DMD) [24, 42, 94] and wavelet-based DMD (WDMD) [21], for various applications. For an exhaustive review of various DMD methods, we refer the readers to [20].

The current study addresses modeling the tracked order vibrations pertaining to a commercial aircraft turbofan engine. In terms of data-driven models applied to this specific problem, Elsaid et al., [155] used a long-short term memory (LSTM) neural network to predict vibrations. However, their approach was limited to predicting vibrations within a

certain time window in the future. Moreover, LSTM lacks physical interpretability and has a known problem of vanishing gradients. In a previous effort, we proposed a modeling strategy using a switching linear dynamical system (SLDS) composed of locally valid data-driven reduced-order models (ROMs) built using WDMD [183] and Delay-DMD [184]. The previous methodology achieved a reasonable accuracy over the whole pass-off tests domain and maintained good physical interpretability. However, handling multiple trajectories was always a challenge, and the previous methodology fell short in capturing various parametric dependencies associated with multiple runs of an engine.

In the present study, we present an efficient parametric data-driven modeling methodology based on the recently developed pole-residue interpolation approach [167]. The parametric data-driven modeling we employ involves interpolating local ROMs at parametric samples, i.e., each of these local ROMs is obtained by applying non-parametric methods at the corresponding parameters. In other words, we build on our previous work [183, 184] and build locally valid ROMs using Delay-DMD [184] and WDMD [183] for every value of the parameter during the training phase. During the testing phase, we interpolate ROMs using the pole-residue interpolation approach for a new parameter value corresponding to the testing phase. Two innovations are proposed in this work. First, we provide a parametric data-driven modeling strategy using pole-residue interpolation with locally valid DMD to handle multiple trajectories. Second, we employ this approach to efficiently model the tracked order vibration in a turbofan engine by accounting for parametric differences.

The paper is structured as follows. First, we present a brief background of the data-driven modeling techniques used in the present work, such as Delay-DMD and WDMD. We then provide an overview of the pole-residue interpolation scheme proposed in [167]. Second, we apply the proposed approach to model a transverse vibration of a numerical FE beam with an axial load, where the axial load is the parameter. The final section applies the proposed methodology to the engine data-set and compares its performance with the previously developed methods.

6.2 Mathematical necessities

In this section, we define the general problem setup using an example of a parametric dynamical system followed by the basic-premise behind the pole-residue interpolation framework. We make use of an axially loaded Euler-Bernoulli finite element beam undergoing transverse vibration to motivate the present example. In the later section, we explain the data-driven surrogates approaches used in the present work.

6.2.1 General problem setup

The frequency domain representation of a dynamical system which is influenced by a parameter $p \in \Re$ can be written as,

$$\begin{aligned}(s\mathbf{I} - \mathbf{A}(p))\mathbf{x}(s,p) &= \mathbf{B}(p)\mathbf{u}(s) \\ \mathbf{y}(s,p) &= \mathbf{C}(p)\mathbf{x}(s,p)\end{aligned} \tag{6.1}$$

where $\mathbf{x}(s,p) \in \Re^N$, $\mathbf{u}(s) \in \Re^m$, $\mathbf{y}(s,p) \in \Re^d$ denotes the state-vector, the measurement vector, and the the input (excitations), respectively. The matrices $\mathbf{A}(p) \in \Re^{N\times N}$, $\mathbf{B}(p) \in \Re^{N\times m}$, and $\mathbf{C}(p) \in \Re^{d\times N}$ denote the state-space matrices. We can denote the input-output mapping using a transfer function as,

$$\mathbf{y}(s,p) = \boldsymbol{\mathcal{H}}(s,p)\mathbf{u}(s); \qquad \boldsymbol{\mathcal{H}}(s,p) = \mathbf{C}(p)(s\mathbf{I} - \mathbf{A}(p))^{-1}\mathbf{B}(p), \tag{6.2}$$

where $\boldsymbol{\mathcal{H}}(s,p) \in \mathbb{C}^{d\times m}$ represent the transfer function of the parametric dynamical system in (6.1). In the present study, we have no information about the internal state-space matrices and we only have access to the inputs and outputs of the dynamical system at a parameter sample. Therefore, we will use our previously developed data-driven modeling approaches such as WDMD [21, 183] and Delay-DMD [184] to build data-driven surrogates at given parameteric samples.

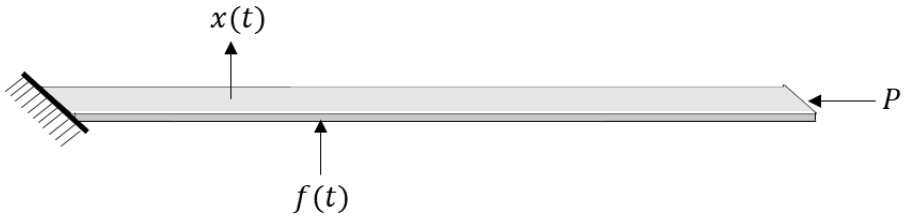


Figure 6.1: Axially loaded finite element beam undergoing transverse vibration

To better help readers understand the parametric dynamical system, we consider a numerical example consisting of an axially loaded finite element beam as shown in Figure A.4. The equation of motion governing the transverse vibration of this axially loaded beam is given as [185],

$$m\frac{\partial^2 x}{\partial t^2} + P\frac{\partial^2 x}{\partial y^2} + EI\frac{\partial^4 x}{\partial t^4} + \mu\frac{\partial x}{\partial t} \;=\; \frac{EA}{2l}\frac{\partial^2 x}{\partial y^2}\int_0^l \left(\frac{\partial x}{\partial y}\right)^2 dx \;+ f(t), \tag{6.3}$$

where m is the mass per unit length (l), $x(y,t)$ is the transverse displacement at position y along the beam at any time t; E is the Youngs modulus; A and I are the area and moment of inertia of the cross section of the beam; $f(t)$ represents the transverse loading on the

beam. The term $\frac{EA}{2l}\frac{\partial^2 x}{\partial y^2}\int_0^l \left(\frac{\partial x}{\partial y}\right)^2 dx$ accounts for the midplane stretching and is assumed to be negligible for small values of axial load P. We further define $p = P/1000$.

We use 30 Euler-Bernoulli beam elements with two degrees of freedom at each nodal point to discritize the beam in Figure A.4. Since, we assumed negligible mid-plane stretching, the effect of axial force reduces to effective reduction in stiffness of the beam. This entails writing (6.3) in terms of its matrix form as,

$$\mathbf{M}\mathbf{x}(t,p) + \mathbf{G}\ \dot{\mathbf{x}}(t,p) + \mathbf{K}\ \mathbf{x}(t,p)\ - \mathbf{N}(p)\ \mathbf{x}(t,p)\ = \mathbf{b}f(t), \tag{6.4}$$

where $\mathbf{M}, \mathbf{K}$, and $\mathbf{G} \in \Re^{60\times 60}$ represents the mass, stiffness and damping matrix, respectively; $\mathbf{b}$ is the loading vector; $\boldsymbol{N} \in \Re^{60\times 60}$ represents the geometric stiffness matrix that includes the effect of the axial force p. We refer interested readers to [185] regarding details about these matrices. In the present numerical simulation, we take axial load as the parameter (p) and obtain the parametric dynamical system from (6.4) as,

$$\mathbf{M}\mathbf{x}(t,p) + \mathbf{G}\ \dot{\mathbf{x}}(t,p) + \hat{\mathbf{K}}(p)\ \mathbf{x}(t,p)\ = \mathbf{b}f(t), \qquad \mathbf{y}(t,p)\ = \mathbf{C}\ \mathbf{x}(t,p) \tag{6.5}$$

where $\widehat{\mathbf{K}}(p) = \mathbf{K} - \mathbf{N}(p)$ is the modified stiffness matrix, accounting for the axial load. The observation matrix $\mathbf{C}$ maps the states into the measured outputs, $\mathbf{y}(t,p)$. The corresponding transfer function of the parametric dynamical system represented by (6.5) is given as,

$$\boldsymbol{\mathcal{H}}(s,p)\ =\ \mathbf{C}\left(s^2\mathbf{M}\ +\ s\mathbf{G}\ +\ \widehat{\mathbf{K}}(p)\right)^{-1}\mathbf{b}. \tag{6.6}$$

where $\boldsymbol{\mathcal{H}}(s,p) \in \mathbb{C}^{d\times m}$ can be expressed in various forms. Note that we use data-driven techniques to learn $\boldsymbol{\mathcal{H}}$ from the input-output trajectory of the system, without having access to any of the system matrices.

There exists various methods in literature to characterize system response for different values of the parameters, interpolate transfer function of parametric dynamical systems. See, eg., [167, 186] and the references therein for further details. Transfer function $\boldsymbol{\mathcal{H}}(s,p)$ can be represented in the pole-residue form as,

$$\boldsymbol{\mathcal{H}}(s,p) = \sum_{j=1}^{r}\left(\frac{\boldsymbol{R}_j(p)}{s-\lambda_j(p)} + \frac{\boldsymbol{R}_j(p)^*}{s-\lambda_j(p)^*}\right), \tag{6.7}$$

where $\lambda_j(p) \in \mathbb{C}$ and $\boldsymbol{R}_j(p) \in \mathbb{C}^{d\times m}$ represents the j^{th} pole and residue of the dynamical system corresponding to parameter sample p, and $.^*$ represents the complex conjugation. The form in (6.7) is a general form of transfer function and as such is challenging to work with. Hence, as a first step, we simplify (6.7) to a single-input-single-output (SISO) system. This entails that residues $\boldsymbol{R}_j$ and the transfer function $\boldsymbol{\mathcal{H}}$ in (6.7) becomes scalar and we represent as R_j and $\mathcal{H}$, respectively.

For a given value of p, local surrogates can be constructed using any suitable data-driven framework. In the present study, we use Delay-DMD and WDMD to construct them. The data-driven techniques used in the present study will be briefly described in Section 6.2.2. Using our recently developed WDMD framework [187], we create data-driven reduced-order models (ROMs) of the dynamic behavior of the beam at the given parametric value. At each parameter sample $p = [p_1, p_2, p_3, \cdots, p_n]$, the learned ROM can be represented as,

$$\left[\tilde{\mathcal{H}}^{(1)}(s),\ \tilde{\mathcal{H}}^{(2)}(s), \tilde{\mathcal{H}}^{(3)}(s), ..., \tilde{\mathcal{H}}^{(n)}(s)\ \right]. \tag{6.8}$$

Further, we convert the learned ROMs into their pole-residue form and build a library as,

$$\boldsymbol{R}_{lib} = \underset{\substack{i=1,2,\cdots n \\ j=1,2,\cdots r}}{\left\{R_j^{(i)}\right\}} \quad \text{and} \quad {}_{lib} = \underset{\substack{i=1,2,\cdots n \\ j=1,2,\cdots r}}{\left\{\lambda_j^{(i)}\right\}} \tag{6.9}$$

Once the poles and residues library is constructed, for any given value of testing parameter p_{test}, we can adopt an interpolation scheme and compute the values of the poles and residues corresponding to the new test parameter. We adopt a weighted interpolation scheme in the present work, thus leading to,

$$R_j^{(test)}(p) = \sum_{i=1}^{n} w_i(p)\ R_j^{(i)} \quad \text{and} \quad \lambda_j^{(test)}(p) = \sum_{i=1}^{n} w_i(p)\ \lambda_j^{(i)} \tag{6.10}$$

where the weight w_i is proportional to the distance of p_{test} from p_i. Using the interpolated residues $R_j^{(test)}$ and poles $\lambda_j^{(test)}$, we obtain the interpolated transfer function as,

$$\tilde{\mathcal{H}}^{(test)}(s,p) = \sum_{j=1}^{r} \left(\frac{R_j^{(test)}(p)}{s - \lambda_j(p)} + \frac{R_j^{(test)}(p)^*}{s - \lambda_j(p)^*} \right). \tag{6.11}$$

6.2.2 Data-driven methods

The data-driven methods and pole-residue interpolation framework will be explained in this section. The next subsection briefly provides an overview of the data-driven modeling techniques such as Delay-DMD and the more recent WDMD. The final subsection briefly summarizes the pole-residue interpolation introduced in [167].

Input-output dynamic mode decomposition

Input-output DMD is an extension of the DMD with control [36] but with specific outputs alongside states and the inputs. As such, ioDMD models a dynamical system of the form,

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{y}(t) &= g(\mathbf{x}(t), \mathbf{u}(t)), \end{aligned} \tag{6.12}$$

where $f : \Re^N \to \Re^N$ and $g : \Re^d \to \Re^d$ represent the evolution operator and the mapping from states to outputs, respectively; $\mathbf{x}(t) \in \Re^N$, $\mathbf{u}(t) \in \Re^m$ and $\mathbf{y}(t) \in \Re^d$ follow definition from (6.1). The ioDMD assumes that the full state vector, the inputs and the outputs are observables for given sampling times, $t_0, t_1, ...t_K$ (equally spaced). As such, this entails writing the following snapshot matrices,

$$\begin{aligned} \mathbf{X}_0 &= [\mathbf{x}(t_0), \mathbf{x}(t_1), \ldots, \mathbf{x}(t_{K-1})] \in \Re^{N \times K} \text{ and } \mathbf{X}_1 = [\mathbf{x}(t_1), \mathbf{x}(t_2), \ldots, \mathbf{x}(t_K)] \in \Re^{N \times K}, \\ \mathbf{U}_0 &= [\mathbf{u}(t_0), \mathbf{u}(t_1), \ldots, \mathbf{u}(t_{K-1})] \in \Re^{m \times K} \text{ and } \mathbf{Y}_0 = [\mathbf{y}(t_0), \mathbf{y}(t_1), \ldots, \mathbf{y}(t_{K-1})] \in \Re^{d \times K}, \end{aligned}$$

where $\mathbf{X}_0$, $\mathbf{Y}_0$, $\mathbf{U}_0$ denote the state snapshot, output snapshot and input snapshot matrices, respectively, and $\mathbf{X}_1$ advances $\mathbf{X}_0$ by a time step. Using the discrete snapshot matrices defined in (6.13), the ioDMD approximates the evolution of (6.12) as a discrete-time linear dynamical system of the form,

$$\begin{aligned} \mathbf{x}(t_{k+1}) &\approx \mathbf{A}\mathbf{x}(t_k) + \mathbf{B}\mathbf{u}(t_k), \\ \mathbf{y}(t_k) &\approx \mathbf{C}\mathbf{x}(t_k) + \mathbf{D}\mathbf{u}(t_k), \end{aligned} \tag{6.13}$$

where $\mathbf{A} \in \Re^{N \times N}, \mathbf{B} \in \Re^{N \times m}, \mathbf{C} \in \Re^{d \times N}, \mathbf{D} \in \Re^{d \times m}$. The original dynamical system in (6.12) can be written in terms of the snapshot matrices as,

$$\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{Y_0} \end{bmatrix} \approx \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X_0} \\ \mathbf{U_0} \end{bmatrix} = \mathbf{\Gamma}\mathfrak{Y} \tag{6.14}$$

where $\mathbf{\Gamma} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \in \Re^{(N+d)\times(N+m)}$, and $\mathfrak{Y} = \begin{bmatrix} \mathbf{X_0} \\ \mathbf{U_0} \end{bmatrix} \in \Re^{K\times(N+m)}$. The ioDMD framework finds the solution using pseudo-inverse of $\mathfrak{Y}$ as,

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{Y_0} \end{bmatrix} \mathfrak{Y}^{\dagger} \tag{6.15}$$

We refer the interested readers to [21, 37] for a detailed discussion.

DMD for system with partial state information

In many real-world dynamical systems, it is not plausible to measure the underlying states of the system, and only system responses to specific inputs are observable. Consider the case of a turbofan engine in operation. The sensors onboard the engine collect data on tracked order vibrations, shaft rotations per minute (RPM), temperatures, etc., inside various stations in the engine, in response to the throttle position. For such systems, direct application of DMD will not be applicable, as one of the major assumptions of DMD is that full-state vectors are observable [20]. For low dimensional systems and systems with limited measurements, researchers have taken inspiration from Taken's embedding theory [96] and proposed applying the DMD procedure on time-shifted coordinates, which came to be known as time-delay embedded DMD [24, 25, 42, 123] In one of our previous efforts, we developed wavelet-based DMD (WDMD) that creates new observables using the maximal overlap discrete wavelet transform (MODWT) coefficients of the outputs and use them as states in the DMD operation [21]. Since the major aim of this study is to present a parametric ROM approach, we use both Delay-DMD and WDMD to construct locally valid ROMs at parametric samples according to the problem-specific.

The principle of Delay-DMD and WDMD is similar as both create new observables from the given measurement data. Let us consider the dynamical system in (6.12), assuming d measurements are made from the dynamical system. The measurement vector $\mathbf{y}(t)$ is written as

$$\mathbf{y}(t_k) = \begin{bmatrix} y_1(t_k) \\ y_2(t_k) \\ \vdots \\ y_d(t_k) \end{bmatrix} \in \Re^d. \tag{6.16}$$

It is possible to write (6.16) using Delay-DMD observables ${}_\tau(y_i(t))$ and MODWT observables ${}_w(y_i(t))$ as,

$$\mathbf{y}(t_k) = \begin{bmatrix} \mathbf{e}_1^T \, {}_\tau(y_1(t_k)) \\ \mathbf{e}_1^T \, {}_\tau(y_2(t_k)) \\ \vdots \\ \mathbf{e}_1^T \, {}_\tau(y_d(t_k)) \end{bmatrix} = \begin{bmatrix} \mathbf{e}^T \, {}_w(y_1(t_k)) \\ \mathbf{e}^T \, {}_w(y_2(t_k)) \\ \vdots \\ \mathbf{e}^T \, {}_w(y_d(t_k)) \end{bmatrix} \tag{6.17}$$

where,

$${}_\tau(y_i(t_k)) = \begin{bmatrix} y_i(t_k) \\ y_i(t_{k-1}) \\ \vdots \\ y_i(t_{k-\tau}) \end{bmatrix} \in \Re^{\tau+1} \text{ and } \quad {}_w(y_i(t_k)) = \begin{bmatrix} [\mathbf{e}_{k+1}^T \tilde{\boldsymbol{D}}_1^i \\ \mathbf{e}_{k+1}^T \tilde{\boldsymbol{D}}_2^i \\ \vdots \\ \mathbf{e}_{k+1}^T \tilde{\boldsymbol{D}}_J^i \\ \mathbf{e}_{k+1}^T \tilde{\boldsymbol{S}}_J^i \end{bmatrix} \in \Re^{J+1}, \tag{6.18}$$

where τ is the number of time-delay; $\tilde{\boldsymbol{D}}_j$ is j^{th} level MODWT detail coefficient, and $\tilde{\boldsymbol{S}}_j$ is

j^{th} level MODWT smooth coefficient. Thus, both these approaches expand the state-sapce dimension by a factor of $(\theta+1)$, where $\theta=\tau$ for Delay-DMD, and $\theta=J$ for WDMD. We can further simplify (6.17) as,

$$\mathbf{y}(t_k)=\begin{cases}\mathbf{C_w}\mathbf{z_w}(t_k); & \text{WDMD}\\ \mathbf{C_d}\mathbf{z_d}(t_k); & \text{Delay-DMD}\end{cases} \tag{6.19}$$

where $\mathbf{z_w}(t_k)=[\ {}_w(y_1(t_k)),\ \ {}_w(y_2(t_k)),\ldots,\ \ {}_w(y_d(t_k))]^T\in\Re^{d\times(J+1)}$ is the wavelet pseudo-state vector, and $\mathbf{z}_\tau(t_k)=[\ {}_\tau(y_1(t_k)),\ \ {}_\tau(y_2(t_k)),\ldots,\ \ {}_d(y_d(t_k))]^T\in\Re^{d\times(\tau+1)}$ is the Delay-DMD pseudo state-vector. The observation matrices corresponding to WDMD and Delay-DMD is formulated as,

$$\mathbf{C_w}=\begin{bmatrix}\mathbf{e}^T & 0 & \cdots & 0 & 0\\ 0 & \mathbf{e}^T & \ddots & 0 & 0\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ 0 & 0 & \ddots & \mathbf{e}^T & 0\\ 0 & 0 & \cdots & 0 & \mathbf{e}^T\end{bmatrix}\in\Re^{d\times d(J+1)},\ \text{and}\ \ \mathbf{C_d}=\begin{bmatrix}\mathbf{e}_1^T & 0 & \cdots & 0 & 0\\ 0 & \mathbf{e}_1^T & \ddots & 0 & 0\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ 0 & 0 & \ddots & \mathbf{e}_1^T & 0\\ 0 & 0 & \cdots & 0 & \mathbf{e}_1^T\end{bmatrix}\in\Re^{d\times d(\tau+1)}. \tag{6.20}$$

We use $\mathbf{z}(t_k)$ to generically represent pseudo-state vector formed using $\mathbf{z_w}(t_k)$ or $\mathbf{z_d}(t_k)$. Given *only* the output measurements $\mathbf{y}(t_k)$ in (6.16) of the underlying dynamical system in (6.12), use (6.18) - (6.20) to form the snapshot matrix of the pseudo state-vector $\mathbf{z}$ as,

$$\mathbf{Z}_0=[\mathbf{z}(t_0)\ \ \mathbf{z}(t_1)\ \ \mathbf{z}(t_2)\ \ \ldots\mathbf{z}(t_{K-1})],\ \text{and}\ \ \mathbf{Z}_1=[\mathbf{z}(t_1)\ \ \mathbf{z}(t_2)\ \ \ldots\mathbf{z}(t_K)]\in\Re^{d(\theta+1)\times(K)}, \tag{6.21}$$

Now, one can replace $\mathbf{X}_0$ and $\mathbf{X}_1$ in (6.13) with $\mathbf{Z}_0$ and $\mathbf{Z}_1$ and follow (6.13) - (6.15) to obtain the system matrices. We refer interested readers to [183, 184] for additional details.

6.3 Numerical simulation using a finite element beam

In the present section, we demonstrate our methodology using numerical simulations carried out on an axially loaded cantilever beam described in Section 6.2.1. This numerical simulation is chosen because it is a simplified problem of the tracked order vibrations in a turbofan engine. We draw parallels between a shaft undergoing transverse vibrations and under axial loading because of the boundary conditions. The temperature inside the turbofan engine increases during operation, thereby causing the shaft to expand. However, the expansion is limited due to the boundary conditions, thereby axially loading the shaft [188].

For a given value of p, we first simulate the response of the beam for an input excitation. Using our recently developed WDMD framework [187], we create data-driven reduced-order

models (ROMs) of the dynamic behavior of the beam at the given parametric value. The quality of the ROM is assessed in frequency and in time domain using the following two error metrics [187],

$$\epsilon_{fd}^{rel} = \sqrt{\frac{\sum_{j=1}^{L_\omega} \|\mathcal{H}(\omega_j) - \tilde{\mathcal{H}}(\omega_j)\|_2^2}{\sum_{j=1}^{L_\omega} \|\mathcal{H}(\omega_j)\|_2^2}} \quad \text{and} \quad \epsilon_{td}^{rel} = \sqrt{\frac{\sum_{i=1}^{K} ||\mathbf{y}(t_i) - \tilde{\mathbf{y}}(t_i)||_2^2}{\sum_{i=1}^{K} ||\mathbf{y}(t_i)||_2^2}}, \tag{6.22}$$

where $\tilde{\mathcal{H}}(\omega)$ and $\tilde{\mathbf{y}}(t)$ denote the FRFand predicted output of the ROM; L_ω and K are the number of frequency samples and time points, respectively. We then repeat this procedure for the other parametric values and store the sets of poles and residues. Given a new axial load, we can obtain the poles and residues corresponding to this loading condition using the interpolation framework explained in Section 6.2.1.

In the present case study, we excite the axially loaded FEM beam using a quadratically varying chirp input over the frequency range $0 - 150$ Hz. We assume that this chirp input and output displacements at 3 nodal locations are measured at a sampling rate of 5000 Hz. In Section 6.3.1 we briefly describe the details of the ROM built using recently developed WDMD methodology and demonstrate the quality of the predicted model using (6.22). Next, we convert the ROMs into their corresponding pole-residue form and use the interpolatory framework to interpolate the poles and residues at new values of axial load. We then compare the interpolated ROMs in the frequency and time domains by exciting the system with known input and observing and comparing the predicted output with the underlying truth.

6.3.1 Data-driven modeling using WDMD

First, we utilize our recently developed WDMD methodology to create high-fidelity ROMs at $p = [0, 0.75, 2, 4]$. This results in 4 different WDMD ROMs with the corresponding sets of poles and residues. For demonstrative purpose, we take the example of $p = 0$ (axial loading is zero). In the present case study, we assume that input-output trajectories of 3 nodal points are measured. Compared to ioDMD, WDMD builds a SISO data-driven model using only the input-output trajectories of the measured nodal points, thereby circumventing the need for all latent states of the system. We also set the following values for the hyper-parameter: $\beta = 10^{-12}, J = 8$. These parameters results in $d \times (J + 1) = 54$ pseudo-states in the WDMD model. Finally, this results in a state-space matrices with the following dimensions $\mathbf{A_w} \in \Re^{54\times54}$, $\mathbf{B_w} \in \Re^{54\times1}$, $\mathbf{C_w} \in \Re^{1\times54}$, and $\mathbf{D_w} \in \Re^{1\times1}$.

Figure 6.2 demonstrates the efficacy of the WDMD ROM in capturing the dynamics of the beam. For a first comparison, we excite the ROM with the same chirp signal used in the training phase. Figure 6.2 (a) illustrates the FEM simulated response of node 13 alongside the ROM predicted response, and they show excellent agreement. The low value of relative error ($\epsilon_{td}^{rel} = 1.49 \times 10^{-2}$) further supports the excellent quality of fit for node 13. Figure 6.2

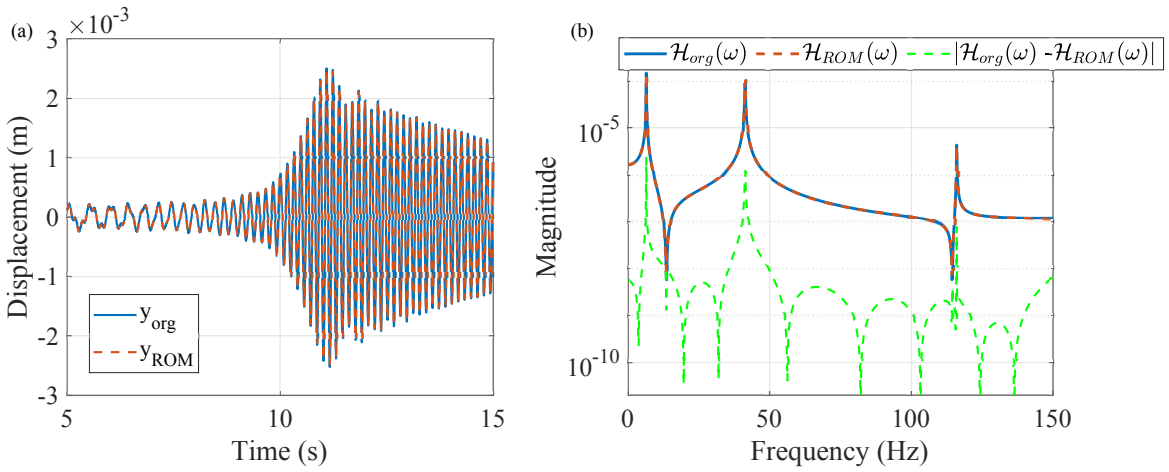


Figure 6.2: (a) Comparison of the ROM predicted response with the original response measured at node 13 when excited with the same training signal (chirp) demonstrating high-fidelity approximation in the training phase. (b) Comparison of the magnitude of the analytical ($\mathcal{H}(\omega)$) with the ROM predicted FRF($\tilde{\mathcal{H}}(\omega)$) at node 13 alongside error magnitudes.

(b) illustrate the performance of the WDMD ROM in capturing the frequency domain characteristics of the underlying system. Having a high fidelity match in the frequency domain is critical as the accuracy of our current methodology relies on the precise capture of the poles and residues of the underlying system. We overlay the FRFof the underlying system and compare it with the ROM FRFin Figure 6.2 (b), which demonstrates a high-fidelity match, capturing the peaks and valleys in original FRFaccurately. The low value of error FRF$\mathcal{H}_{err}(\omega) = |\tilde{\mathcal{H}}(\omega) - \mathcal{H}(\omega)|$, depicted in green in Figure 6.2 (b) further substantiates the good quality of fit. This results in a low $\epsilon_{fd}^{rel} = 1.51 \times 10^{-2}$. The results reported in this case study align well with the finding of [21].

6.3.2 Parameteric ROM

We now proceed to interpolate the poles and residues, for a test-case $\hat{p}$, using the library of poles and residues constructed at $p = [0, 3, 4]$. As per [185], we expect the poles and residues to shift with the increase in axial load, and the motion of the poles follow an elliptical path. Figure 6.3 (a) shows the amplitude frequency responses of the ROMs constructed at $p = [0, 3, 4]$. While for $p = 3$ the WDMD ROM results in $\epsilon_{td}^{rel} = 3.93 \times 10^{-2}$ and $\epsilon_{fd}^{rel} = 2.16 \times 10^{-2}$, for $p = 4$ the WDMD ROM results in $\epsilon_{td}^{rel} = 7.36 \times 10^{-2}$ and $\epsilon_{fd}^{rel} = 3.88 \times 10^{-2}$. Comparing the FRFcorresponding to $p = 0$ with FRFcorresponding to $p = 4$ shows the challenging nature of the problem. With a further increase in value of p, we expect the first pole (frequency) to disappear, however because of the associated instability, we have restricted our study for p values less than buckling load. For any given value of $\hat{p} \in [0, 4]$, we can use an interpolation scheme and obtain the poles and residues of the new system.

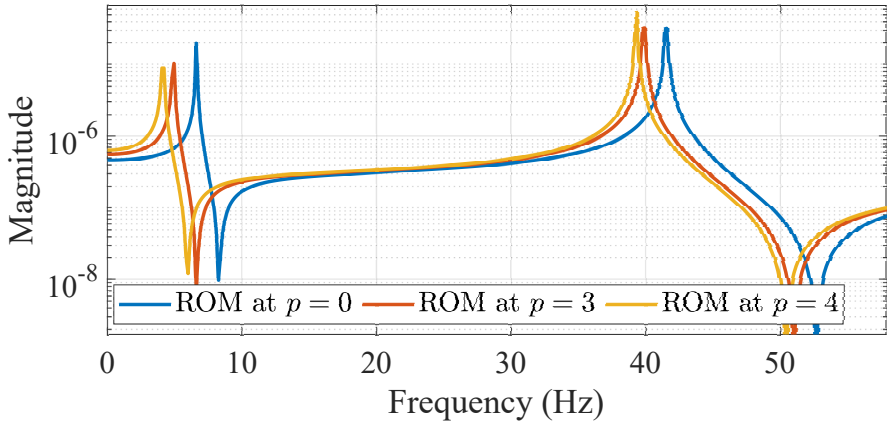


Figure 6.3: WDMD ROM predicted $\mathsf{FRF}(\tilde{\mathcal{H}}(\omega))$ at various $p = 0, p = 3$, and $p = 4$

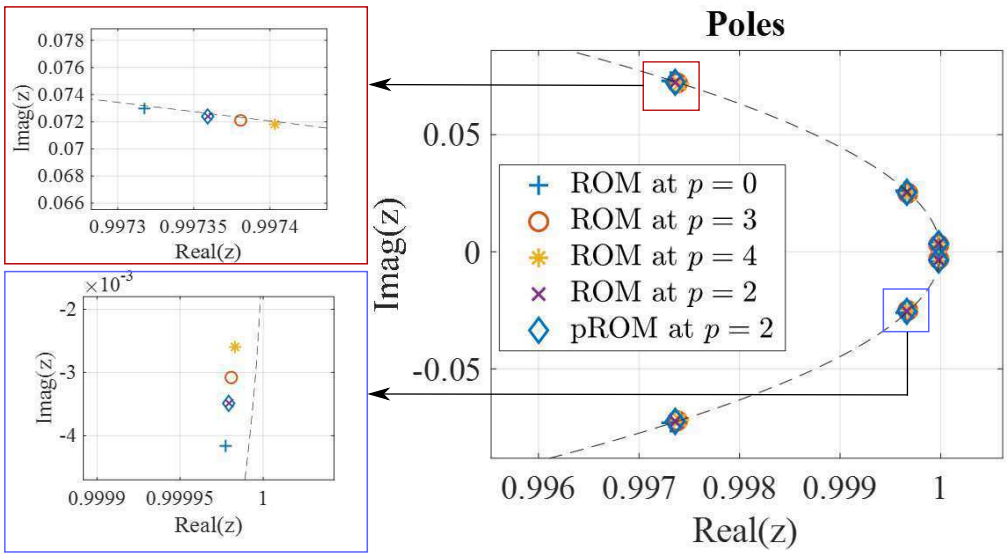


Figure 6.4: Poles of ROM at $p = 0, p = 3$, and $p = 4$ with the interpolated pROM poles at $p = 2$, demonstrating high accuracy

To illustrate the efficacy of our parametric WDMD methodology, we interpolate the ROMs corresponding to values of $\hat{p} = 2$. Figure 6.4 shows the poles of the pROM interpolated at $p = 2$ using the data points available from $p = 0, p = 3$, and $p = 4$. Note that we only show a subset of poles for brevity. The figure demonstrates the accuracy of pROMs compared to the ROM at $p = 2$, thereby illustrating that the interpolatory approach managed to capture the poles of the system accurately. As before, this ROM is evaluated in the time domain and the frequency domain using eq. (6.22), which results in $\epsilon_{td}^{rel} = 3.79 \times 10^{-2}$ and $\epsilon_{fd}^{rel} = 3.26 \times 10^{-2}$.

6.4 Data-driven modeling of turbofan engine vibration using the pole-residue interpolation of reduced order surrogates

This section provides an experimental case study of modeling a turbofan engine tracked order vibration using the current pole-residue interpolation approach. First, we provide an overview about the turbofan engine vibration and order analysis in general. We then proceed to describe a data-driven model building framework to predict the tracked order for a given parameter $\hat{p}$. Finally, we present the results of modeling using the pole-residue interpolation method and compare its efficacy with our previously developed methodology from [183].

6.4.1 Overview of a turbofan engine and order analysis

The data for the present study is generated using a turbofan engine used for powering a typical commercial aircraft. The engine under study is comprised of three independent coaxial shafts with a single annular combustor. The shafts connect the turbines at the back of the engine with the corresponding stage compressors. As with any rotating machinery, the rotation of these shafts creates periodic excitations, which produce vibrations. These vibration signals are composed of frequency components that are integer or non-integer multiples of the rotating frequencies, which are popularly called orders. Mathematically, we can define a p^{th} order as,

$$\psi(t) = Z_p(t)e^{i2\pi\frac{p}{T}t}, \tag{6.23}$$

where $\psi(t)$ is the order component; $Z_p(t)$ is a complex number representing the amplitude and phase of the p^{th} order, and T is the time period of the primary order (in sec). For a turbofan engine, the original measured vibration signal $y(t)$ is comprised of several different orders as,

$$y(t) = \sum_{p=0}^{\infty} Z_p(t)e^{i2\pi\frac{p}{T}t}. \tag{6.24}$$

When viewed alongside the RPM profile, the order amplitude time series reflects important information such as an imbalance in the rotor shafts, bearing failure, misalignment, and resonance amplification. Hence, the order amplitudes are of interest, and there exists a variety of methods in the literature, called order tracking methods, to extract these order amplitudes given the raw vibration signal $y(t)$ and the RPM profile. Some of the most popular order tracking methods are time-domain sampling-based FFT [162], resampling-based order tracking [163], and Vold-Kalman filter-based order tracking [149].

Even though order tracking methods are widely used, it is challenging to interpret the order amplitudes and use them as baselines for flagging potential anomalies. This is because of the dependence of operational conditions in shaping the tracked order response of an engine and

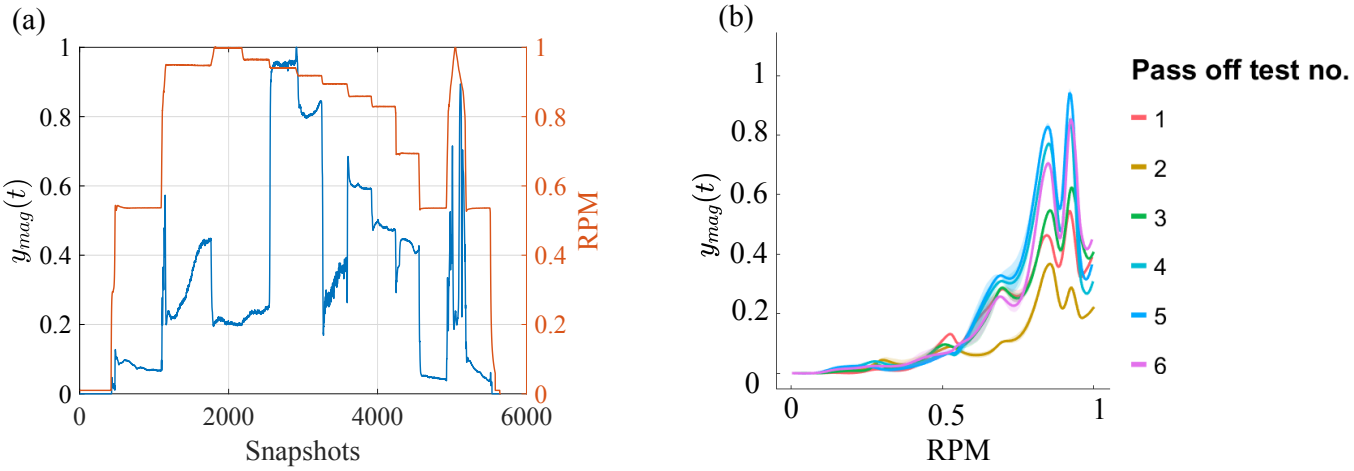


Figure 6.5: (a) 1^{st} order (tracked order) amplitudes alongside the input pass-off test RPM profile, and (b) order spectrum for various pass-off tests demonstrating high inter run variability.

this dependence is unaccounted for when order amplitudes are looked in isolation with the RPM profile. For example, we consider the data from a turbofan engine that has undergone a series of pass-off tests. All engine manufacturers perform a series of pass-off tests on the engines' before entering them into service. In Figure 6.5, we illustrate the data from an engine undergoing consecutive pass-off tests within a span of a few days. Figure 6.5 (a) shows the controlled pass-off test RPM profile alongside the recorded tracked order from one of the stages of the engine. To protect the confidentiality of the data, we normalized all the axes from $0-1$. Figure 6.5 (b) shows the order amplitude as a function of the RPM curve. Traditionally, the order spectrum curve is compared with the expected or baseline curve to flag potential anomalies. However, from Figure 6.5 (b), it can be discerned that even though the input excitation profile (RPM) is the same for all 6 pass-off test curves, the character and amplitude of the order spectrum is vastly different. Figure 6.5 summarizes the challenges in utilizing the order amplitudes for vibration monitoring.

To be effectively used for failure diagnosis calls for an accurate model of the tracked order amplitude with all operational condition dependencies built into the model. However, building an accurate physics-based model of the tracked order amplitudes is challenging because of the sheer number of operating conditions involved. Hence, in our previous efforts, we developed data-driven modeling strategies using a switching linear dynamical system model (SLDS), wherein locally valid WDMD and Delay-DMD ROMs were built at each operational point. An SLDS is a natural extension of a linear dynamical system, where we assume the existence of n distinct linear models, $M = M_l | 1 < l < n$, where each model M_l is a data-driven ROM built using either Delay-DMD or WDMD, as described in Section 6.2.2. In our previous effort, the whole pass-off curve in Figure 6.5 is divided into 12 different OCs,

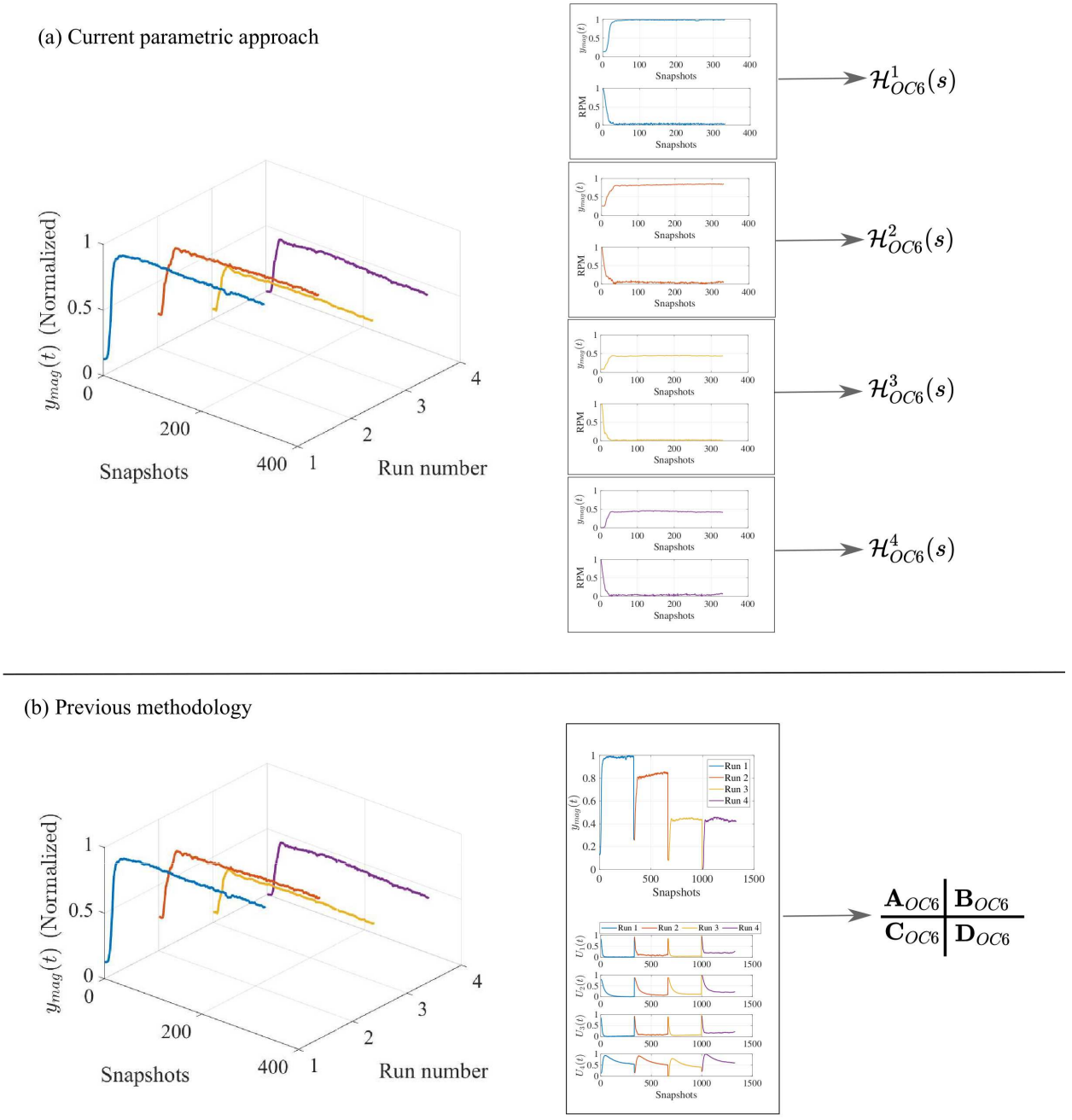


Figure 6.6: Comparison of current parametric training approach with our previous methodology training approach.

proceeding sequentially from $OC2a$ to $OC12$. To handle multiple trajectories (or pass-off tests) associated with training data, we used the following formulation,

$$\mathbf{Y}_0 = \left[\mathbf{y}^{(1)}(t_0),\ ,\ldots,\ \mathbf{y}^{(1)}(t_{K_1-1}), \mathbf{y}^{(2)}(t_0),\ \ldots,\ \ \mathbf{y}^{(2)}(t_{K_2-1}),\ldots,\ \mathbf{y}^{(R)}(t_{K_R-1})\right] \in \Re^{\sum_{i=1}^{R} K_i}$$
$$\mathbf{Y}_1 = \left[\mathbf{y}^{(1)}(t_1),\ \ldots,\ \mathbf{y}^{(1)}(t_{K_1}), \mathbf{y}^{(2)}(t_1),\ \mathbf{y}^{(2)}(t_2),\ldots,\ \ \mathbf{y}^{(2)}(t_{K_2}),\ldots,\ \mathbf{y}^{(R)}(t_{K_R})\right] \in \Re^{\sum_{i=1}^{R} K_i},$$

where $\mathbf{y}^{(i)}(t_k)$ refers to the data corresponding to an OC obtained from i^{th} run of an engine; K_i is the number of data snapshots in the i^{th} run corresponding to the same OC. In our previous approach, we also used 4 inputs in $\mathbf{U}$ matrix. The previously developed methodologies provided reasonable quality of fit around $\epsilon_{td}^{rel} = 0.165$ on average. We direct the interested readers to [183, 184].

The previously developed approaches had a drawback: the data-driven ROMs were generic and fell short of accounting for the parametric differences among pass-off curves. In the case of the turbofan engine, there have been various evidence of temperatures inside the engine and the total time for which engine being cooled down shaping the vibration profile. Hence, in this present study, we apply the pole-residue interpolation approach discussed earlier to develop ROMs at known values of the parameter as in (6.7).

$$\mathcal{H}(s,p) = \sum_{j=1}^{r} \mathbf{c}_j^{II}(p)\ (sI\ -\ \Lambda_j(p))^{-1}\mathbf{b}_j^{II}(p). \tag{6.25}$$

We then proceed to storing all the learned ROMs in the form of a library as below,

$$\left[\tilde{\mathcal{H}}^{(1)}(s),\ \tilde{\mathcal{H}}^{(2)}(s), \tilde{\mathcal{H}}^{(3)}(s), ..., \mathcal{H}^{(n)}(s)\ \right] \tag{6.26}$$

where $\tilde{\mathcal{H}}^{(1)}(s), \tilde{\mathcal{H}}^{(2)}(s), \ldots, \tilde{\mathcal{H}}^{(n)}(s)$ refer to the ROMs build at parameter samples $p_1, p_2, \ldots, p_n$, respectively. As demonstrated in the Section 6.3, we interpolate the learned dynamical systems using the pole-residue interpolation approach and obtain the dynamical system for any value of $\hat{p}$. Once locally valid ROMs corresponding to $\hat{p}$ are interpolated from the library, we stitch together the whole pass-off tests curve using the SLDS approach.

Figure 6.6 provides an overview of the current interpolatory approach and contrasts it with the previously developed data-driven modeling strategy for $OC6$. While the current approach builds locally valid data-driven ROMs corresponding to each pass-off test, the previous modeling philosophy combines all different trajectories into one, as shown in (6.25), and considers them as one trajectory to build a ROM. Compared to the previous methodology, we only consider RPM as the only input for the ROM. Extending it to multiple inputs is a work in progress and will be taken up separately.

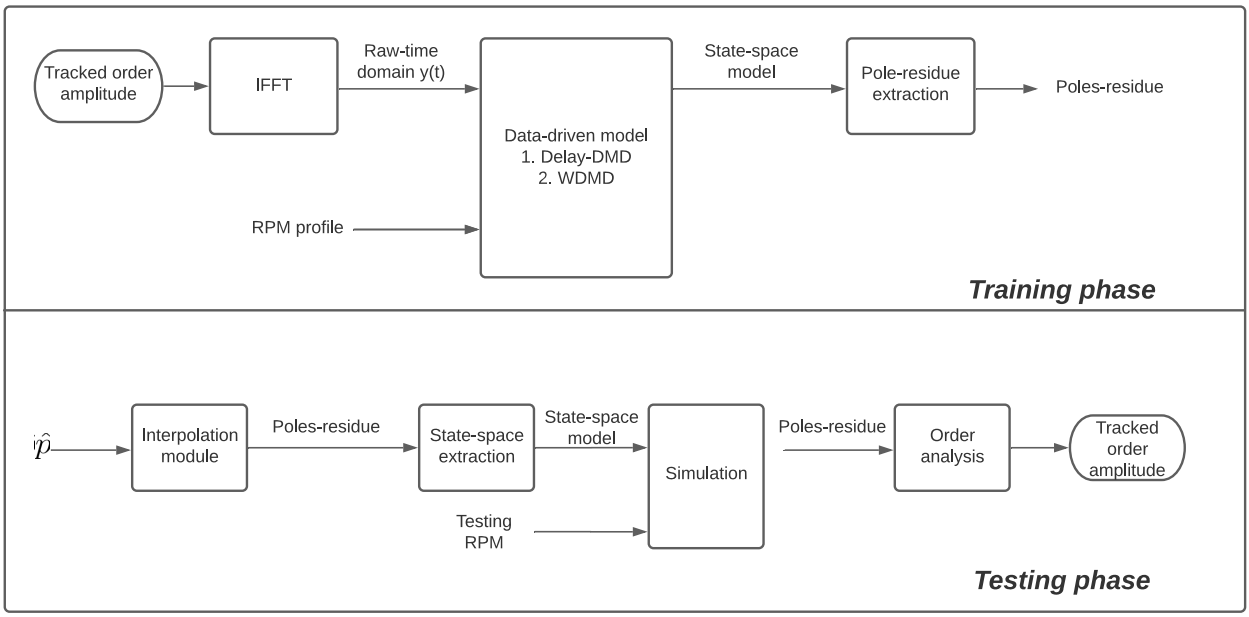


Figure 6.7: Flowchart illustrating various stages of the training and testing.

6.4.2 Data-driven modeling strategy

In this section, we explain our current data-driven modeling strategy for tracked order amplitudes for various operating conditions. Using the same SLDS approach, we build locally valid data-driven ROMs corresponding to every OC in each pass-off test curve. For example, consider the present case of 5 pass-off test profiles. Assuming we utilize 4 pass-off test data for building the models and 1 for testing the model. Note that each pass-off tests consists of same sequence of OCs from $OC2a$ to $OC12$ [183]. As explained before, we build 4 different models (one corresponding to every p) for each of the 12 OCs and store the ROM in pole-residue form and interpolate them for any given value of parameter $\hat{p}$.

The most important requisite for the present approach is that different poles and residues across different ROMs should be trackable, i.e., across different engine runs, the number of real and complex poles should be uniform. However, it is challenging to have the same types of poles across different ROMs, as we are constructing the ROMs using a data-driven approach. The poles are frequently mixed up when the ROM is built using pseudo-states constructed from order amplitudes. To avoid this, we can potentially construct pseudo-states from raw time-domain acceleration signals and build ROMs to predict acceleration signals, given an input RPM profile. Note that this is in sharp contrast to our previous methodology, which constructs WDMD ROMs to predict the tracked order amplitudes directly. However, getting the raw time-domain acceleration signal with the given sensor suite in the commercial engines is challenging. Hence, in the present work, we recreate a pseudo-time-domain signal from the given raw order amplitudes and build ROMs to recreate the time-domain signal as output.

To create the raw time-domain signal from the order amplitudes, we use the order analysis principles. We assume that the time-domain signal is comprised of only the 1^{st} order amplitudes. This is a fair assumption, as we are not interested in recreating the raw acceleration profile of the engine. Figure 6.7 provides an overview of the current data-driven modeling strategy for the training and testing phases. The starting point in the training phase is the 1^{st} order amplitudes, which is the variable to be modeled. Using order analysis and IFFT principles, we recreate the time-domain signal, which has 1^{st} order amplitudes as our starting order amplitudes. We then proceed to model the pseudo-time-domain signal using the data-driven methods explained in Section 6.2.2, with RPM as the input. The data-driven single-input-single-output (SISO) state-space model is converted into its poles and residue form using the approach mentioned in Section 6.2.1. We then store this pole-residue model in the library alongside the p value. For the testing phase, a reverse procedure is adopted. We start with a new parameter $\hat{p}$ and, using an interpolation scheme to obtain the pole and residues corresponding to the $\hat{p}$ in the testing run. We use the state-space extraction module to get back the state-space form, which can be excited using the RPM signal corresponding to this testing phase to obtain the raw pseudo-time-domain signal. Using the principles of order analysis, we obtain back the 1^{st} order amplitudes.

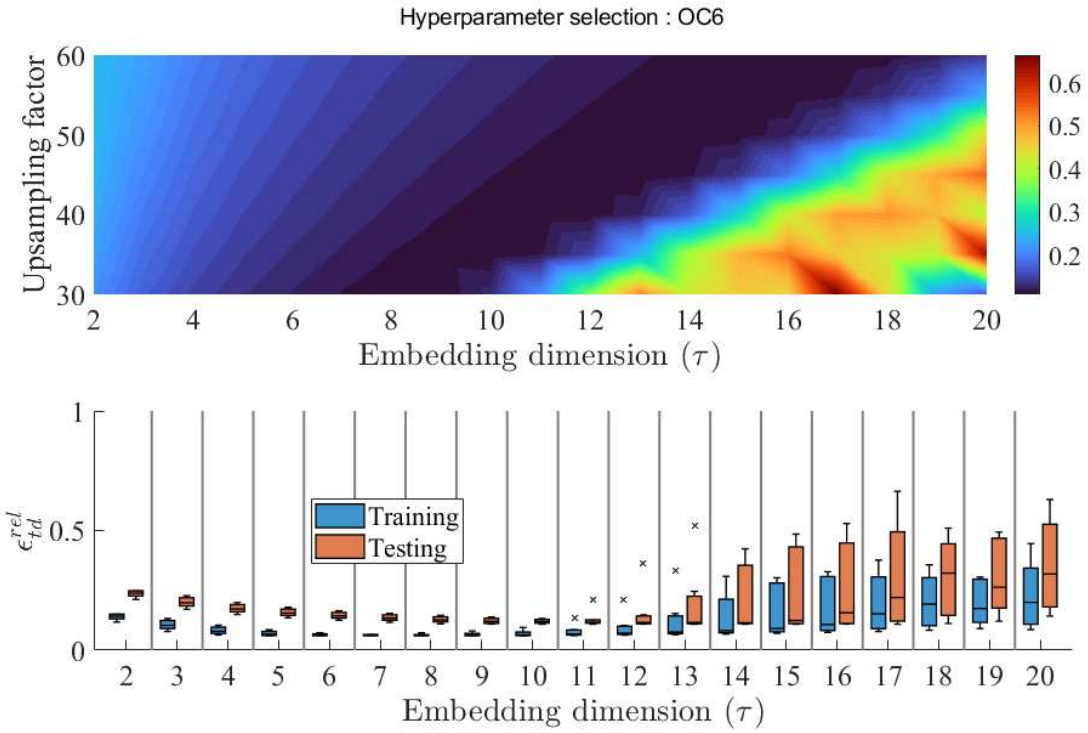


Figure 6.8: ϵ_{td}^{rel} surface resulting from hyper-parameter tuning using 4-fold CV approach

Note that in the present example, we choose Delay-DMD over WDMD to build locally valid ROMs. This is because of higher instances of non-uniformity in the types of poles associated

with WDMD ROMs. This issue can be avoided with a higher number of pass-off tests data and will be taken up in a future work. We reiterate that the major contribution of the present work is the interpolatory approach, and the ROMs can be created using any time-domain data-driven modeling strategy. From Figure 6.7, upsampling window length is one of the major factors affecting the algorithm. From previous literature [24, 42, 94], sampling frequency affects the number of time-delays (τ) in the observable vector, which is also a hyper-parameter. Hence, the upsampling factor and the number of time-delays (τ) are the two hyper-parameters that require tuning in the current methodology.

To avoid confusion, we explicitly refer to the upsampling factor and number of delays (τ) as hyper-parameters and p as the parameter. The two hyper-parameters discussed before pertain only to the data-driven ROMs and are different from the parameter sample (p) at which we build the data-driven ROMs. We use an empirical parameter as p in the current example. This empirical parameter is a function of the temperature and pressure at one of the measuring points in the engine. To avoid potential intellectual property issues, we have masked the formulation of this empirical parameter. In the current study, we restricted our approach to a single parameter. The multi-parametric interpolation is a challenging problem and will be taken up in a future endeavor.

We perform hyper-parameter tuning using a 3-fold cross-validation (CV) study to select the two hyper-parameters. For a given value of τ, we vary the upsampling factor from $30 - 60$ and build the ROM. We then excite this ROM to recreate the training signals and perform a testing phase with a new pass-off test. In both the training and testing phase, we evaluate the quality of the fit using eq. (6.22). We repeat this procedure for every value of τ from 2 to 20. The hyper-parameters which result in the lowest testing error are selected. Figure 6.8 shows the results of hyper-parameter tuning for $OC6$. Figure 6.8 (a) illustrates the ϵ_{td}^{rel} surface for every combination of upsampling factor and embedding dimension, τ. Figure 6.8 (b) displays the distribution of ϵ_{td}^{rel} for every value of τ. The τ value, which has the minimum effect of the upsampling factor, is ideal as it would be robust to variations in the sampling rate. It is also interesting to note that all the minimum value of ϵ_{td}^{rel} lies in an inclined diagonal. This is expected as the ideal number of time delays in the observable vector ϕ increases linearly with the sampling rate. Finally, the hyper-parameter study results in an upsampling value of 35 for a $\tau = 10$.

6.4.3 Results

We utilize the framework in Figure 6.7 to build locally valid parametric ROMs at each of the OCs. We report the results of hyper-parameter selection alongside the 4-fold CV ϵ_{td}^{rel} for all the OCs in Table 6.1. The Table 6.1 also shows the model specifications (J, level of decomposition) for our previous WDMD model and the associated 3-fold CV ϵ_{td}^{rel}. The average CV error as denoted by ϵ_{td}^{rel} in Table 6.1 is lower for our current methodology in OCs with higher expected vibration, such as $OC6 - OC9$. From the point of view of engine

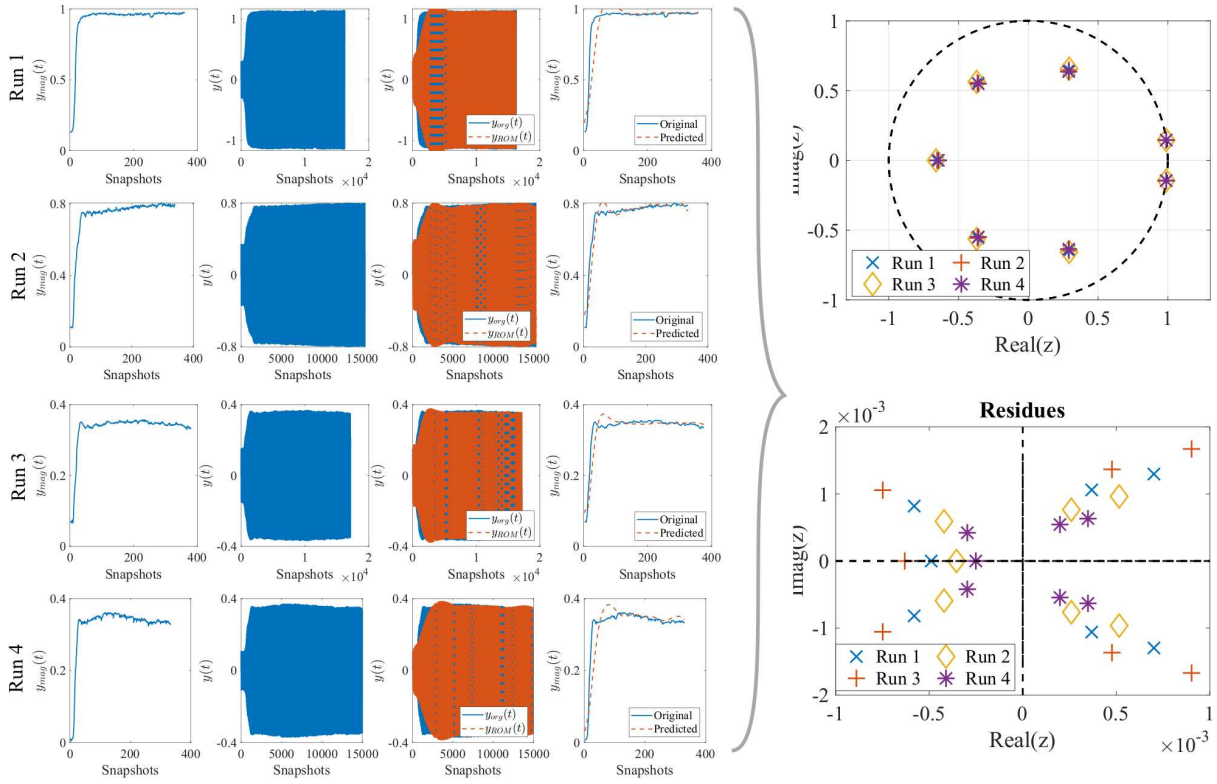


Figure 6.9: Demonstration of the training phase alongside the resulting poles and residues corresponding to each pass-off test.

manufacturers, it is critical to capture the dynamics of the engine accurately at these OCs.

For demonstrative purposes, we take the example of a model building using $OC6$. From the results of 4-fold CV as reported in Table 6.1, we set the value of the hyper-parameter as $\tau = 10$. This results in a state-space model with the matrices : $\mathbf{A} \in \Re^{10\times10}$, $\mathbf{B} \in \Re^{10\times1}$, $\mathbf{C} \in \Re^{1\times10}$, and $\mathbf{D} \in \Re^{1\times1}$. Figure 6.9 illustrates the training steps mentioned in Figure 6.7. Each row corresponds to an engine pass-off test. We start with the tracked order amplitude to be modeled. We then convert these 1^{st} order amplitudes into a pseudo-time-domain signal. Utilizing Delay-DMD, we build the SISO model with RPM as the excitation force. The third column in Figure 6.9 illustrates the predicted pseudo signal and compares it with the original pseudo-time-series, and the results are a close match. In the fourth column, we compare the original 1^{st} order amplitude with the Delay-DMD ROM predicted amplitudes. Further, we convert the state-space model into its pole-residue forms. The poles are composed of 3 ($n_c = 3$) pairs of complex eigenvalues and 1 ($n_r = 1$) real eigenvalue.

One of the major advantages of our present model-building strategy is tracking the movement

	Current approach (Delay-DMD p-ROM)		Previous approach (WDMD ROM)	
OC(%)	Model specification	ϵ_{td}^{rel}	Model specification	ϵ_{td}^{rel}
OC2a	$\tau = 11$; Upsample = 40	0.39	J = 4	0.56
OC3	$\tau = 4$; Upsample = 30	0.31	J = 5	0.27
OC4	$\tau = 10$; Upsample = 60	0.19	J = 4	0.11
OC5	$\tau = 9$; Upsample = 40	0.20	J = 6	0.09
OC6	$\tau = 10$; Upsample = 35	0.09	J = 6	0.21
OC7	$\tau = 14$; Upsample = 55	0.07	J = 5	0.16
OC8	$\tau = 20$;, Upsample = 60	0.15	J = 8	0.31
OC9	$\tau = 7$; Upsample = 40	0.11	J = 8	0.14
OC10	$\tau = 9$; Upsample = 40	0.09	J = 7	0.07
OC11	$\tau = 8$; Upsample = 55	0.17	J = 8	0.16
OC12	$\tau = 2$; Upsample = 60	0.34	J = 4	0.62

Table 6.1: Model building summary and comparison

of the poles. Due to the time-domain construction, the movement of poles between all 4 ROMs can easily be tracked. From Figure 6.9, we note that the motion of the residues is considerably larger than those of poles. However, tracking them does not pose a challenge owing to the nature of construction of the $\mathbf{R}_{lib}$ and λ_{lib}, which ties the residues with the corresponding poles. Hence, as long as the motion of the poles is captured, the residues are bound to be captured accurately. Note that in the present work we used the same weights for the poles and residues and one can resort to a different weighting scheme for poles and residues.

Once individual parametric ROMs are built for each of the OCs, we proceed to compute the quality of fit produced by these data-driven ROMs by exciting the ROMs with training and testing input signals and calculating ϵ_{td}^{rel}. As mentioned before, we use 4 pass-off tests for training and 1 pass-off tests during the testing phase. Using the SLDS approach, we further recreate the whole pass-off tests curve by switching between the various p-ROMs corresponding to each OCs. Figure 6.10 (a) shows our current interpolatory model predicted $y_{mag}(t)$ and compares it with the measured $y_{mag}(t)$ and results are a close match. Figure 6.10 also plots the generic WDMD ROM (previous methodology) predicted response for comparison purposes. From Figure 6.10, we see that the current parametric approach matches the original tracked order responses comparatively better than the previous approach, especially near the end, near resonances, and also towards the end. The error associated with each of the OC is represented in the form of a bar diagram in Figure 6.10. This further demonstrates the superior quality of fit associated with our current approach in higher vibration OCs such as $OC6, OC7$, and $OC8$.

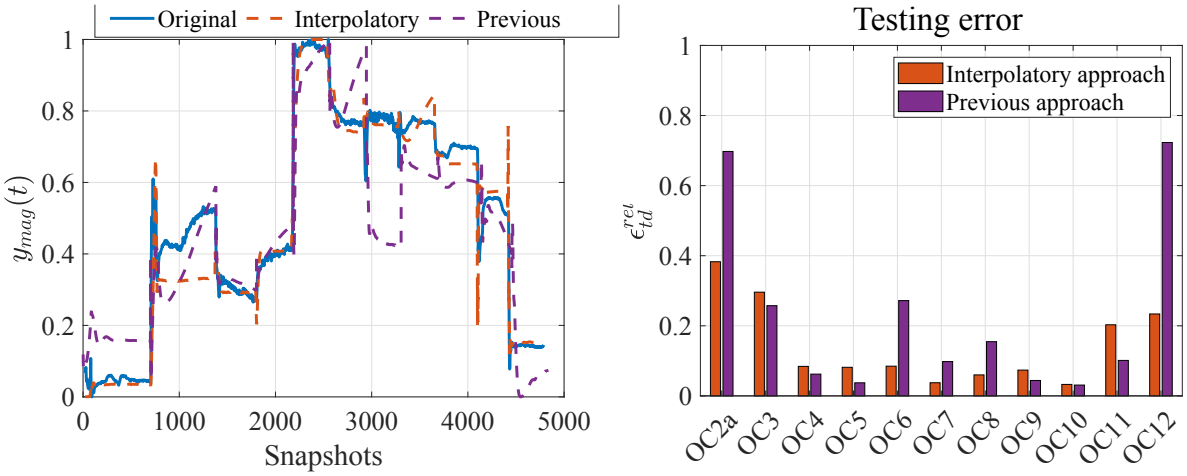


Figure 6.10: (a) Comparison of parametric approach and our previous data-driven modeling techniques' predicted order amplitudes alongside the original tracked order responses, and (b) ϵ_{td}^{rel} corresponding to each OC, demonstrating superior quality of fit using parametric approach compared to the previous approach in higher vibration OCs.

6.5 Conclusions

In summary, the current study presents a parametric approach to model tracked order vibrations in turbofan engine using data-driven reduced order models. In particular, we utilized a recently developed pole-residue interpolation approach to interpolate the constructed ROMs at different parameter samples. First, we provided a brief mathematical background about data-driven modeling techniques, such as Delay-DMD and WDMD and presented an overview of the pole-residue interpolation framework. Next, the methodology was demonstrated using a numerical example of an axially loaded finite element (FE) beam undergoing transverse vibrations. We built WDMD ROMs at different values of the axial loading and using the interpolation scheme, constructed the representative model of the dynamical system at new values of axial loading. Finally, we presented the major contribution of our work by demonstrating the efficacy of our current parametric approach to model tracked order vibrations. The current approach resulted in better capture of the underlying dynamics compared to our previous methodology, especially in operating conditions which historically has a higher vibration associated. Future studies will address multi-parametric interpolation framework and extension to multiple inputs.

Acknowledgments

Authors would like to acknowledge the support received through the Rolls-Royce Fellowship and Rolls-Royce Derby, UK for providing the relevant data set for the present work. Authors

would also like to thank Mr. Glyn Fox, Rolls-Royce Derby for the helpful suggestions and discussions regarding the engine architecture and data set. Dr. Tarazaga would also like to acknowledge the support received through the John R. Jones III Faculty Fellowship. Mr. Krishnan would also like to acknowledge the support received through the John R. Jones III Graduate Fellowship. The work of Dr. Gugercin was supported in parts by NSF through Grant DMS-1819110 and DMS-1720257.

Chapter 7

Conclusions & Future work

This dissertation describes various data-driven strategies to model tracked order vibrations in commercial turbofan engines. Although traditional techniques, such as order tracking methodologies, are widely popular for vibration monitoring, no physics-based models exist for the resulting order amplitudes. This creates ambiguity in interpreting the order amplitudes, leading to higher downtime and loss of revenue for commercial engine manufacturers. Hence, the present dissertation details the efforts undertaken towards developing various data-driven techniques and illustrates the results of applying them to model the tracked order amplitudes. In precise, the order amplitudes corresponding to the second stage of the turbofan engine are considered for analysis. The key conclusions are presented below in addition to suggestions for further improvement towards the development of a broad framework for tracked order vibration modeling in the turbofan engine.

The main contribution of the present dissertation is the development of a novel data-driven methodology to model dynamical systems from its input-output trajectories, without having access to governing physical equations or full internal state dynamics. This was achieved using wavelets in conjunction with the ioDMD approach, leading to the wavelet-based DMD (WDMD). The effectiveness of WDMD algorithm was demonstrated using a numerical case study involving the dynamical response of a finite element cantilever beam. WDMD was utilized to develop a data-driven SIMO state-space dynamical model of the FEM beam based on measured input-output response. The major advantage with the WDMD methodology is that unlike its predecessor ioDMD, WDMD utilizes only a subset of the measurements and approximates the underlying dynamics via a linear model using the maximal overlap discrete wavelet transform (MODWT) coefficients of the measured outputs as the auxiliary state-vector. The error convergence studies illustrated that even with a few measured outputs, WDMD was able to model the underlying dynamical system accurately. The experimental case study on a simple free-free beam demonstrated the efficacy of WDMD methodology as an appropriate candidate for modeling practical dynamical systems despite having no access to internal state measurements.

Once WDMD methodology was developed, further efforts were made to apply this novel technique to model tracked order vibrations. Towards the same, exploratory data analysis and cross-correlation studies were conducted to better inform the models. The variability and dispersion in tracked order vibration within steady-state operational conditions were examined using the dataset measured from a testbed engine undergoing pass-off tests. Gaussian

mixture models (GMMs) contextualized the dispersions associated with the tracked order vibration in operational temperatures. GMM analysis revealed various subpopulations within each OC, thereby partially explaining the dispersion of tracked order vibration. By applying the MS-DPCCA approach, the complex interactions and evolution of partial cross-correlation structure over multiple time scales were analyzed. The ρ_{DPCCA} coefficient values presented in the form of heatmap were subsequently utilized as a threshold for feature selection in building data-driven models.

Since the major aim of the present dissertation is to predict the order amplitude over the whole range of pass-off test, data-driven modeling strategies were directed to cover the entire pass-off test rather than one single operating condition. This entailed developing switching linear dynamical systems (SLDS), wherein data-driven reduced order models (ROMs) were developed using WDMD and then compared with other data-driven models such as Lasso regression and state-of-the-art Delay-DMD. Our current investigation revealed the advantage and efficacy of WDMD to accurately model the tracked order vibration in multiple operating conditions of the engine. The low relative testing error of less than $\epsilon_{td}^{rel} = 0.15$ for WDMD in key operational conditions such as $OC4$, $OC5$, $OC7$, and $OC9$, further demonstrated the efficacy of WDMD. Although WDMD algorithm performed reasonably in most operational conditions, further studies are required to reduce the testing error in OCs such as $OC2a$ and $OC12$.

Finally, the last chapter presented a parametric modeling approach utilizing a recently developed pole-residue interpolation framework. In particular, this framework involved constructing data-driven ROMs at different parameter samples and interpolating the poles and residues of these ROMs for a new parameter sample. The methodology was demonstrated using a numerical example of an axially loaded finite element (FE) beam undergoing transverse vibrations. WDMD ROMs constructed at different values of the axial loading were converted into their corresponding pole-residue form and subsequently interpolated at new values of axial loading to form the representative model of the dynamical system. The parametric approach was further utilized to model and predict the order amplitudes over the whole range of the pass-off test. The parametric approach resulted in better capture of the underlying dynamics compared to our previous methodology, especially in operating conditions that historically have higher vibrations associated. The low relative testing error of less than $\epsilon_{td}^{rel} = 0.10$ for WDMD in key operational conditions such as $OC6$, $OC7$, and $OC9$, further demonstrated the efficacy of parametric approach.

Although the various data-driven modeling strategies resulted in good agreement between the predicted and observed tracked order responses, there are various further avenues for improvement. For starters, further studies are warranted to quantify and improve the performance of the WDMD algorithm in presence of higher levels of noise. Secondly, future research should also address the impact of sensor placement on the WDMD algorithm. Thirdly, the input excitation requirements are also a topic of importance and needs further investigation. Future studies will also address damping and the effect of non-linearity.

The work presented herein demonstrates the feasibility of predicting the tracked order vibration in a commercial turbofan in a test bed environment. In the present work, locally valid linear ROMs were stitched together using a SLDS approach. This approach although valid for pass-off tests, might not perform satisfactorily when scaled to an engine operating on a real aircraft. Extending it to in-flight engine calls for research into non-linear models tying together various operational conditions into one single global framework. This is challenging and hence, the most logical step is extending it to parametric linear models, where the parameter could be operational RPM. In other words, a parametric approach could be developed within each pass-off test itself. This is a natural extension of the last chapter of the present dissertation. Future research will address extension of the pole-residue approach for a multiple-input-multiple-output (MIMO) system. Future studies will also address multi-parametric interpolation framework which calls for a grey-box modeling approach. In particular, a deep neural network that can predict the poles and residues of the underlying dynamical system, given a set of parameters.

Bibliography

[1] S. Gade, H. Herlufsen, H. Konstantin-Hansen, and N. Wismer, "Order tracking analysis," *Technical review*, vol. 2, 1995.

[2] D. Childs, *Turbomachinery rotordynamics: phenomena, modeling, and analysis*. John Wiley & Sons, 1993.

[3] R. Brincker and C. Ventura, *Introduction to operational modal analysis*. John Wiley & Sons, 2015.

[4] P. Avitabile, "Experimental modal analysis," *Sound and vibration*, vol. 35, no. 1, pp. 20–31, 2001.

[5] B. Peeters, P. Gajdatsy, P. Aarnoutse, K. Janssens, and W. Desmet, "Vibro-acoustic operational modal analysis using engine run-up data," in *Proceedings of the 3rd international operational modal analysis conference (IOMAC), Ancona*, 2009.

[6] E. Di Lorenzo, S. Manzato, F. Vanhollebeke, S. Goris, B. Peeters, W. Desmet, and F. Marulo, "Dynamic characterization of wind turbine gearboxes using order-based modal analysis," in *Proceedings of international conference on noise and vibration engineering (isma2014) and international conference on uncertainty in structural dynamics (usd2014)*, pp. 4349–4362, KATHOLIEKE UNIV LEUVEN, DEPT WERKTUIGKUNDE, 2014.

[7] D. Ewins, "Control of vibration and resonance in aero engines and rotating machinery–an overview," *International Journal of Pressure Vessels and Piping*, vol. 87, no. 9, pp. 504–510, 2010.

[8] M. Krishnan, I. A. Sever, and P. A. Tarazaga, "Determining interdependencies and causation of vibration in aero engines using multiscale cross-correlation analysis," in *Model Validation and Uncertainty Quantification, Volume 3*, pp. 265–272, Springer, 2020.

[9] M. Krishnan, S. Gugercin, I. Sever, and P. Tarazaga, "Dynamic data driven modeling of aero engine response," in *Model Validation and Uncertainty Quantification, Volume 3*, pp. 273–278, Springer, 2020.

[10] M. Krishnan, R. Jin, I. A. Sever, and P. A. Tarazaga, "Data based modeling of aero engine vibration responses," in *Sensors and Instrumentation, Aircraft/Aerospace, Energy Harvesting & Dynamic Environments Testing, Volume 7*, pp. 365–368, Springer, 2020.

[11] M. Krishnan, V. V. S. Malladi, and P. A. Tarazaga, "Leveraging a data-driven approach to simulate and experimentally validate a mimo multiphysics vibroacoustic system," *Mechanical Systems and Signal Processing*, vol. 166, p. 108414, 2022.

[12] V. V. S. Malladi, M. I. Albakri, M. Krishnan, S. Gugercin, and P. A. Tarazaga, "Estimating experimental dispersion curves from steady-state frequency response measurements," *Mechanical Systems and Signal Processing*, vol. 164, p. 108218, 2022.

[13] K. Dong, H. Zhang, and Y. Gao, "Dynamical mechanism in aero-engine gas path system using minimum spanning tree and detrended cross-correlation analysis," *Physica A: Statistical Mechanics and its Applications*, vol. 465, pp. 363–369, 2017.

[14] N. Yuan, Z. Fu, H. Zhang, L. Piao, E. Xoplaki, and J. Luterbacher, "Detrended partial-cross-correlation analysis: a new method for analyzing correlations in complex system," *Scientific reports*, vol. 5, p. 8143, 2015.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

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

[17] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[18] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2017.

[19] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 17, no. 5, p. 315, 1931.

[20] J. H. Tu, *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.

[21] M. Krishnan, S. Gugercin, and P. A. Tarazaga, "A wavelet-based dynamic mode decomposition for vibration analysis of underdetermined systems," 2021.

[22] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016.

[23] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[24] J. Tu, C. Rowley, D. Luchtenburg, S. Brunton, and N. J. Kutz, "On dynamic mode decomposition: Theory and applications," *Journal of Computational Dynamics*, vol. 1, no. 2, pp. 391–421, 2014.

[25] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz, "Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition," *Journal of neuroscience methods*, vol. 258, pp. 1–15, 2016.

[26] W. I. T. Uy and B. Peherstorfer, "Operator inference of non-Markovian terms for learning reduced models from partially observed state trajectories," *arXiv preprint arXiv:2103.01362*, 2021.

[27] E. Di Lorenzo, *Operational Modal Analysis for rotating machines*. PhD thesis, University of Naples" Federico II, 2017.

[28] K. Fyfe and E. Munck, "Analysis of computed order tracking," *Mechanical Systems and Signal Processing*, vol. 11, no. 2, pp. 187–205, 1997.

[29] P. G. Hill and C. R. Peterson, "Mechanics and thermodynamics of propulsion," *Reading, MA, Addison-Wesley Publishing Co., 1992, 764 p.*, 1992.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[31] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.

[32] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Multistep neural networks for data-driven discovery of nonlinear dynamical systems," *arXiv preprint arXiv:1801.01236*, 2018.

[33] C. Wehmeyer and F. Noé, "Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics," *The Journal of chemical physics*, vol. 148, no. 24, p. 241703, 2018.

[34] M. Quade, M. Abel, K. Shafi, R. K. Niven, and B. R. Noack, "Prediction of dynamical systems by symbolic regression," *Physical Review E*, vol. 94, no. 1, p. 012214, 2016.

[35] D. L. Ly and H. Lipson, "Learning symbolic representations of hybrid dynamical systems," *Journal of Machine Learning Research*, vol. 13, no. Dec, pp. 3585–3618, 2012.

[36] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic Mode Decomposition with Control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[37] P. Benner, C. Himpe, and T. Mitchell, "On Reduced Input-Output Dynamic Mode Decomposition," *Advances in Computational Mathematics*, vol. 44, pp. 1751–1768, Dec 2018.

[38] J. N. Kutz, X. Fu, and S. L. Brunton, "Multiresolution Dynamic Mode Decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 2, pp. 713–735, 2016.

[39] K. Li, J.-X. Peng, and G. W. Irwin, "A fast nonlinear model identification method," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1211–1216, 2005.

[40] T. Chen, M. S. Andersen, L. Ljung, A. Chiuso, and G. Pillonetto, "System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2933–2945, 2014.

[41] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[42] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit," *Proceedings of the Royal Society A*, vol. 474, no. 2219, p. 20180335, 2018.

[43] K. P. Champion, S. L. Brunton, and J. N. Kutz, "Discovery of nonlinear multiscale systems: Sampling strategies and embeddings," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 1, pp. 312–333, 2019.

[44] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.

[45] L. Ljung, "Approaches to identification of nonlinear systems," in *Proceedings of the 29th Chinese Control Conference*, pp. 1–5, IEEE, 2010.

[46] L. Ljung, "System identification," *Wiley Encyclopedia of Electrical and Electronics Engineering*, 2001.

[47] M. Riedl, A. Suhrbier, H. Malberg, T. Penzel, G. Bretthauer, J. Kurths, and N. Wessel, "Modeling the cardiovascular system using a nonlinear additive autoregressive model with exogenous input," *Physical Review E*, vol. 78, no. 1, p. 011919, 2008.

[48] Q. Chen, K. Worden, P. Peng, and A. Leung, "Genetic algorithm with an improved fitness function for (n) arx modelling," *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 994–1007, 2007.

[49] A. Khan and N. Vyas, "Non-linear parameter estimation using volterra and wiener theories," *Journal of Sound and Vibration*, vol. 221, no. 5, pp. 805–821, 1999.

[50] A. Khan and N. Vyas, "Nonlinear bearing stiffness parameter estimation in flexible rotor–bearing systems using volterra and wiener approach," *Probabilistic Engineering Mechanics*, vol. 16, no. 2, pp. 137–157, 2001.

[51] J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens, and R. Pintelon, "Identification of nonlinear systems using polynomial nonlinear state space models," *Automatica*, vol. 46, no. 4, pp. 647–656, 2010.

[52] J.-P. Noël and G. Kerschen, "Nonlinear system identification in structural dynamics: 10 more years of progress," *Mechanical Systems and Signal Processing*, vol. 83, pp. 2–35, 2017.

[53] R. Haber and H. Unbehauen, "Structure identification of nonlinear dynamic systems—a survey on input/output approaches," *Automatica*, vol. 26, no. 4, pp. 651–677, 1990.

[54] M. B. Rahmoune, A. Hafaifa, A. Kouzou, M. Guemana, and S. Abudura, "Control and diagnostic of vibration in gas turbine system using neural network approach," in *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 573–577, IEEE, 2016.

[55] J. Schoukens, M. Vaes, and R. Pintelon, "Linear system identification in a nonlinear setting: Nonparametric analysis of the nonlinear distortions and their impact on the best linear approximation," *IEEE Control Systems Magazine*, vol. 36, no. 3, pp. 38–69, 2016.

[56] Q. YANG and X. WU, "10 challenging problems in data mining research," *International Journal of Information Technology & Decision Making*, vol. 05, no. 04, pp. 597–604, 2006.

[57] G. C. Reinsel, *Elements of multivariate time series analysis*. Springer Science & Business Media, 2003.

[58] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[59] C. Huang, "Structural identification from ambient vibration measurement using the multivariate ar model," *Journal of Sound and Vibration*, vol. 241, no. 3, pp. 337–359, 2001.

[60] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econometric Reviews*, vol. 29, no. 5-6, pp. 594–621, 2010.

[61] G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, "Machine learning strategies for time series forecasting," in *European business intelligence summer school*, pp. 62–77, Springer, 2012.

[62] V. A. Maksimenko, S. A. Kurkin, E. N. Pitsik, V. Y. Musatov, A. E. Runnova, T. Y. Efremova, A. E. Hramov, and A. N. Pisarchik, "Artificial neural network classification of motor-related eeg: An increase in classification accuracy by reducing signal complexity," *Complexity*, vol. 2018, 2018.

[63] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[64] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[65] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[66] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015.

[67] D. Eck and J. Schmidhuber, "A first look at music composition using lstm recurrent neural networks," *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*, vol. 103, p. 48, 2002.

[68] M. Wöllmer, M. Kaiser, F. Eyben, B. Schuller, and G. Rigoll, "Lstm-modeling of continuous emotions in an audiovisual affect recognition framework," *Image and Vision Computing*, vol. 31, no. 2, pp. 153–163, 2013.

[69] I. Melnyk, A. Banerjee, B. Matthews, and N. Oza, "Semi-markov switching vector autoregressive model-based anomaly detection in aviation systems," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1065–1074, ACM, 2016.

[70] A. Z. Shirazi, M. Hatami, M. Yaghoobi, and S. J. S. M. Chabok, "An intelligent approach to predict vibration rate in a real gas turbine," *Intelligent Industrial Systems*, vol. 2, no. 3, pp. 253–267, 2016.

[71] A. ElSaid, F. El Jamiy, J. Higgins, B. Wild, and T. Desell, "Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration," *Applied Soft Computing*, vol. 73, pp. 969–991, 2018.

[72] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with lstm recurrent neural networks," *arXiv preprint arXiv:1511.03677*, 2015.

[73] H. Goel, I. Melnyk, N. Oza, B. Matthews, and A. Banerjee, "Multivariate aviation time series modeling: Vars vs. lstms," 2016.

[74] A. ElSaid, B. Wild, J. Higgins, and T. Desell, "Using lstm recurrent neural networks to predict excess vibration events in aircraft engines," in *2016 IEEE 12th International Conference on e-Science (e-Science)*, pp. 260–269, IEEE, 2016.

[75] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, p. 452, 2015.

[76] V. Peterka, "Bayesian approach to system identification," in *Trends and Progress in System identification*, pp. 239–304, Elsevier, 1981.

[77] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[78] C.-J. Kim, C. R. Nelson, *et al.*, "State-space models with regime switching: classical and gibbs-sampling approaches with applications," *MIT Press Books*, vol. 1, 1999.

[79] C. K. Carter and R. Kohn, "On gibbs sampling for state space models," *Biometrika*, vol. 81, no. 3, pp. 541–553, 1994.

[80] T. B. Schön, A. Svensson, L. Murray, and F. Lindsten, "Probabilistic learning of nonlinear dynamical systems using sequential monte carlo," *Mechanical Systems and Signal Processing*, vol. 104, pp. 866–883, 2018.

[81] G. Grisetti, C. Stachniss, W. Burgard, *et al.*, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, p. 34, 2007.

[82] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[83] A. Hefny, C. Downey, and G. J. Gordon, "Supervised learning for dynamical system learning," in *Advances in neural information processing systems*, pp. 1963–1971, 2015.

[84] D. Hsu, S. M. Kakade, and T. Zhang, "A spectral algorithm for learning hidden markov models," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1460–1480, 2012.

[85] C. Downey, *Unified Models for Dynamical Systems*. PhD thesis, Carnegie Mellon University Pittsburgh, PA, 2017.

[86] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Nonparametric bayesian learning of switching linear dynamical systems," in *Advances in neural information processing systems*, pp. 457–464, 2009.

[87] B. Podobnik and H. E. Stanley, "Detrended cross-correlation analysis: a new method for analyzing two nonstationary time series," *Physical review letters*, vol. 100, no. 8, p. 084102, 2008.

[88] R. Vassoler and G. Zebende, "Dcca cross-correlation coefficient apply in time series of air temperature and air relative humidity," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 7, pp. 2438–2443, 2012.

[89] G. F. Zebende, "Dcca cross-correlation coefficient: quantifying level of cross-correlation," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 4, pp. 614–618, 2011.

[90] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, p. 5–28, 2010.

[91] A. Alla and J. N. Kutz, "Nonlinear Model Order Reduction via Dynamic Mode Decomposition," *SIAM Journal on Scientific Computing*, vol. 39, no. 5, pp. B778–B796, 2017.

[92] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.

[93] J. Wang and O. R. Barry, "Exploring data-driven modeling and analysis of nonlinear pathological tremors," *Mechanical Systems and Signal Processing*, vol. 156, p. 107659, 2021.

[94] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz, "Time-Delay Observables for Koopman: Theory and Applications," *SIAM Journal on Applied Dynamical Systems*, vol. 19, pp. 886–917, Jan. 2020. Publisher: Society for Industrial and Applied Mathematics.

[95] B. Bhowmik, M. Krishnan, B. Hazra, and V. Pakrashi, "Real-time unified single-and multi-channel structural damage detection using recursive singular spectrum analysis," *Structural Health Monitoring*, vol. 18, no. 2, pp. 563–589, 2019.

[96] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical systems and turbulence, Warwick 1980*, pp. 366–381, Springer, 1981.

[97] I. A. Sever, "Nonlinear vibration phenomena in aero-engine measurements," in *Dynamics of Coupled Structures, Volume 4*, pp. 241–252, Springer, 2016.

[98] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[99] S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian approaches to gaussian mixture modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1133–1142, 1998.

[100] M. Chen, "Dirichlet process gaussian mixture model." www.mathworks.com/matlabcentral/fileexchange/55865-dirichlet-process-gaussian-mixture-model, Mar. 2016.

[101] A. E. Gelfand, S. E. Hills, A. Racine-Poon, and A. F. Smith, "Illustration of bayesian inference in normal data models using gibbs sampling," *Journal of the American Statistical Association*, vol. 85, no. 412, pp. 972–985, 1990.

[102] A. E. Gelfand, "Gibbs sampling," *Journal of the American statistical Association*, vol. 95, no. 452, pp. 1300–1304, 2000.

[103] N. Yuan, E. Xoplaki, C. Zhu, and J. Luterbacher, "A novel way to detect correlations on multi-time scales, with temporal evolution and for multi-variables," *Scientific reports*, vol. 6, no. 1, pp. 1–11, 2016.

[104] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "Nonparametric bayesian learning of switching linear dynamical systems," *Advances in Neural Information Processing Systems*, vol. 21, pp. 457–464, 2008.

[105] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, "Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 406, p. 132401, 2020.

[106] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[107] O. Ghattas and K. Willcox, "Learning physics-based models from data: perspectives from inverse problems and model reduction," *Acta Numerica*, vol. 30, pp. 445–554, 2021.

[108] A. C. Antoulas, S. Lefteriu, and A. C. Ionita, *Chapter 8: A Tutorial Introduction to the Loewner Framework for Model Reduction*, pp. 335–376.

[109] A. C. Antoulas, C. A. Beattie, and S. Güğercin, *Interpolatory Methods for Model Reduction*. Philadelphia: SIAM, 2020.

[110] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. , Rozza, W. Schilders, and L. M. Silveira, eds., *Model Order Reduction: Volume 1: System- and Data-Driven Methods and Algorithms*. De Gruyter, 2021.

[111] M. R. Souza, D. Beli, N. S. Ferguson, J. R. d. F. Arruda, and A. T. Fabro, "A Bayesian approach for wavenumber identification of metamaterial beams possessing variability," *Mechanical Systems and Signal Processing*, vol. 135, p. 106437, 2020.

[112] M. Bertha and J.-C. Golinval, "Multivariate ARMA Based Modal Identification of a Time-varying Beam," in *Topics in Modal Analysis & Testing, Volume 10*, pp. 273–280, Springer, 2016.

[113] M. El-Kafafy, B. Peeters, T. Geluk, and P. Guillaume, "The MLMM modal parameter estimation method: A new feature to maximize modal model robustness," *Mechanical Systems and Signal Processing*, vol. 120, pp. 465–485, 2019.

[114] D. Formenti and M. Richardson, "Parameter estimation from frequency response measurements using rational fraction polynomials (twenty years of progress)," in *Proceedings of International Modal Analysis Conference XX*, Citeseer, 2002.

[115] P. Guillaume, P. Verboven, S. Vanlanduit, H. Van Der Auweraer, and B. Peeters, "A poly-reference implementation of the least-squares complex frequency-domain estimator," in *Proceedings of IMAC*, vol. 21, pp. 183–192, A Conference & Exposition on Structural Dynamics, Society for Experimental …, 2003.

[116] T. Kim, O.-S. Kwon, and J. Song, "Response prediction of nonlinear hysteretic systems by deep neural networks," *Neural Networks*, vol. 111, pp. 1–10, 2019.

[117] M. I. Albakri, V. V. S. Malladi, S. Gugercin, and P. A. Tarazaga, "Estimating dispersion curves from frequency response functions via vector-fitting," *Mechanical Systems and Signal Processing*, vol. 140, p. 106597, 2020.

[118] V. V. S. Malladi, M. I. Albakri, M. Krishnan, S. Gugercin, and P. A. Tarazaga, "Estimating experimental dispersion curves from steady-state frequency response measurements," *Mechanical Systems and Signal Processing*, vol. 164, p. 108218, 2022.

[119] I. Mezić and A. Banaszuk, "Comparison of systems with complex behavior," *Physica D: Nonlinear Phenomena*, vol. 197, no. 1-2, pp. 101–133, 2004.

[120] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of fluid mechanics*, vol. 641, pp. 115–127, 2009.

[121] K. K. Chen, J. H. Tu, and C. W. Rowley, "Variants of dynamic mode decomposition: boundary condition, Koopman, and Fourier analyses," *Journal of nonlinear science*, vol. 22, no. 6, pp. 887–915, 2012.

[122] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, "Sparsity-promoting dynamic mode decomposition," *Physics of Fluids*, vol. 26, no. 2, p. 024103, 2014.

[123] S. Le Clainche and J. M. Vega, "Higher Order Dynamic Mode Decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 2, pp. 882–925, 2017.

[124] I. Kevrekidis, C. Rowley, and M. Williams, "A Kernel-Based Method for Data-Driven Koopman Spectral Analysis," *Journal of Computational Dynamics*, vol. 2, no. 2, pp. 247–265, 2015.

[125] Y. Yuan, K. Zhou, W. Zhou, X. Wen, and Y. Liu, "Flow prediction using dynamic mode decomposition with time-delay embedding based on local measurement," *Physics of Fluids*, vol. 33, no. 9, p. 095109, 2021.

[126] J. L. Proctor and P. A. Eckhoff, "Discovering dynamic patterns from infectious disease data using dynamic mode decomposition," *International Health*, vol. 7, pp. 139–145, 02 2015.

[127] J. Grosek and J. N. Kutz, "Dynamic mode decomposition for real-time background/-foreground separation in video," *arXiv preprint arXiv:1404.7592*, 2014.

[128] A.-M. Yan, G. Kerschen, P. De Boe, and J.-C. Golinval, "Structural damage diagnosis under varying environmental conditions—part ii: local PCA for non-linear cases," *Mechanical Systems and Signal Processing*, vol. 19, no. 4, pp. 865–880, 2005.

[129] M. Krishnan, B. Bhowmik, B. Hazra, and V. Pakrashi, "Real time damage detection using recursive principal components and time varying auto-regressive modeling," *Mechanical Systems and Signal Processing*, vol. 101, pp. 549–574, 2018.

[130] B. Kramer, P. Grover, P. Boufounos, S. Nabi, and M. Benosman, "Sparse Ssensing and DMD-Based Identification of Flow Regimes and Bifurcations in Complex Flows," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 2, pp. 1164–1196, 2017.

[131] N. B. Erichson, L. Mathelin, J. N. Kutz, and S. L. Brunton, "Randomized Dynamic Mmode Ddecomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 4, pp. 1867–1891, 2019.

[132] W. I. T. Uy, Y. Wang, Y. Wen, and B. Peherstorfer, "Active operator inference for learning low-dimensional dynamical-system models from noisy data," *arXiv preprint arXiv:2107.09256*, 2021.

[133] P. Goyal and P. Benner, "Learning Dynamics from Noisy Measurements using Deep learning with a Runge-Kutta Constraint," *arXiv preprint arXiv:2109.11446*, 2021.

[134] A. Mayo and A. Antoulas, "A framework for the solution of the generalized realization problem," *Linear algebra and its applications*, vol. 425, no. 2-3, pp. 634–662, 2007.

[135] Z. Drmač, S. Gugercin, and C. Beattie, "Vector Fitting for Matrix-Valued Rational Approximation," *SIAM Journal on Scientific Computing*, vol. 37, no. 5, pp. A2346–A2379, 2015.

[136] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Transactions on power delivery*, vol. 14, no. 3, pp. 1052–1061, 1999.

[137] A. Mauroy, I. Mezić, and Y. Susuki, *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*, vol. 484. Springer Nature, 2020.

[138] Z. Drmac, I. Mezić, and R. Mohr, "Data Driven Koopman Spectral Analysis in Vandermonde–Cauchy Form via the DFT: Numerical Method and Theoretical Insights," *SIAM Journal on Scientific Computing*, vol. 41, no. 5, pp. A3118–A3151, 2019.

[139] Z. Drmač, "Dynamic Mode Decomposition—A Numerical Linear Algebra Perspective," in *The Koopman Operator in Systems and Control*, pp. 161–194, Springer, 2020.

[140] J. Kou and W. Zhang, "Dynamic mode decomposition with exogenous input for data-driven modeling of unsteady flows," *Physics of Fluids*, vol. 31, no. 5, p. 057106, 2019.

[141] J. Annoni, P. Gebraad, and P. Seiler, "Wind farm flow modeling using input-output dynamic mode decomposition," in *American Control Conference (ACC)*, pp. 506–512, 2016.

[142] I. Mezić, "Spectral Properties of Dynamical Systems, Model Reduction and Decompositions," *Nonlinear Dynamics*, vol. 41, no. 1, pp. 309–325, 2005.

[143] S. Mallat, *A Wavelet Tour of Signal Processing*. Elsevier, 1999.

[144] I. Daubechies, *Ten lectures on wavelets*. SIAM, 1992.

[145] D. B. Percival and A. T. Walden, *Wavelet Methods for Time Series Analysis*, vol. 4. Cambridge university press, 2000.

[146] R. J. Allemang, "The Modal Assurance Criterion–Twenty Years of Use and Abuse," *Sound and vibration*, vol. 37, no. 8, pp. 14–23, 2003.

[147] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz, "Time-Delay Observables for Koopman: Theory and Applications," *SIAM Journal on Applied Dynamical Systems*, vol. 19, no. 2, pp. 886–917, 2020.

[148] J. R. Blough, *Improving the analysis of operating data on rotating automotive components*. University of Cincinnati, 1998.

[149] H. Vold and J. Leuridan, "High resolution order tracking at extreme slew rates, using kalman tracking filters," tech. rep., SAE Technical Paper, 1993.

[150] A. Brandt, T. Lago, K. Ahlin, and J. Tuma, "Main principles and limitations of current order tracking methods," *Sound and Vibration*, vol. 39, no. 3, pp. 19–22, 2005.

[151] B. Anderson and A. Antoulas, "Rational interpolation and state-variable realizations," *Linear Algebra and its Applications*, vol. 137, pp. 479–509, 1990.

[152] Z. Drmač, S. Gugercin, and C. Beattie, "Quadrature-based vector fitting for discretized h_2 approximation," *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. A625–A652, 2015.

[153] H. Wang and T. Wu, "Knowledge-enhanced deep learning for wind-induced nonlinear structural dynamic analysis," *Journal of Structural Engineering*, vol. 146, no. 11, p. 04020235, 2020.

[154] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[155] A. ElSaid, F. El Jamiy, J. Higgins, B. Wild, and T. Desell, "Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration," *Applied Soft Computing*, vol. 73, pp. 969–991, Dec. 2018.

[156] M. L. Adams, *Rotating machinery vibration: from analysis to troubleshooting*. CRC Press, 2000.

[157] S. A. McInerny and Y. Dai, "Basic vibration signal processing for bearing fault detection," *IEEE Transactions on education*, vol. 46, no. 1, pp. 149–156, 2003.

[158] K. Wang, D. Guo, and P. S. Heyns, "The application of order tracking for vibration analysis of a varying speed rotor with a propagating transverse crack," *Engineering Failure Analysis*, vol. 21, pp. 91–101, 2012.

[159] A. Sapena-Bano, M. Riera-Guasp, R. Puche-Panadero, J. Martinez-Roman, J. Perez-Cruz, and M. Pineda-Sanchez, "Harmonic order tracking analysis: A speed-sensorless method for condition monitoring of wound rotor induction generators," *IEEE Transactions on Industry Applications*, vol. 52, no. 6, pp. 4719–4729, 2016.

[160] Z. Feng, X. Chen, and M. Liang, "Joint envelope and frequency order spectrum analysis based on iterative generalized demodulation for planetary gearbox fault diagnosis under nonstationary conditions," *Mechanical Systems and Signal Processing*, vol. 76, pp. 242–264, 2016.

[161] Y. Li, K. Ding, G. He, and X. Jiao, "Non-stationary vibration feature extraction method based on sparse decomposition and order tracking for gearbox fault diagnosis," *Measurement*, vol. 124, pp. 453–469, 2018.

[162] J. R. Blough, "A survey of dsp methods for rotating machinery analysis, what is needed, what is available," *Journal of sound and vibration*, vol. 262, no. 3, pp. 707–720, 2003.

[163] D. McDonald and M. Gribler, "Digital resampling: a viable alternative for order domain measurements of rotating machinery," in *Proceedings of the 9 th Annual International Modal Analysis Conference, Part*, vol. 2, pp. 15–18, 1991.

[164] M. Krishnan, I. Sever, and P. A. Tarazaga, "Data-driven modeling and analysis of tracked order vibration in turbo-fan engine under different operating conditions," 2021.

[165] M. Krishnan, I. A. Sever, and P. A. Tarazaga, "Determining interdependencies and causation of vibration in aero engines using multiscale cross-correlation analysis," in *Model Validation and Uncertainty Quantification, Volume 3* (Z. Mao, ed.), (Cham), pp. 265–272, Springer International Publishing, 2020.

[166] J. R. Blough, *IMPROVING THE ANALYSIS OF OPERATING DATA ON ROTATING AUTOMOTIVE COMPONENTS*. PhD thesis, Michigan Technological University, 1991.

[167] Y. Yue, L. Feng, and P. Benner, "Reduced-order modelling of parametric systems via interpolation of heterogeneous surrogates," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 6, no. 1, pp. 1–33, 2019.

[168] N. Fonzi, S. L. Brunton, and U. Fasel, "Data-driven nonlinear aeroelastic models of morphing wings for control," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 476, p. 20200079, July 2020. Publisher: Royal Society.

[169] A. J. Mayo, and A. C. Antoulas, "A framework for the solution of the generalized realization problem," *Linear Algebra and Its Applications*, vol. 425, no. 2-3, pp. 634–662, 2007.

[170] M. Berljafa and S. Güttel, "The RKFIT algorithm for nonlinear rational approximation," *SIAM J. Sci. Comput.*, vol. 39, no. 5, pp. 2049–2071, 2017.

[171] Yuji Nakatsukasa, and Oliver Sete, and L. N. Trefethen, "The AAA algorithm for rational approximation," *SIAM J. Sci. Comput.*, vol. 40, pp. A1494–A1522, Dec. 2018.

[172] J. M. Hokanson and C. C. Magruder, "Least squares rational approximation," *arXiv preprint arXiv:1811.12590*, 2018.

[173] J. M. Hokanson, "Projected nonlinear least squares for exponential fitting," *SIAM J. Sci. Comput.*, vol. 39, no. 6, pp. A3107–A3128, 2017.

[174] A. C. Antoulas, C. A. Beattie, and S. Gugercin, *Interpolatory Methods for Model Reduction*. Computational Science & Engineering, Philadelphia, PA: SIAM, 2020.

[175] I. V. Gosea and A. C. Antoulas, "Data-driven model order reduction of quadratic-bilinear systems," *Numerical Linear Algebra with Applications*, vol. 25, no. 6, p. e2200, 2018.

[176] A. C. Rodriguez and S. Gugercin, "The p-aaa algorithm for data driven modeling of parametric dynamical systems," *arXiv preprint arXiv:2003.06536*, 2020.

[177] Y. Zhao, Z. H. Guo, and J. M. Yan, "Vibration signal analysis and fault diagnosis of bogies of the high-speed train based on deep neural networks," *Journal of vibroengineering*, vol. 19, no. 4, pp. 2456–2474, 2017.

[178] B. Peherstorfer and K. Willcox, "Data-driven operator inference for nonintrusive projection-based model reduction," *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 196 – 215, 2016.

[179] "Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 406, p. 132401, 2020.

[180] E. Qian, B. Kramer, A. N. Marques, and K. E. Willcox, *Transform & Learn: A data-driven approach to nonlinear model reduction.*

[181] D. S. Karachalios, I. V. Gosea, and A. C. Antoulas, "On bilinear time-domain identification and reduction in the Loewner framework," in *Model Reduction of Complex Dynamical Systems*, vol. 171 of *International Series of Numerical Mathematics*, pp. 3–30, Birkhäuser, Cham, 2021.

[182] S. D. Fassois and F. P. Kopsaftopoulos, "Statistical time series methods for vibration based structural health monitoring," in *New trends in structural health monitoring*, pp. 209–264, Springer, 2013.

[183] M. Krishnan, I. Sever, S. Gugercin, and P. A. Tarazaga, "Wavelet-based dynamic mode decomposition for tracked order vibration modeling in turbo-fan engine.," *arXiv preprint arXiv:2110.12990*, 2021.

[184] M. Krishnan, I. Sever, and P. A. Tarazaga, "Data-driven modeling and analysis of tracked order vibration in turbo-fan engine under different operating conditions," *arXiv preprint arXiv:2110.12990*, 2021.

[185] A. H. Nayfeh and S. A. Emam, "Exact solution and stability of postbuckling configurations of beams," *Nonlinear Dynamics*, vol. 54, no. 4, pp. 395–408, 2008.

[186] P. Benner, S. Gugercin, and K. Willcox, "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM review*, vol. 57, no. 4, pp. 483–531, 2015.

[187] M. Krishnan, S. Gugercin, and P. A. Tarazaga, "A wavelet-based dynamic mode decomposition for modeling mechanical systems from partial observations," 2021.

[188] C. Tan and B. Kang, "Free vibration of axially loaded, rotating timoshenko shaft systems by the wave-train closure principle," *International Journal of Solids and Structures*, vol. 36, no. 26, pp. 4031–4049, 1999.

Appendices

Appendix A

Toolbox implementation

A.1 Overview of the toolbox

1. Contextual characteristics curve and Visualization
2. Input and parameter extraction
3. Validation
4. Prediction

A.2 Demonstration

A.2.1 Contextual characteristics curve and visualization

The data matrix should consists of 24 sensor data in the order that toolbox was designed for. Please consider looking into the readme file attached with the toolbox.

- **I.A** - Enter the engine name here. Currently there are only two options available: (i) Alpha and (ii) Beta.
- **I.B** - Once you enter the engine name, the toolbox proceeds to automatically divide the whole RPM profile into various runs and plots the smoothed characteristic curve, which is basically the tracked order amplitude against the RPM profile.
- **I.C** - This sub-figure plots the tracked order spectrum on top of the second stage shaft RPM. This is basically a way to visualize the whole tracked order amplitude for the whole run. The challenges associated with modeling the order time series can be discerned from this figure.
- **I.D** - In the second row the user can choose from various runs of the original dataset
- **I.E** - Plots the RPM curve of the selected run
- **I.F** - Plots the results of the contextual characteristic curve (Gaussian mixture model)

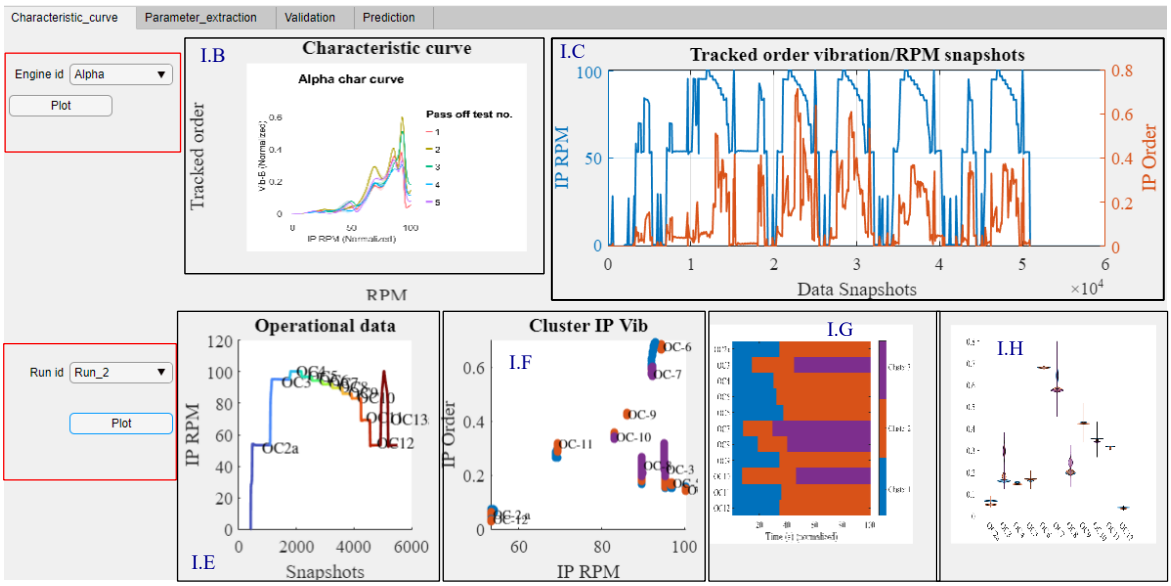


Figure A.1: Characteristics curve and visualization bar

- **I.G** - Plots the various clusters obtained from GMM
- **I.H** - Plots the violin plot of the obtained vibrational clusters.

A.2.2 Parameter extraction

- **II.A** - Pressing the "RUN" starts the parameter extraction program
- **II.B** - Choose the operational condition for visualization purposes. Please not that once you press RUN, the program automatically extracts all the parameters in the OC.
- **II.C** - Parallel plot of the selected OC.
- **II.D** - Comparison of RPM on side by side basis for all the runs.
- **II.E** - Presents a tabulated form of the parallel plot.

A.2.3 Validation

- **III.A** - "Run Prediction" starts the program for computing the data-driven predicted tracked order responses. This program utilizes the input dataset and outputs the tracked order responses.

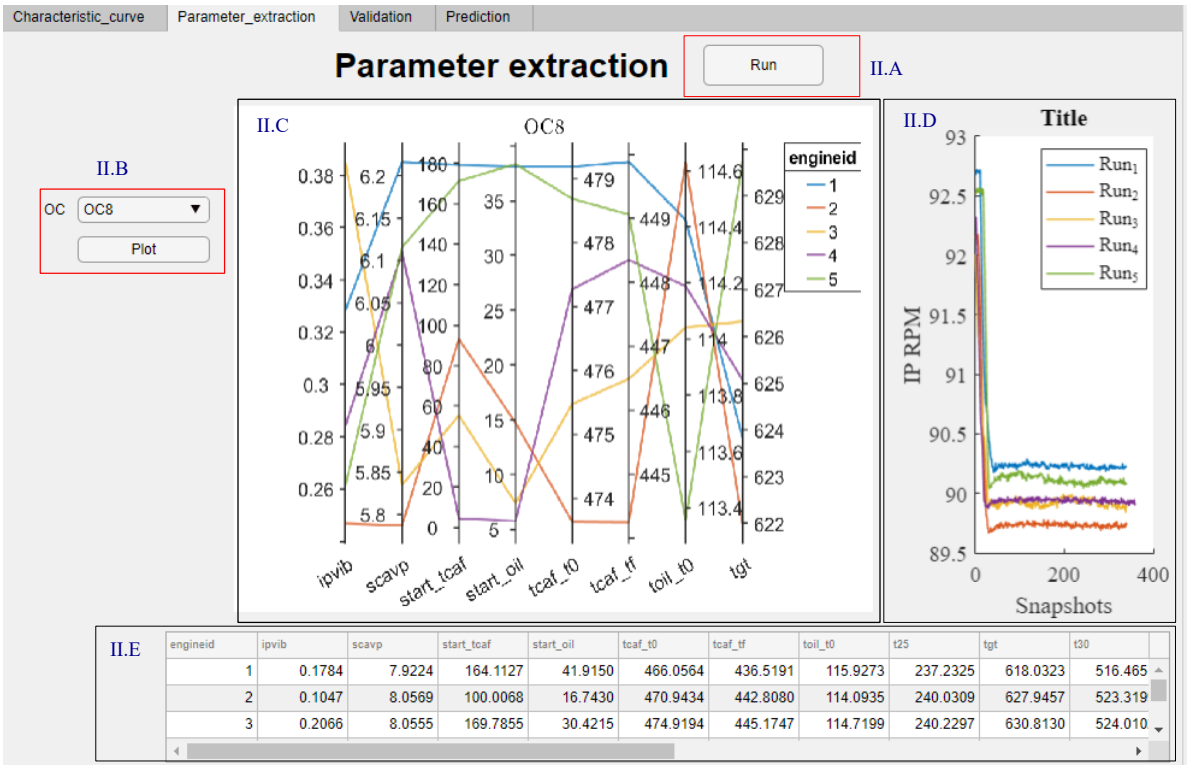


Figure A.2: Input and parameter extraction

- **III.B** - Compares the model predicted response with the original measured tracked order responses.
- **III.C** - Illustrates the error bar chart using our standard error metric.
- **III.D** - Presents the R^2 error for the prediction.

A.2.4 Prediction

- **IV.A** - Given a new running RPM profile, this separates each run and ask user to run plot the RPM curve for that particular engine run. In order for this to work, the user must supply a set of sequence runs. Also keep in mind that each runs should follow the specific sequence from OC2a to OC13.
- **IV.B** - This button executes the program on our interpolatory framework and predicts the tracked order responses.
- **IV.C** - Figure window illustrating the RPM profile.

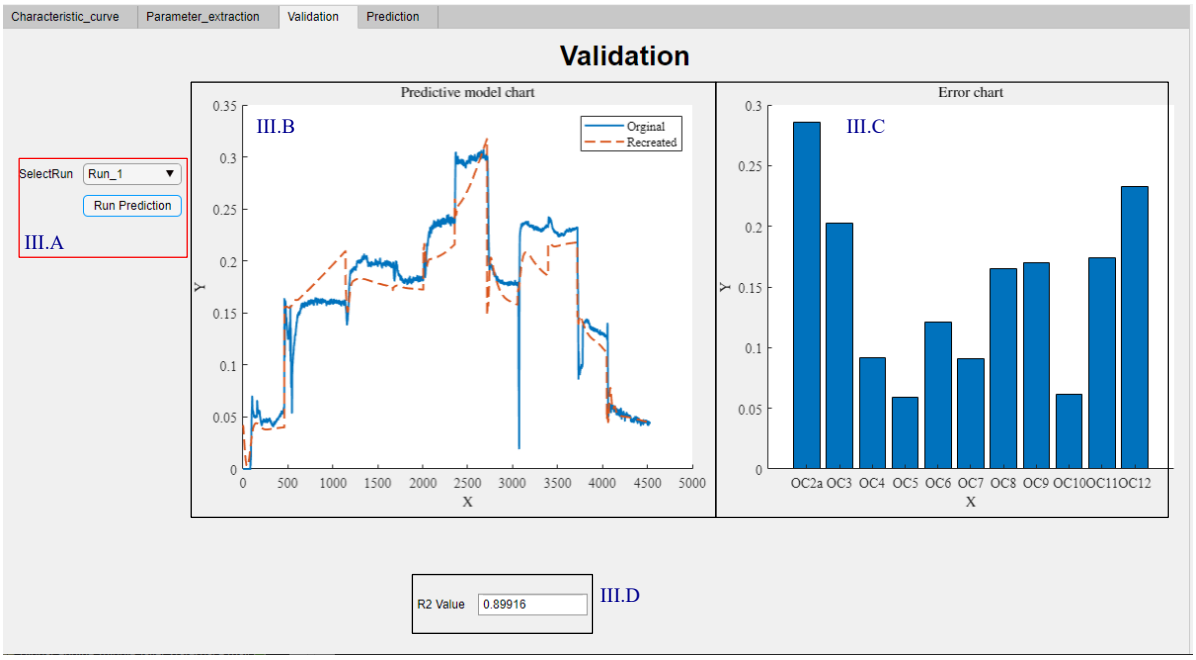


Figure A.3: Validation of the run

- **IV.D** - Figure window illustrating the tracked order responses alongside the RPM profile.

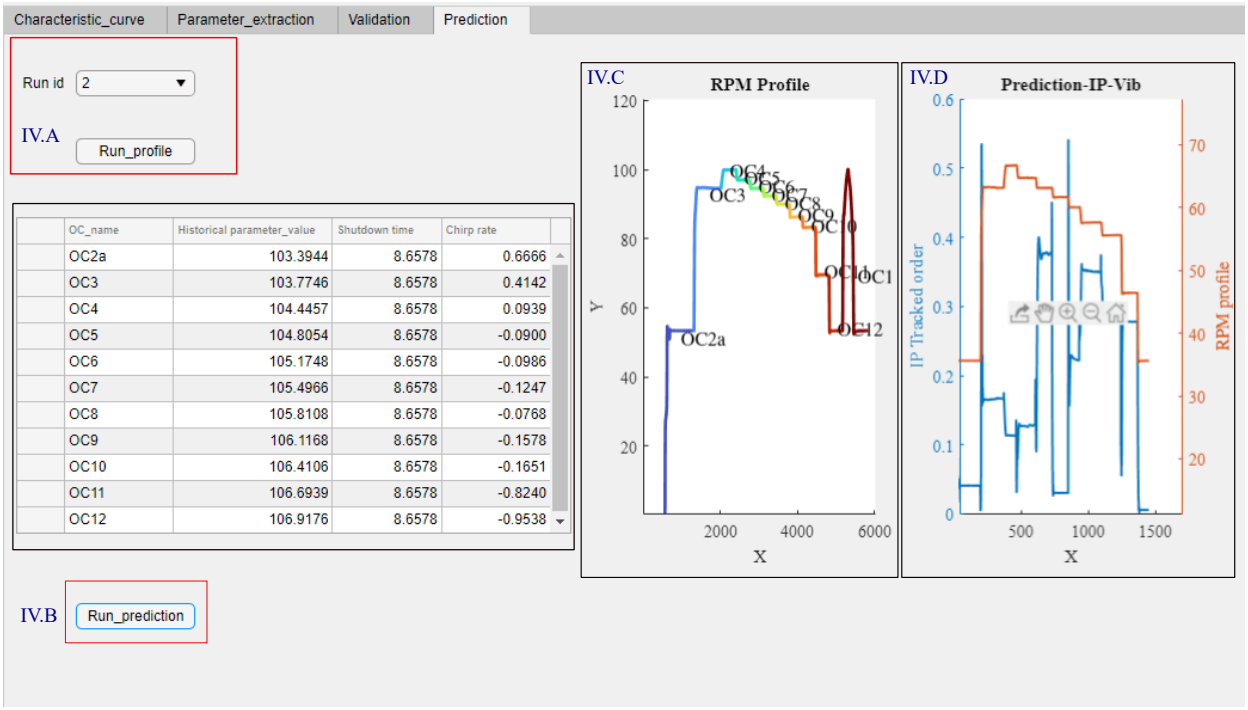
A.2.5 On historical parameter

We define the following parameters which accounts for the history of the engine.

- Cumulative energy parameter : $\varsigma = \int RPM dt$
- Cumulative rate of change of RPM : $\xi = \Sigma \frac{d(RPM)}{dt}$

The cumulative energy parameter is corrected for the shutdown time as, $\varsigma_{norm} = \varsigma - (\frac{4.5 \times 10^5}{12})\theta_{shutdown}$, where $\theta_{shutdown}$ is the shutdown time in hours.

Figure A.5 (b) and Figure A.5 (c) shows the example of how ς and ξ varies for the 3 RPM profile shown in Figure A.5 (a), respectively.



OC_name	Historical parameter_value	Shutdown time	Chirp rate
OC2a	103.3944	8.6578	0.6666
OC3	103.7746	8.6578	0.4142
OC4	104.4457	8.6578	0.0939
OC5	104.8054	8.6578	-0.0900
OC6	105.1748	8.6578	-0.0986
OC7	105.4966	8.6578	-0.1247
OC8	105.8108	8.6578	-0.0768
OC9	106.1168	8.6578	-0.1578
OC10	106.4106	8.6578	-0.1651
OC11	106.6939	8.6578	-0.8240
OC12	106.9176	8.6578	-0.9538

Figure A.4: Prediction

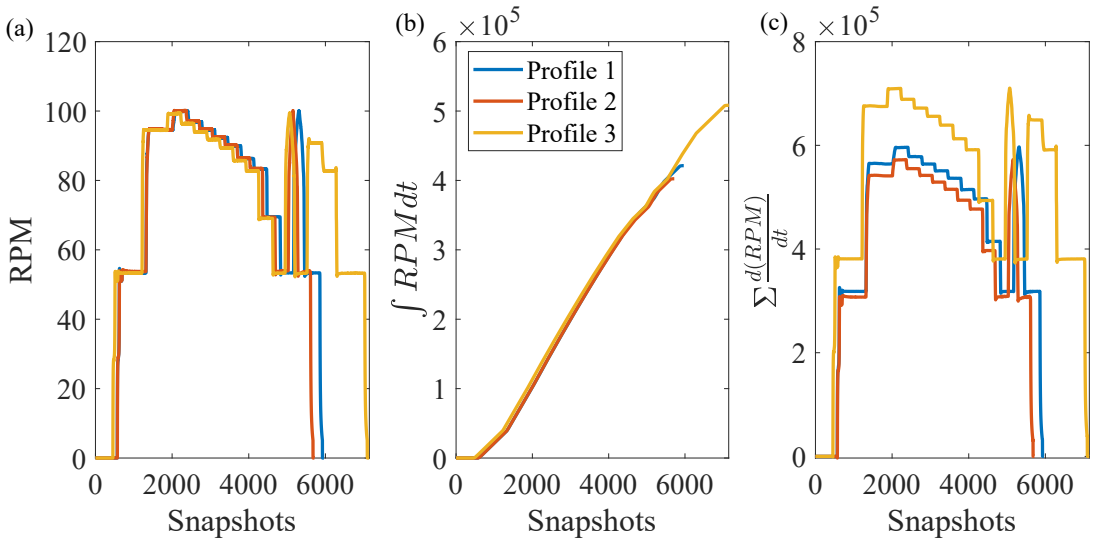


Figure A.5: Historical parameters