University of Sheffield | Mechanical Engineering

MEng Mechanical Engineering

Developing an Open-Source Frequency Domain Modal Analysis Algorithm in Python

Adam AWADALLA

May 2025

Timothy J. Rogers

Report submitted to the University of Sheffield in partial fulfillment of the requirements for the degree of Master of Engineering

Word Count: 8001

Contents

Chapter 1

Introduction

The theoretical and experimental study of structural dynamics has regularly helped engineers grasp the behaviour of systems and structures encountered in everyday life, and has consequently aided in making them safer, lighter, and greener. This ranges from the design of aeroplanes and cars to the stability of buildings as well as the functionality of household appliances like washing machines or air conditioners. Hence, engineers and researchers have been consistently striving to further understand and predict the dynamic characteristics of these various structures. This understanding is crucial for ensuring safety and compliance, performance, and reliability. [1–3]

1.1 Modal Analysis

1.1.1 Theoretical Overview

In linear structural dynamics, Modal Analysis is universally recognized as the pre-eminent solution for the identifying and characterizing structures or systems. It achieves this by studying the structure's modal properties or parameters — its natural frequencies, mode shapes, and damping ratios. Whether using mathematical modelling or experimental testing, modal analysis is key for engineers to understand the response of structures to various excitations or forces, ensuring safety, compliance with standards and regulations, and supporting many research areas such as Structural Health Monitoring or System Identification. [4–6]

Fundamentally, modal analysis is the decomposition of the complex oscillatory behaviour of structures into smaller, more intuitive components called modes. Each mode is a mathematical representation of a specific vibration pattern associated with a natural frequency, the frequency (or set of) at which a system tends to oscillate when displaced, and a corresponding mode shape, a vector which describes the relative movement among the degrees-of-freedom. The damping ratio, a unitless parameter, quantifies the energy dissipation of the system for each mode. The modal properties are defined by the interaction of the system's physical properties, its mass, stiffness, and damping. These inherent properties guide the system's vibration when subject to external forcing or initial displacements or velocities.

The standard procedure for modal analysis begins with forming the equations of motion (EOMs) that represent a system, typically, second order matrix differential equations. Consider a system with an arbitrary number N, degrees-of-freedom (DOFs) as shown in Figure 1.1. Using Newton's second law [7],

$$\sum F_i = m_i \ddot{\mathbf{x}}_i$$

Where F_i is a force acting on the i-th DOF, m_i is the mass, and $\mathbf{x}_i$ is the coordinate, or alternatively the Euler-Lagrange equation as such [7]:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} = Q_i$$

Where q_i is a generalized coordinate (take $q_i = x_i$ for this system), T is the system's kinetic energy, U is the potential energy, and Q_i represents the non-conservative forces. The resulting equations of motion for the system are given in Equation 1.1.

$$\begin{aligned} m_1\ddot{x}_1 + (c_1 + c_2)\dot{x}_1 - c_2\dot{x}_2 + (k_1 + k_2)x_1 - k_2x_2 &= F_1 \\ m_2\ddot{x}_2 + (c_2 + c_3)\dot{x}_1 - c_2\dot{x}_1 - c_3\dot{x}_3 + (k_2 + k_3)x_2 - k_2x_1 - k_3x_3 &= F_2 \\ \ldots \\ m_N\ddot{x}_N + (c_N + c_{N+1})\dot{x}_N - c_{N-1}\dot{x}_{N-1} + (k_N + k_{N+1})x_N - k_{N-1}x_{N-1} &= F_N \end{aligned} \tag{1.1}$$

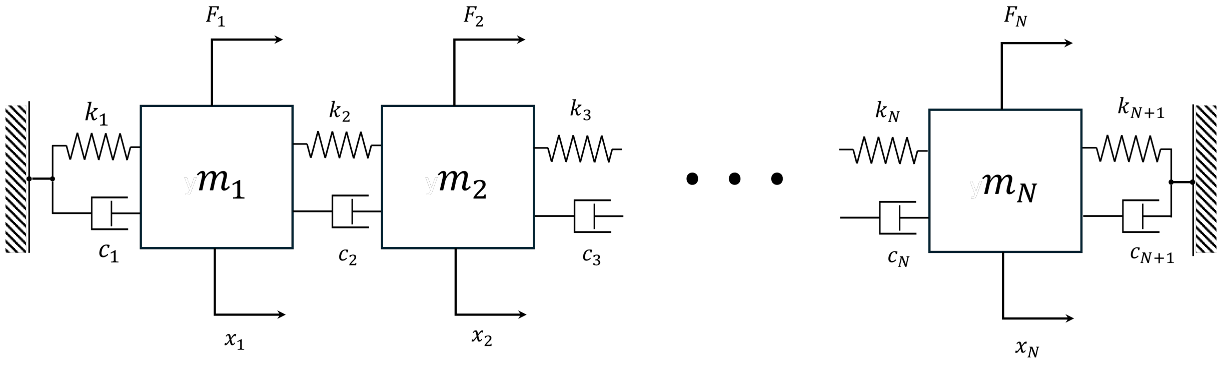

Figure 1.1: A "lumped-mass" system of N degrees-of-freedom

By assembling the physical parameters into respective matrices as highlighted in Equation 1.2, the characteristic differential equation of the system is obtained:

$$\begin{aligned} [\mathbf{K}] &= \begin{bmatrix} k_1 + k_2 & -k_2 & 0 & \ldots & 0 \\ -k_2 & k_2 + k_3 & -k3 & \ldots & 0 \\ 0 & -k_3 & k_3 + k_4 & -k_4 & 0 \\ \ldots & \ldots & \ldots & \ddots & \ldots \\ 0 & 0 & 0 & -k_N & k_N + k_{N+1} \end{bmatrix} \\ [\mathbf{C}] &= \begin{bmatrix} c_1 + c_2 & -c_2 & 0 & \ldots & 0 \\ -c_2 & c_2 + c_3 & -c3 & \ldots & 0 \\ 0 & -c_3 & c_3 + c_4 & -c_4 & 0 \\ \ldots & \ldots & \ldots & \ddots & \ldots \\ 0 & 0 & 0 & -c_N & c_N + c_{N+1} \end{bmatrix} \\ [\mathbf{M}] &= \mathbf{diag}(m_i) \qquad \forall i = 1, 2, \ldots, N \end{aligned} \tag{1.2}$$

$$\therefore\ [\mathbf{M}]\ddot{\mathbf{x}} + [\mathbf{C}]\dot{\mathbf{x}} + [\mathbf{K}]\mathbf{x} = \mathbf{F} \tag{1.3}$$

For simplicity, assume the system presented vibrates freely, and is undamped, so $\mathbf{F}$ and $[\mathbf{C}]$ are zero. If the solution of the presented differential equation is harmonic, i.e. $\mathbf{x}(t) = \Psi e^{j\omega t}$, one finds that equation 1.3 reduces to:

$$([\mathbf{K}] - \omega^2[\mathbf{M}])\Psi = 0 \tag{1.4}$$

Rearranging the terms, the equation into takes the general form of an eigenvalue problem:

$$\begin{aligned}[\mathbf{A}] &= [\mathbf{M}]^{-1}[\mathbf{K}] \\ [\mathbf{A}]\Psi &= \omega^2\Psi\end{aligned} \tag{1.5}$$

Here, ω^2 is the eigenvalue of matrix $\mathbf{A}$ — often referred to as an *eigenfrequency* — and Ψ is the corresponding eigenvector representing the mode shape.

In discrete systems, with a finite number of degrees of freedom, the modal analysis is performed using the matrix $[\mathbf{A}]$, which encapsulates the system's inertial and elastic properties. When extending the approach to systems modelled as continuous mediums, the physical parameters are distributed spatially. This results in a partial differential equation that governs the motion, where finite-dimensional matrices are replaced by linear operators. Consequently, an eigenvalue problem is formed in terms of a differential operator, with eigenvalues corresponding to the natural frequencies and the associated eigenfunctions representing the mode shapes in the spatial domain[1]. For example this principle applied to transverse vibration of an Euler-Bernoulli beam yields the eigenvalue problem:

$$\Delta^2 Y(x) = \beta^4 Y(x) \tag{1.6}$$

where Δ is the Laplacian operator (i.e. second partial derivative in Cartesian coordinate system), and $\beta^4 = \frac{\omega^2}{c^2}$, in which ω is the natural frequency, and c^2 is the ratio of the beam's flexural rigidity and its inertia. [1, 8]

In both discrete and continuous cases, the process of reducing the governing equations into eigenvalue problems is mathematically elegant, robust, and applicable to all linear systems. Despite this approach's elegance, its practical applications rarely exist, as real world structures and systems seldom conform to the idealized assumptions of lumped-mass systems or Euler-Bernoulli beams with known boundary conditions. Thus, academics and practitioners are driven to approach modal analysis differently, to compensate for this limitation.

1.1.2 Modal Analysis in practice, academia, and industry

There are two mainstream modal analysis procedures which address this limitation: Numerical Modal Analysis, and Experimental Modal Analysis, often referred to as Modal Testing. Numerical Modal Analysis is used to simulate dynamic behaviour when modal testing is unimplementable or unnecessary. It involves discretization — typically using finite element modelling — and solving the resulting equations of motion under appropriate initial and boundary conditions in a process similar to that shown in Equations 1.1-1.5. Numerical Modal Analysis offers a faster means of evaluating simpler systems without the need for experimental testing.

In contrast, Modal Testing relies on the principle that, in a mostly linear system, the same modal parameters used to predict the system's response can be obtained from a measurement of

[1]An *eigenvector*, or an *eigenfunction*, of a matrix or linear operator defined on some vector/function space is any non-zero vector/function in said space that when multiplied or acted upon by the matrix/linear operator is equivalent to being multiplied by some scalar factor, said scalar factor is referred to as the *eigenvalue*.

that response. It utilizes various experimental methods such as shaker testing or impact testing to gather response data using sensors like accelerometers or laser vibrometers. Typically, one would pair the experimental set-up with computational algorithms to extract the modal parameters from the test data. An example set-up for a modal test using a shaker is showcased in Figure 1.2. Modal Testing is often the first step for applications such validating models, or measuring the effect of environmental and operational conditions on structures. This underscores why modal testing is typically preferred and more prevalent, as it directly captures the system's response in a way that numerical modal analysis is unable to. Due to the widespread adoption of modal testing in structural dynamics, the term modal analysis is generally interpreted as modal testing, and due to the nature of the work in this report, the term modal analysis will also be referring to modal testing.

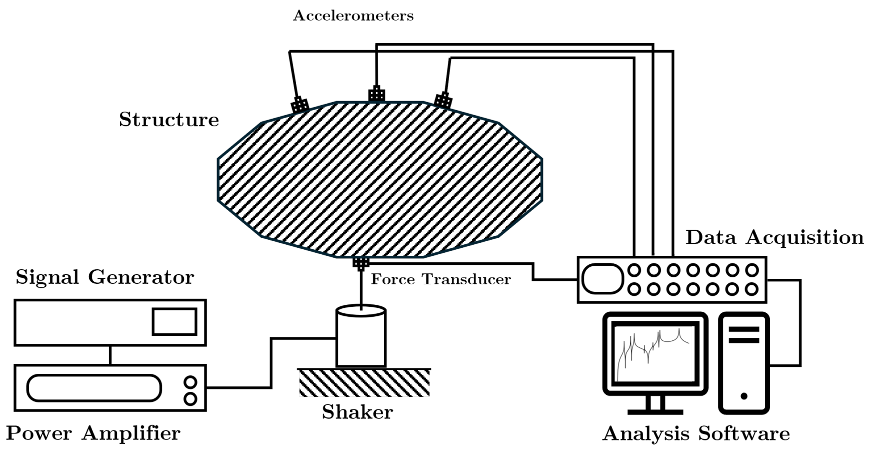

Figure 1.2: Typical electrodynamic shaker experimental set up, adapted from He & Fu [5]

In modern research areas such as structural health monitoring (SHM)—where the primary objective is to observe changes in an asset's structural condition through continuous monitoring—modal analysis is frequently employed because a system's modal parameters are highly sensitive to its physical state [9]. Natural frequencies and mode shapes are considered damage-sensitive features; whether changes result from cracks, corrosion, environmental effects, or fatigue, any structural alteration will modify the physical parameters and lead to a quantifiable change in the modal parameters. By statistically comparing a structure's current modal parameters with its baseline measurements, typically taken when the structure is in good condition, early signs of damage and degradation can be detected. This approach enables the effective monitoring of essential infrastructure, including bridges [10–12], buildings [13], and wind turbines [14, 15], making modal analysis an invaluable tool for ensuring the safety and longevity of many everyday structures. See Figure 1.3, and Figure 1.5 for examples of modal testing in SHM contexts for civil and structural engineering and for aerospace respectively.

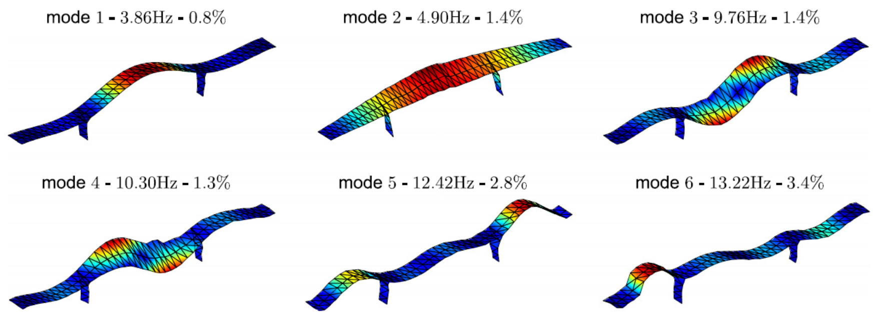

Figure 1.3: Model analysis results of the Z24 bridge where Maeck et al. in [11, 12, 16] performed an experimental campaign on a bridge in Switzerland to benchmark vibration based techniques for damage identification

Modal analysis is widely used across various engineering fields. In the automotive industry, it plays an essential role in enhancing vehicle design, safety, comfort, and performance by analysing responses to factors like axle gear noise, unbalanced loads, and road harshness, thereby optimizing lightweight and high-strength designs [17–21]. In the aerospace sector, detailed dynamic analysis is crucial for balancing structural integrity with weight reduction. Here, modal analysis verifies computational models, controls unwanted vibrations, and evaluates effects such as wind-induced loads and fluid-induced phenomena. Together, these applications provide vital insights that improve overall structural resilience and performance [4, 5, 22–27].

Figure 1.4: Peeters et al. [25], performed modal tests on wind-tunnel model and full sized F16 aircraft

Figure 1.5: BAE systems Hawk T1A aircraft

In civil and structural engineering, modal analysis — particularly Operational Modal Analysis (OMA) — is extensively used to predict and evaluate the behaviour of structures subjected to seismic, crowd, wind, wave, or traffic-induced loading. This technique is critical for assessing tall buildings, dams, and bridges, where accurate response measurements and parameter identification directly inform design decisions that enhance resilience and safety. Ultimately, the insights provided by modal analysis help extend the lifespan of structures and safeguard human lives. [4, 5, 10–13, 28–33]

1.2 Computation in engineering, and the software engineering industry

Building on the earlier discussion that highlighted the integration of computation with modal testing, it is evident that the engineering industry is heavily reliant on software and computational methods. This reliance arises from the extensive volumes of data—such as laboratory results or continuous monitoring outputs—that require processing, as well as from the complexity of calculations that are impractical to perform manually. Common examples include the use of advanced 3D modelling and drawing software, such as Fusion and SolidWorks, programming for data analysis using languages like Python, MATLAB or C++, and various simulation tools like Simulink, ANSYS, or OpenFOAM. It is a reasonable assumption that software usage is indispensable to engineering practices.

Multiple classifications, interpretations, and naming conventions exist among software developers regarding software categorization. To avoid ambiguity in this report, the terms ”proprietary software” and ”open-source software” are defined explicitly. Open-source software (OSS) refers to software licensed to allow free use, modification, and redistribution of both the software and its source code. In contrast, proprietary software denotes software tools or programming environments that require the purchase of a licence for use and restrict access to the source code, thereby preventing modification and redistribution. Previous work [34] presented a more comprehensive comparison of the two software distribution paradigms. To avoid redundancy, only a brief summary of those findings is provided here, followed by an overview of the conclusions and the implications for future work.

The earliest efforts in software commercialization demonstrated considerable foresight, causing a widespread adoption of proprietary tools in engineering industries. These tools gained traction due to rigorous testing, continuous support, adherence to industry standards, and user-friendly interfaces that reduced errors and enhanced usability. Conversely, inherent drawbacks such as high costs, usage restrictions, lack of source code access, potential obsolescence, and privacy concerns have emerged. In contrast, open-source alternatives offer cost-free access, greater user control, and community-driven support, though they often face challenges with limited testing and installation complexity. As a result, selecting between open-source and proprietary solutions requires a careful evaluation of these trade-offs, balancing reliability and robustness against cost and flexibility.

1.3 Project Scope

The pivotal role that modal analysis plays in advancing engineering fields provides motivation for this project. As highlighted in Section 1.1.2, modal analysis is essential for ensuring the comfort, performance, and safety of vehicles and aircraft, as well as for safeguarding critical infrastructure such as buildings, bridges, railway tracks, and offshore structures. Despite its widespread application, many engineers and researchers rely on software tools to perform modal analysis without fully understanding the underlying theory and algorithms, which hinders their ability to interpret results or adapt the methods for unique challenges that are presented quite frequently. This reliance has enabled companies, e.g. Siemens or Structural Vibration Solutions, to charge exorbitant prices for software licences — a situation that thrives in a market dominated by a monopoly of robust solutions. To address this gap, the project introduces an accessible, well-documented, and rigorously tested open-source software tool that produces accurate results, elucidates the fundamental theory, and permits customization for novel applications. The development of a Python library for modal analysis represents the first robust and reliable open-source solution in this domain.

1.3.1 Defining a clear Aim and realistic Objectives

This project is part of a broader development initiative aimed at developing and providing modal-analysis open-source software, making it essential to establish a clear aim that outlines specific objectives and goals. Accordingly, this project contributes to the initiative by integrating and testing the polyreference least-squares complex frequency domain modal parameter estimator (pLSCF, or PolyMAX© commercially) in a Python library. Modal Analysis assists in engineering design decisions for compliance standards, and various research contexts, such as structural health/condition monitoring, and system identification. The software aims reduce uncertainty in data interpretation, allowing engineers and researchers to direct their focus on efficient experimental design and testing. This aim can be effectively achieved through the following objectives, which provide a structured approach to achieving a full implementation of pLSCF in the library.

- Research the pLSCF method, and obtain a complete algorithmic understanding of the method for implementation.
- Integrate pLSCF in the open-source Python library.
- Perform unit tests on the algorithm's individual functions and classes, and test the complete algorithm using simulated and experimental data as a benchmark for the algorithm's efficiency and consistency.
- Perform the same objectives for the Least-Squares Frequency Domain method, which is a simpler, supplementary algorithm used to estimate the mode shapes.
- Help in further supplementary tasks, like signal processing and conditioning, algorithm optimization, and providing documentation for users, which will contribute to the usability and robustness of the published software.

Chapter 2

Software Development

insert introductory paragraph to this chapter

2.1 Modal Analysis Algorithms: mathematical overview

insert something to do with modal curvefitting

2.1.1 Frequency Response Functions and *poles*

The process for obtaining the modal parameters presented in Section 1.1.1 represents a complete solution to the system depicted in Figure 1.1 under unforced conditions. In practical applications, however, an input force is applied to elicit the system response from which the modal parameters are then derived. The response of the structure is typically captured using a numeric construct called the Frequency Response Function (FRF). An FRF is a transfer function, which represents the ratio between a structure's response to the input excitation forces in the frequency-domain [5,35,36]. Revisiting the system's differential equation to include both damping and an external force transforms Equation 1.4 into

$$([\mathbf{K}] + j\nu[\mathbf{C}] - \nu^2[\mathbf{M}])\mathbf{X} = \mathbf{F} \tag{2.1}$$

Where ν denotes the frequency of the excitation force $\mathbf{F}$, and $\mathbf{X}$ represents the system's response. Consequently, the response can be expressed as

$$\mathbf{X} = [H(\nu)]\mathbf{F} \tag{2.2}$$

with the FRF matrix defined as $H(\nu) = ([\mathbf{K}] + j\nu[\mathbf{C}] - \nu^2[\mathbf{M}])^{-1}$. According to the derivation provided in Appendix A.2, the FRF correlating the i^{th} output to the j^{th} input is represented using the system's modal parameters as

$$H_{ij}(\nu) = \frac{X_i}{F_j} = \sum_{k=1}^{N}(\frac{\varphi_{ik}\varphi_{jk}}{\lambda_k^2 - \nu^2}) \tag{2.3}$$

In the free vibration analysis of a damped system, the eigenvalue problem yields the system poles, which are complex numbers encapsulating both the natural frequencies and damping ratios. In the undamped case, these poles reduce to the natural frequencies of the system (set $\zeta = 0$). For each mode, the conjugate pair of poles is given by

$$\lambda_i, \lambda_i^* = -\zeta_i\omega_i \pm j\omega_i\sqrt{1-\zeta_i^2} \tag{2.4}$$

allowing the natural frequencies and damping ratios to be determined from

$$\begin{aligned} \omega_i &= |\lambda_i| \\ \zeta_i &= \frac{\Re(\lambda_i)}{\omega_i} \end{aligned} \tag{2.5}$$

In practice, most deterministic modal analysis algorithms yield a set of system poles that encapsulate the structure's dynamic characteristics. These poles are evaluated against specific stability criteria to determine their physical relevance. Stable poles, which meet the required conditions, represent valid dynamic modes, while unstable poles may indicate unphysical results or simply artefacts of overfitted models.

Stabilisation Diagrams

Practitioners typically assess the stability of poles using a stabilisation diagram. An example stabilisation plot is shown in Figure 2.1.

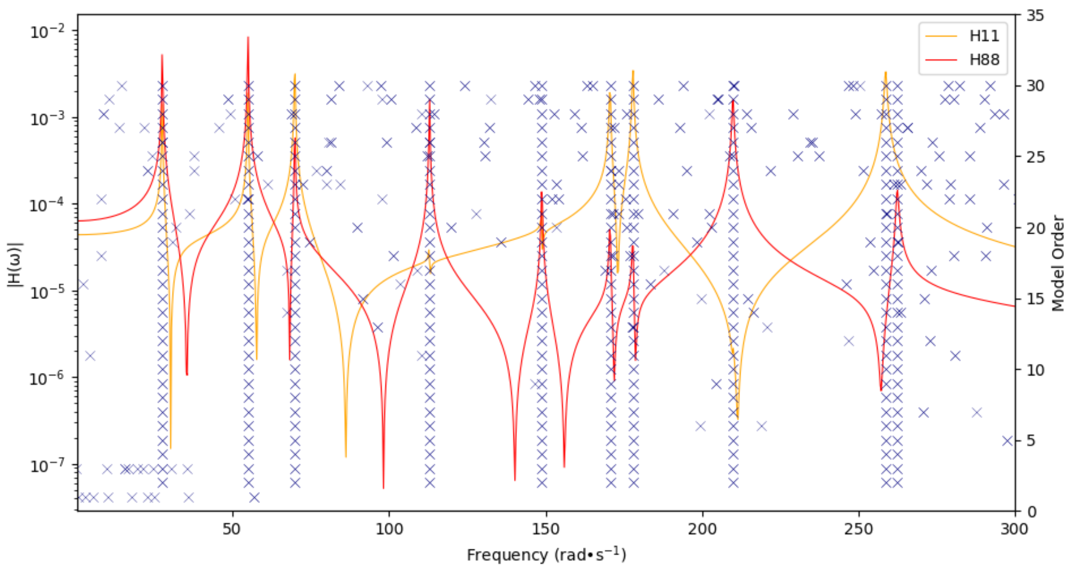

Figure 2.1: Stabilisation diagram for a 10 degree-of-freedom oscillator, for model orders between 1 and 30. Poles are marked as 'X' on the plot.
H_{11} denotes the FRF of x_1 and f_1, same for H_{88}.

In modal curve-fitting, algorithms typically fit the analytical functions based on an expected number of modes, or model order. Stabilisation diagrams are used to find the model order which provides the most stable poles based on the stability criteria. This is done by finding the poles for all the model orders within a desired range and plotting the stability diagram. For linear systems, a stability criteria can look like:

1. $Re(\lambda) < 0$, a positive real part of a root means that either the natural frequency or damping ratio are negative, which is unphysical.

2. For an arbitrary model order M, a pole for a specific mode converges on a natural frequency and damping ratio as $M \to \infty$, within a certain tolerance interval.

2.1.2 Least Squares Complex Exponential (LSCE) Method

The LSCE method is a time-domain modal curve-fitting algorithm that correlates the **I**mpulse **R**esponse **F**unction (FRF in the time-domain) of a multi-degree-of-freedom system with its poles and modal residues using a complex exponential. An autoregressive (AR) model is subsequently constructed, and its solution produces a polynomial whose roots correspond to the system's complex poles [5, 37]. Although the LSCE method was quite popular due to its wide range of applicability, it had a certain number of drawbacks, which motivated the need for better methods;

- Although it is polyreference in nature, it does not always perform well with a large number of references (input excitation)
- The method is not well suited for noisy data, which results in unclear stabilisation plots, and inaccuracy in modal parameter estimation.

2.1.3 Least Squares Complex Frequency Domain

A new method that counters the drawbacks of the LSCE method was proposed, a frequency-domain implementation of the method. The LSCF method uses a common denominator transfer function model, to fit a weighted least-squares estimate FRF to measured FRF data [37, 38].

$$H_{ij}(\omega) = \frac{N_{ij}(\omega)}{d(\omega)} \tag{2.6}$$

Where d is a denominator polynomial, and N is a numerator matrix polynomial. Finding a cost function, and solving a set of normal equation yields the polynomial coefficients for the denominator, similar to LSCE, the roots of that polynomial represent the complex poles of the system. The LSCF method offers multiple benefits to the LSCE:

- The use of a weighted regression, which eliminates the effects of noise in the measurement chain, by favouring data points with less variance.
- The LSCF method provides "fast stabilising" diagrams, i.e. the poles converge quicker. A comparison of a stabilisation diagram from both LSCE and LSCF is showcased in Figure 2.2.

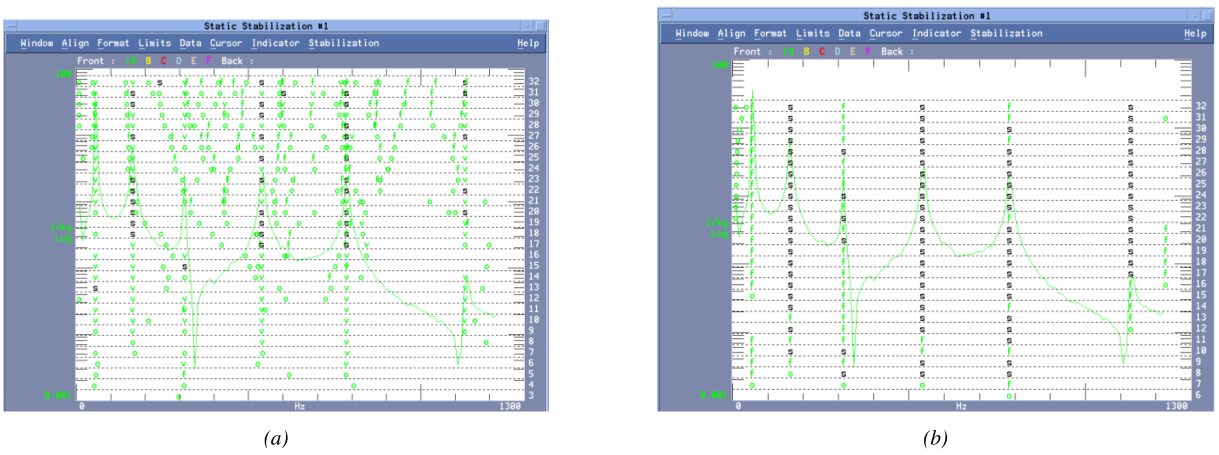

Figure 2.2: Stabilisation Diagrams from LSCE time-domain algorithm (a), and LSCF frequency domain algorithm (b) from Guillaume et al. [38]. Stable poles are marked as black 's' on chart, while spurious or unstable poles are marked in green.

2.1.4 The polyreference implementation of LSCF

By closely examining the derivation of the LSCF method proposed by Guillaume et al. in [37,38], it is apparent that the method isn't inherently polyreference. In practice, this means the algorithm fits each input-output pair individually, a process that becomes computationally taxing when the number of inputs and outputs is large. This inefficiency motivated the development of a polyreference implementation. Peeters et al. proposed such an implementation, known as 'PolyMAX' [39–42]. The polyreference LSCF (pLSCF) method offers several key benefits: it enhances computational efficiency, enables the separation of closely spaced modes, and eliminates the need for the Singular Value Decomposition of the modal residues. The last advantage is discussed in depth in Section 2.1.5.

The pLSCF method adopts a right matrix fractional polynomial model for fitting measured FRF data, the following is a brief overview of the algorithms process, a full derivation is available in B:

$$[H(\omega)] = [N(\omega)][D(\omega)]^{-1} \tag{2.7}$$

Such that $H(\omega) \in \mathbb{C}^{N_{outputs} \times N_{inputs}}$ is the FRF matrix, where $D(\omega) \in \mathbb{C}^{N_{inputs} \times N_{inputs}}$, is the denominator matrix polynomial, and $N(\omega) \in \mathbb{C}^{N_{outputs} \times N_{inputs}}$, is the numerator matrix polynomial. The rows corresponding to each output o in the FRF matrix can be represented as such:

$$\langle H_o(\omega)\rangle = \langle N_o(\omega)\rangle\, [D(\omega)]^{-1} \tag{2.8}$$

The row vector numerator polynomial for the o^{th} output, and the denominator matrix polynomial are defined in terms of a polynomial basis function, $\Omega(\omega)$, and their respective polynomial coefficients, β and α as such:

$$\langle N_o(\omega)\rangle = \sum_{r=1}^{p} \Omega_r(\omega)\, \langle \beta_{or}(\omega)\rangle \tag{2.9}$$

$$[D(\omega)] = \sum_{r=1}^{p} \Omega_r(\omega)[\alpha_r] \tag{2.10}$$

With the polynomial basis function $\Omega_r(\omega) = e^{j\omega\Delta tr}$. Although not initially obvious as polynomials with conventional form $p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n$, the basis functions are expressed in the s-domain where $s = e^{j\omega\Delta t}$. The polynomial coefficients, $\alpha_r \in \mathbb{R}^{N_{inputs} \times N_{inputs}}$ and $\beta_{or} \in \mathbb{R}^{1 \times N_{inputs}}$, are assembled into matrix form.

$$\theta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \cdots \\ \beta_{N_o} \\ \alpha \end{pmatrix} \in \mathbb{R}^{(N_{outputs}+N_{inputs})(p+1)\times N_{inputs}} \tag{2.11}$$

The weighted nonlinear least squares error is defined by the difference of the model FRF, $H(\omega_k)$, and the measured FRF, $\hat{H}(\omega_k)$, as

$$\epsilon_o^{NLS}(\theta, \omega_k) = w_o(\omega_k)(H_o(\omega_k) - \hat{H}_o(\omega_k)) \tag{2.12}$$

This error produces the nonlinear least squares cost function.

$$l^{NLS}(\theta) = \sum_{o=1}^{N_{out}} \sum_{k=1}^{N_f} \mathbf{tr}\{(\epsilon_o^{NLS}(\theta, \omega_k))^H \epsilon_o^{NLS}(\theta, \omega_k)\} \tag{2.13}$$

Where, $\bullet^H$ is the Hermitian transpose of a matrix, and $\mathbf{tr}\{\bullet\}$ is its trace. A subsequent linearisation is performed to approximate this least squares problem, through right-multiplying the error term by the denominator polynomial D. The linearisation results in a cost function in terms of the polynomial coefficients:

$$l^{LS}(\theta) = \mathbf{tr}\{\theta^T J^H J \theta\} \tag{2.14}$$

In Equation 2.14, the term J is the Jacobian matrix, which contains values relating to the frequency content, "x-axis", of the measurement and the response, "y-axis", of the measurement. Minimising the cost function gives the normal equation:

$$2Re(J^H J)\theta = 0 \tag{2.15}$$

Polynomial coefficients, α and β are obtained by imposing a constraint on the system equations and solving the resulting linear system $\mathbf{A} \cdot \mathbf{x} = \mathbf{B}$. When fitting theoretical transfer functions to measured data, the mathematical tools themselves rarely present difficulties; indeed, extracting a polynomial from FRF measurements is straightforward using the pLSCF method. The true challenge lies in constructing a model that remains physically meaningful. Ultimately, pLSCF seeks to determine modal parameters via the system poles. One must return to the fundamental definition of a system pole — the frequency at which the response becomes infinite — and thus enforce the condition that the denominator polynomial equals zero. It is quickly apparent that the poles are the roots of the polynomial D. Using the coefficients of the denominator polynomial, the roots can be found using the so-called companion matrix. The companion matrix of a matrix polynomial can be defined as:

$$C = \begin{pmatrix} 0 & I & \dots & 0 & 0 \\ 0 & 0 & I & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & I \\ -\alpha_0^T & -\alpha_1^T & -\alpha_2^T & \dots - \alpha_{p-2}^T & -\alpha_{p-1}^T \end{pmatrix} \tag{2.16}$$

An interesting property of companion matrices is that its eigenvalues are the roots of its polynomial, as such, the poles and the modal participation factors can be found using an eigendecomposition of the companion matrix.

$$CQ = \Lambda Q \tag{2.17}$$

Where Q is the eigenmatrix, and contains the modal participation factors, and Λ is a diagonal matrix which has the system's discrete time poles on its diagonal entries. A model of polynomial order p will give $N_{inputs} \cdot p$ number of poles and $Q \in \mathbb{C}^{N_{inputs} \cdot p \times N_{inputs} \cdot p}$ participation factors.

2.1.5 Modeshape estimation using LSFD

In practice, modeshapes are estimated using the Least Squares Frequency Domain algorithm, which is a linear regression designed to find the modal residues, a product of the modeshapes and the participation factors, and the upper and lower residuals, values which account for non-ideal vibratory behaviour. The LSFD fits a new model to measured FRF data as shown in Equation 2.18.

$$H(s) = \sum_{m=1}^{N_{modes}} \left(\frac{\Psi_m \cdot L_m^T}{s - \lambda_m} + \frac{\Psi_m^* \cdot L_m^H}{s - \lambda_m^*} \right) + \frac{\mathbf{LR}}{s^2} + \mathbf{UR} \tag{2.18}$$

Where Ψ_m is the modeshape vector for the mode m, L_m is the mode's participation factor, $\mathbf{LR}$ and $\mathbf{UR}$ are the upper and lower residuals, and $s = j\omega$.

2.2 Implementing pLSCF

For all these subsections follow this process:

- The problem/importance of this.
- How it was done.
- What this resulted in. and what that means for the user/engineer.

2.2.1 Project environment: software development practices

The overarching aim of the project is the development of Python library for structural dynamics and modal analysis. However, it is impractical for one individual to develop a comprehensive and robust library for such a large scope, and hence the focused aim of implementing and testing one popular algorithm. The library is being developed collaboratively by academics, and students as part of a large coordinated effort. In such large-scale projects, effective management is needed to prevent disruptive mishaps to the codebase and to check that the contributors are using an up-to-date version of the source code. Version control systems such as Git

- Involving many people. = problem.
- Using Git, and an Agile environment of Continuous improvement, testing, and integration.
- Results in a more streamlined process which makes development quicker, and more efficient.

2.2.2 Software Testing

- Consistently working algorithms. = not always guaranteed.
- Explain the unit test, and the unit testing performed for the functions. (use an intuitive one like the companion matrix)
- this means that by having a standardized input-output system, one can then optimize the algorithm, make it quicker, or more accessible, and have that benchmark ready to check that the algorithm is correctly functioning.
- then go into integration testing. after all the unit tests pass individually, you'd have to then test that they all work in unison.

2.2.3 Optimisation: Time complexities

- Briefly describe what a time and space complexity using big O notation.
- Small measure of what the performance might look like, and not indicative of how much time.
- Big datasets to analyse = big computation time and ram requirements = big problem if ur testing an entire plane or bridge.
- using Numpy broadcasting helps remove overhead.
- Using BLAS packages also helps, as this goes to the bare-bones of the machine and directly uses the processing unit.

2.2.4 Code readability and usability

- Written code is not always understandable. It is a known that code is more often read than made.
- Docstrings for functions, and classes, and sometimes files if the code there doesn't class as any.
- this means that the reader has:
  - An explanation of what each little thing does.
  - A scientific reference like a journal publication or maybe just an online article.
- PEP-8 for naming conventions, using capitalization, underscores etc.
- makes it seem professional, and also other developers can sort of understand some thought processes better.

maths-y bit

- Frequency response function
- dynamical system poles
- logic/control flow
- complexities and big O notation
- unit testing and integration testing.
- Version control.
- lsce (brief)
- lscf (brief)
- plscf (important bits and refer to appendix for full derivation, use own notation and wording).
- Why the companion matrix solution works. $\mathbf{eig}(C(p)) = \lambda_i \rightarrow p(\lambda) = 0$
- here one must explain what poles are for a system.
- lsfd for modeshapes.

software-y bit

- time complexity optimization. discuss that O notation doesn't always equal less time.
- unit testing, explain pytest stuff and fixturing and bla bla bla.
- using numpy (BLAS routines/subroutines)
- user facing code.
- formatting guidelines.

Chapter 3

Results

3.1 Simulated systems: 10 degree-of-freedom oscillator

To benchmark pLSCF, a number of lumped mass systems, as well as experimental data was used. Consider a 10 degree-of-freedom oscillator, with physical parameters:

$$\mathbf{M} = \mathbf{diag}(1,2,3,4,5,1,2,3,4,5)kg$$

$$\mathbf{K} = \begin{pmatrix} 50 & -30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -30 & 70 & -40 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -40 & 50 & -10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -10 & 70 & -60 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -60 & 80 & -20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -20 & 50 & -30 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -30 & 70 & -40 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -40 & 90 & -50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -50 & 80 & -30 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30 & 70 \end{pmatrix} kN \cdot m^{-1}$$

$$\mathbf{C} = \mathbf{K} \cdot 10^{-5} Ns \cdot m^{-1}$$

Following the calculation of the modal properties, the analytical FRF is found using equation 2.3. This synthetic data is passed through the implemented pLSCF algorithm and a stabilisation diagram is constructed for a maximum model order of 30, Figure 3.1 showcases the diagram.

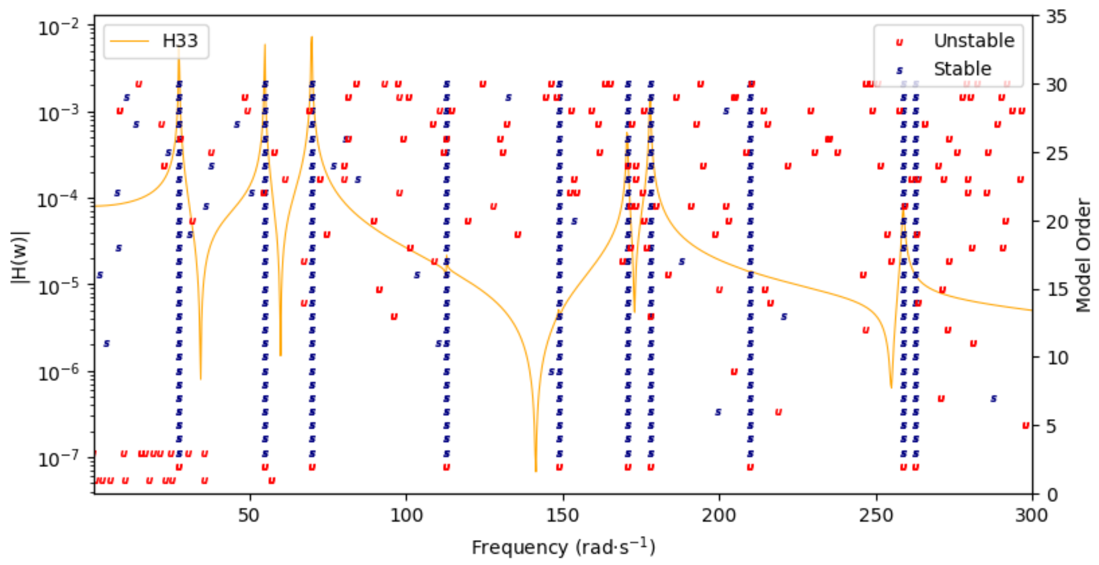

Figure 3.1: Stabilisation of the direct FRF of x_3, stability criteria used (1) tolerance of 0.2rad/s in natural frequency, and 0.02 in damping ratio for increasing model order. (2) $Re(\lambda) < 0$

- Describe that this simulation was done using the equations from the whatever section.
- Stabilisation plot.
- Modeshape plot.
- Compare both to theoretically calculated values.

3.1.1 Introducing noise into the measurements

- Introduce Gaussian (normal distribution) white noise into the system.
- Again, stabilisation plot and modeshapes.
- Compare both to theoretical values again.

3.2 Experimentally tested structure

- Either use MEC444, or Hawk Data.
- Showcase how quick it is idk.

Chapter 4

Discussion, and Potential Improvement

- Now that polymax is implemented, now do we publish.
- A ”beta” version can be published under the provision that there will be lots of improvement.
- More modal analysis algorithms that are for specific cases etc.
- Better user-friendliness/API/frontend.

Appendix A

Damped Multi-degree-of-freedom systems

A.1 Free Vibration Solution

A.2 Forced Vibration Solution

Appendix B

Derivation of p-LSCF modal parameter estimation method

B.1 The right-matrix rational fractional model

The polyreference least-squares complex frequency domain method employs a right matrix fractional model to fit MIMO Frequency Response Function measurements into a set of rational polynomial transfer functions:

$$[H(\omega)] = [N(\omega)][D(\omega)]^{-1} \tag{B.1}$$

Such that $H(\omega) \in \mathbb{C}^{N_{outputs} \times N_{inputs}}$ is the FRF matrix, where $D(\omega) \in \mathbb{C}^{N_{inputs} \times N_{inputs}}$, is the denominator matrix polynomial, and $N(\omega) \in \mathbb{C}^{N_{outputs} \times N_{inputs}}$, is the numerator matrix polynomial. The rows corresponding to each output o in the FRF matrix can be represented as such:

$$\langle H_o(\omega) \rangle = \langle N_o(\omega) \rangle \, [D(\omega)]^{-1} \tag{B.2}$$

The row vector numerator polynomial for the o^{th} output, and the denominator matrix polynomial are defined in terms of a polynomial basis function, $\Omega(\omega)$, and their respective polynomial coefficients, β and α as such:

$$\langle N_o(\omega) \rangle = \sum_{r=1}^{p} \Omega_r(\omega) \, \langle \beta_{or}(\omega) \rangle \tag{B.3}$$

$$[D(\omega)] = \sum_{r=1}^{p} \Omega_r(\omega) [\alpha_r] \tag{B.4}$$

With the polynomial basis function $\Omega_r(\omega) = e^{j\omega\Delta tr}$. Although not initially obvious as polynomials with conventional form $p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n$, the basis functions are expressed in the s-domain where $s = e^{j\omega\Delta t}$. The polynomial coefficients, $\alpha_r \in \mathbb{R}^{N_{inputs} \times N_{inputs}}$ and $\beta_{or} \in \mathbb{R}^{1 \times N_{inputs}}$, are assembled into matrix form:

$$\beta_o = \begin{pmatrix} \beta_{o0} \\ \beta_{o1} \\ \beta_{o2} \\ \cdots \\ \beta_{op} \end{pmatrix} \in \mathbb{R}^{(p+1) \times N_{inputs}} \tag{B.5}$$

$$\alpha = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \cdots \\ \alpha_p \end{pmatrix} \in \mathbb{R}^{N_{inputs}*(p+1)\times N_{inputs}} \tag{B.6}$$

$$\theta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \cdots \\ \beta_{N_o} \\ \alpha \end{pmatrix} \in \mathbb{R}^{(N_{outputs}+N_{inputs})(p+1)\times N_{inputs}} \tag{B.7}$$

B.2 Minimizing the sum of the squared residuals

The collection of both sets of coefficients into one variable θ, makes performing the least squares problem simpler, in a sense, as it becomes the one unknown in this least squares model. As typical in any fitting method, one must minimize the error between the model and the real or measured value. The nonlinear least-squares error for:

Measured FRF:	$\hat{H}_o(\omega_k)$
Model FRF:	$H_o(\omega_k)$

is weighted such that:

$$\epsilon_o^{NLS}(\theta, \omega_k) = w_o(\omega_k)(H_o(\omega_k) - \hat{H}_o(\omega_k)) \tag{B.8}$$

Where $\epsilon_o^{NLS} \in \mathbb{C}^{1\times N_{inputs}}$, $w_o(\omega_k)$ is a scalar weighing function which captures the variation and deviation between multiple inputs on the same measurement point, and $\forall k = 0, 1, 2, \ldots, N_{frequency}$. Said weighing function is typically denoted by

$$w_o(\omega_k) = \frac{1}{\sqrt{\mathbf{var}[H_o(\omega_k)]}} \tag{B.9}$$

(See [reference for weighted linear regressions] for more information on weighted least squares.) One can then define the nonlinear cost function as the sum of the error "squared", (hermitian inner product), over the data points, in this case, spectral lines and outputs;

$$l^{NLS}(\theta) = \sum_{o=1}^{N_{out}} \sum_{k=1}^{N_f} \mathbf{tr}\{(\epsilon_o^{NLS}(\theta, \omega_k))^H \epsilon_o^{NLS}(\theta, \omega_k)\} \tag{B.10}$$

In this equation, $\mathbf{tr}\{\bullet\}$ denotes the trace of a matrix, also known as the sum of diagonal elements, and $\bullet^H$ denotes the Hermitian (conjugate) transpose. The trace operator is used as the trace of a product of 2 matrices $\mathbf{A} \in \mathbb{C}^{m\times n}$ and $\mathbf{B} \in \mathbb{C}^{m\times n}$ will equal the sum of each individual element in $\mathbf{A}$ with the individual elements of $\mathbf{B}$. This provides a sum of all square residuals/errors in the cost function.

$$\mathbf{tr}\{\mathbf{A}^H\mathbf{B}\} = \mathbf{tr}\{\mathbf{A}\mathbf{B}^H\} = \mathbf{tr}\{\mathbf{B}^H\mathbf{A}\} = \mathbf{tr}\{\mathbf{B}\mathbf{A}^H\} = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}b_{ij} \tag{B.11}$$

B.3 Linearizing the error

One can then obtain the polynomial coefficients through minimizing the cost function in B.10, by setting the derivative $\frac{\partial l^{NLS}}{\partial \theta}$ equal to zero, however a nonlinear cost function will yield nonlinear derivative equations, (typically called normal equations in linear regression). A subsequent linearization of the cost function can approximate (suboptimally) the least squares problem, this is achieved through right multiplying the cost function with denominator polynomial $\mathbf{D}$. This gives a linear error:

$$\begin{aligned}\epsilon_o^{LS}(\omega_k,\theta) &= w_o(\omega_k)(N_o(\omega_k,\beta_o) - \hat{H}_o(\omega_k)D(\omega_k,\alpha)) \\ &= w_o(\omega_k)\sum_{r=0}^{p}(\Omega_r(\omega_k)\beta_{or} - \Omega_r(\omega_k)\hat{H}_o(\omega_k)\alpha_r)\end{aligned} \tag{B.12}$$

Stacking the error in terms for all spectral lines in one matrix $E_o^{LS}(\theta) \in \mathbb{C}^{N_f \times N_{in}}$:

$$E_o^{LS}(\theta) = \begin{pmatrix} \epsilon_o^{LS}(\omega_1,\theta) \\ \epsilon_o^{LS}(\omega_2,\theta) \\ \epsilon_o^{LS}(\omega_3,\theta) \\ \dots \\ \epsilon_o^{LS}(\omega_{N_f},\theta) \end{pmatrix} = \begin{pmatrix} X_o & Y_o \end{pmatrix}\begin{pmatrix} \beta_o \\ \alpha \end{pmatrix} \tag{B.13}$$

Here, new variables $\mathbf{X}$ and $\mathbf{Y}$ are introduced:

$$X_o = \begin{pmatrix} w_o(\omega_1)\Big(\Omega_0(\omega_1) + \Omega_1(\omega_1)\dots\Omega_p(\omega_1)\Big) \\ w_o(\omega_2)\Big(\Omega_0(\omega_2) + \Omega_1(\omega_2)\dots\Omega_p(\omega_2)\Big) \\ \vdots \\ w_o(\omega_{N_f})\Big(\Omega_0(\omega_{N_f}) + \Omega_1(\omega_{N_f})\dots\Omega_p(\omega_{N_f})\Big) \end{pmatrix} \in \mathbb{C}^{N_f \times (p+1)} \tag{B.14}$$

$$Y_o = \begin{pmatrix} -w_o(\omega_1)\Big(\Omega_0(\omega_1) + \Omega_1(\omega_1)\dots\Omega_p(\omega_1)\Big) \otimes \hat{H}_o(\omega_1) \\ -w_o(\omega_2)\Big(\Omega_0(\omega_2) + \Omega_1(\omega_2)\dots\Omega_p(\omega_2)\Big) \otimes \hat{H}_o(\omega_2) \\ \vdots \\ -w_o(\omega_{N_f})\Big(\Omega_0(\omega_{N_f}) + \Omega_1(\omega_{N_f})\dots\Omega_p(\omega_{N_f})\Big) \otimes \hat{H}_o(\omega_{N_f}) \end{pmatrix} \in \mathbb{C}^{N_f \times N_{in}(p+1)} \tag{B.15}$$

Where $\otimes$ is the Kronecker product. In these equations, $\mathbf{X}$ is used to capture the frequency content of the least squares problem, and $\mathbf{Y}$ is used to capture both the frequency content and the measured response data. Using these matrices, one can reconstruct the nonlinear cost function into one that is linear:

$$\begin{aligned} l^{LS}(\theta) &= \sum_{o=1}^{N_{out}}\sum_{k=1}^{N_f}\mathbf{tr}\{(\epsilon_o^{LS}(\omega_k,\theta))^H\epsilon_o^{LS}(\omega_k,\theta)\} \\ &= \sum_{o=1}^{N_{out}}\mathbf{tr}\Big\{(E_o^{LS}(\theta))^H E_o^{LS}(\theta)\Big\} \\ &= \sum_{o=1}^{N_{out}}\mathbf{tr}\Big\{\begin{pmatrix}\beta_o^T & \alpha^T\end{pmatrix}\begin{pmatrix}X_o^H \\ Y_o^H\end{pmatrix}\begin{pmatrix}X_o & Y_o\end{pmatrix}\begin{pmatrix}\beta_o \\ \alpha\end{pmatrix}\Big\} \end{aligned} \tag{B.16}$$

If one defines a *Jacobian* matrix $\mathbf{J} \in \mathbb{C}^{N_f N_{out} \times (N_{in}+N_o ut)(p+1)}$ for the problem as such:

$$\mathbf{J} = \begin{pmatrix} X_1 & 0 & \dots & 0 & Y_1 \\ 0 & X_2 & \dots & 0 & Y_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & X_{N_{out}} & Y_{N_{out}} \end{pmatrix} \tag{B.17}$$

The cost function can be represented as:

$$l^{LS}(\theta) = \mathbf{tr}\{\theta^T \mathbf{J}^H \mathbf{J} \theta\} \tag{B.18}$$

To obtain real values of θ, one must place a constraint on the cost function such that:

$$l^{LS}(\theta) = \mathbf{tr}\{\theta^T Re(\mathbf{J}^H \mathbf{J}) \theta\} \tag{B.19}$$

Where the Gramian matrix of $\mathbf{J}$ can be represented in terms a set of variables, $\mathbf{R}$,$\mathbf{S}$ and $\mathbf{T}$:

$$Re(\mathbf{J}^H \mathbf{J}) = \begin{pmatrix} R_1 & 0 & \dots & 0 & S_1 \\ 0 & 0 & \dots & 0 & S_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & R_{N_{out}} & S_{N_{out}} \\ S_1^T & S_2^T & \dots & S_{N_{out}}^T & \sum_{o=1}^{N_{out}} T_o \end{pmatrix} \tag{B.20}$$

in which:

$$R_o = Re(X_o^H X_o) \tag{B.21}$$

$$S_o = Re(X_o^H Y_o) \tag{B.22}$$

$$T_o = Re(Y_o^H Y_o) \tag{B.23}$$

B.4 The *Normal Equations*, and extracting the modal parameters

The cost function can then be minimized in terms of α and β to find the best least-squares fit:

$$\frac{\partial l^{LS}(\theta)}{\partial \beta_o} = 2(R_o \beta_o + S_o \alpha) = 0 \qquad \forall O = 1, 2, \dots, N_{out} \tag{B.24}$$

$$\frac{\partial l^{LS}(\theta)}{\partial \alpha} = 2 \sum_{o=1}^{N_{out}} (S_o^T \beta_o + T_o \alpha) \tag{B.25}$$

Giving normal equations of this least squares problem in terms of the wanted polynomial coefficients, one can also assemble those normal equations into 1 equation:

$$\frac{\partial l^{LS}(\theta)}{\partial \theta} = 2 Re(\mathbf{J}^H \mathbf{J}) \theta = 0 \tag{B.26}$$

The denominator coefficients α are used to obtain the poles and the modal participation factors, which are sufficient information for the constructing a stabilization diagram. Hence, one can further reduce the normal equations by setting:

$$\beta_o = R_o^{-1} S_o \alpha \tag{B.27}$$

This yields the reduced normal equation:

$$\begin{aligned} \left\{ 2 \sum_{o=1}^{N_{out}} (T_o - S_o^T R_o^{-1} S_o) \right\} \alpha = 0 \\ \mathbf{M}\alpha = 0 \end{aligned} \tag{B.28}$$

For a non-trivial solution to the normal equation, a constraint is set on α, where:

$$\alpha_p = \mathbf{I}_{N_{in}} \tag{B.29}$$

The rest of the denominator coefficients are then found using:

$$\mathbf{M}(1 : N_{in} * p, 1 : N_{in} * p) \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_{p-1} \end{pmatrix} = \mathbf{M}(1 : N_{in} * p, N_{in} * p + 1 : N_{in} * (p+1)) \tag{B.30}$$

The least-squares estimate for α is then:

$$\hat{\alpha}_{LS} = \begin{Bmatrix} \alpha \\ \mathbf{I}_{N_{in}} \end{Bmatrix}$$

This makes the denominator polynomial $\mathbf{D}$ a *monic* polynomial. Based on the fundamental definition of system poles, which is the points at which the system's response is "infinite", one can say that for an arbitrary rational polynomial $p(x)$:

$$p(x) = \frac{a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n}{b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^n} \tag{B.31}$$

To achieve said "infinite" value, one must find values of x that represent the roots of the denominator polynomial. In the pLSCF model, one can exploit the monic property of the denominator polynomial. Frobenius companion matrices are square matrices that represent monic polynomials, given one has a monic polynomial

$$p(x) = x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0$$

The companion matrix of said polynomial is defined as:

$$C(p) = \begin{bmatrix} 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & \dots & 0 & -a_1 \\ 0 & 1 & \dots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -a_{n-1} \end{bmatrix} \tag{B.32}$$

A property of the companion matrix $C(p) \in \mathbb{R}^{n \times n}$ is that its eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ are the roots of $p(x)$, where $p(\lambda_i) = 0, \forall i = 1, 2, \dots, n$. This property aids in finding the system poles, after constructing the companion matrix for the denominator polynomial, an Eigendecomposition of the matrix yields the discrete time poles as the eigenvalues, and the corresponding eigenvectors are the modal participation factors:

$$C(\mathbf{D}) = \begin{pmatrix} 0 & I & \dots & 0 & 0 \\ 0 & 0 & I & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & I \\ -\alpha_0^T & -\alpha_1^T & -\alpha_2^T & \cdots - \alpha_{p-2}^T & -\alpha_{p-1}^T \end{pmatrix} \tag{B.33}$$

$$C(\mathbf{D})[\mathbf{L}] = \Lambda[\mathbf{L}] \tag{B.34}$$

Where the matrix $[\mathbf{L}]$ is the Eigenmatrix (matrix with columns as eigenvectors), representing the modal participation factors of each mode, and the matrix Λ contains the discrete time poles on its diagonal elements. The transpose of a companion matrix, this however does not affect the numerical value of the poles or participation factors [reference proof]. A p-order right matrix-fraction polynomial estimation should yield pN_{in} number of poles.

B.5 Finding the modeshapes using the Least-Squares Frequency Domain (LSFD) method

This part should be in actual report: When fitting theoretical models to experimental data, the difficulty does not typically lie in the mathematical framework enabling the modelling process. Instead, the challenge is in constructing a model that provides physically meaningful insights. Given that the goal of many modal parameter estimation methods is to find a set of poles which As apparent, it is straightforward to find the polynomial coefficients using measured data.

Bibliography

[1] S. S. Rao and P. Griffin, *Mechanical vibrations*, 6th ed. Harlow, England: Pearson, 2018.

[2] N. M. M. Maia, D. D. Maio, A. Carrella, F. Marulo, C. Zang, J. E. Cooper, K. Worden, and T. A. N. Silva, *Structural Dynamics in Engineering Design*. Wiley, 2024.

[3] K. Worden and G. R. Tomlinson, *Nonlinearity in structural dynamics : detection, identification, and modelling*. Bristol [England] ; Philadelphia: Institute of Physics, 2001.

[4] D. Ewins, *Modal Testing: Theory, Practice and Application*, ser. Engineering dynamics series. Wiley, 2000.

[5] Z. Fu and J. He, *Modal Analysis*. Butterworth-Heinemann, 2001.

[6] N. M. M. Maia and J. M. J. M. Montalvao e Silva, *Theoretical and experimental modal analysis*, ser. Mechanical engineering research studies. Engineering dynamics series ; 9. Baldock : New York: Research Studies Press ; Wiley, 1997.

[7] S. T. Thornton and J. B. Marion, *Classical dynamics of particles and systems*. Andover: Cengage Learning, 2014. [Online]. Available: https://www.worldcat.org/title/classical-dynamics-of-particles-and-systems/oclc/951438161&referer=brief_results

[8] R. Blevins, *Formulas for Dynamics, Acoustics and Vibration*. Wiley, 11 2015.

[9] C. Farrar and K. Worden, *Structural Health Monitoring A Machine Learning Perspective*. Wiley, 01 2013.

[10] A. Bunce, D. S. Brennan, A. Ferguson, C. O'Higgins, S. Taylor, E. J. Cross, K. Worden, J. Brownjohn, and D. Hester, "On population-based structural health monitoring for bridges: Comparing similarity metrics and dynamic responses between sets of bridges," *Mechanical Systems and Signal Processing*, vol. 216, p. 111501, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327024003996

[11] J. Maeck, B. Peeters, and G. DeRoeck, "Damage identification on the z24 bridge using vibration monitoring," *Smart Materials and Structures*, vol. 10, pp. 512–517, 06 2001.

[12] B. Peeters and G. De Roeck, "One year monitoring of the z24-bridge: Environmental influences versus damage events," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2, 05 2000.

[13] S. Saeed, S. H. Sajid, and L. Chouinard, "Optimal sensor placement for enhanced efficiency in structural health monitoring of medium-rise buildings," *Sensors*, vol. 24, no. 17, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/17/5687

[14] L. Bull, P. Gardner, J. Gosliga, T. Rogers, N. Dervilis, E. Cross, E. Papatheou, A. Maguire, C. Campos, and K. Worden, “Foundations of population-based shm, part i: Homogeneous populations and forms,” *Mechanical Systems and Signal Processing*, vol. 148, p. 107141, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327020305276

[15] S. Tsiapoki and C. Colomer Segura, “Seven years shm of offshore wind turbine foundations: Review, experiences and outlook,” 2024, Conference paper, cited by: 0; All Open Access, Gold Open Access. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85202581460&doi=10.58286%2f29681&partnerID=40&md5=23be5f28faab56d14121c846f24d1619

[16] J. MAECK and G. DE ROECK, “Description of z24 benchmark,” *Mechanical Systems and Signal Processing*, vol. 17, no. 1, pp. 127–131, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327002915487

[17] Z. Sun, G. C. Steyer, G. Meinhardt, and M. Ranek, “Nvh robustness design of axle systems,” *SAE Transactions*, vol. 112, pp. 1746–1754, 2003. [Online]. Available: http://www.jstor.org/stable/44745550

[18] E. Abe and H. Hagiwara, “Advanced method for reduction in axle gear noise,” *SAE Transactions*, vol. 84, pp. 670–682, 1975. [Online]. Available: http://www.jstor.org/stable/44681958

[19] J. W. Martz, E. L. Peterson, G. W. Knobeloch, and G. D. Angus, “Four steps for vehicle ride improvement,” *SAE Transactions*, vol. 88, pp. 814–827, 1979. [Online]. Available: http://www.jstor.org/stable/44633919

[20] M. French and M. Jay, “An introduction to automotive nvh testing,” *Experimental Techniques*, vol. 22, no. 4, pp. 32–33, 1998. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1747-1567.1998.tb02336.x

[21] Y. Liu, H. Dai, P. Wu, Z. Jing, Y. Song, and T. Li, *Research on Vibration and Noise of SUV Chassis Suspension System*, 2020, pp. 307–310.

[22] J. Wright and J. Cooper, *Introduction to Aircraft Aeroelasticity and Loads*, ser. Aerospace Series. Wiley, 2008. [Online]. Available: https://books.google.co.uk/books?id=BU_tRQaz9gIC

[23] D. A. Chamberlain and C. K. Mechefske, “Experimental modal analysis of a half-scale model twin-engine aircraft rear fuselage engine mount support frame,” *Proceedings of the ASME Design Engineering Technical Conference*, vol. 8, 2017, cited by: 7. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85034862660&doi=10.1115%2fDETC2017-67389&partnerID=40&md5=314237f6ffca59db53004dd5b88f91ce

[24] Z. Saffry, D. L. Majid, F. I. Romli, F. Mustapha, and E. J. Abdullah, “Identification of modal properties of composite thin plate using oma in wind tunnel environment,” in *Advanced Research in Material Science and Mechanical Engineering*, ser. Applied Mechanics and Materials, vol. 446. Trans Tech Publications Ltd, 1 2014, pp. 606–610.

[25] J. Lau, B. Peeters, J. Debille, Q. Guzek, W. Flynn, D. Lange, and T. Kahlmann, *Ground Vibration Testing Master Class: modern testing and analysis concepts applied to an F-16 aircraft*, 04 2011, pp. 221–228.

[26] L. Ünlüsoy, M. Şahin, and Y. Yaman, "Verification of a finite element model of an unmanned aerial vehicle wing torque box via experimental modal testing," vol. 1, 07 2012.

[27] M. Haywood-Alexander, R. S. Mills, M. D. Champneys, M. R. Jones, M. S. Bonney, D. Wagg, and T. J. Rogers, "Full-scale modal testing of a hawk t1a aircraft for benchmarking vibration-based methods," *Journal of Sound and Vibration*, vol. 576, p. 118295, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022460X24000592

[28] C. Rainieri and G. Fabbrocino, *Operational Modal Analysis of Civil Engineering Structures : An Introduction and Guide for Applications*, 1st ed. New York, NY: Springer New York : Imprint: Springer, 2014.

[29] M. Archila, R. Boroschek, C. Ventura, and S. Molnar, *Modal Testing of a Repaired Building After 2010 Chile Earthquake*, 05 2013, pp. 119–125.

[30] M. Abdelnour and V. Zabel, "Identification of closely spaced modes in space truss structures as base for the determination of member normal forces," *Journal of Physics: Conference Series*, vol. 2647, p. 192014, 06 2024.

[31] P. Reynolds, I. Díaz, and D. Nyawako, "Vibration testing and active control of an office floor," 01 2009.

[32] D. Roia, F. Gara, E. Speranza, L. Gioiella, and L. Dezi, "Operational modal analysis on a r.c. building for the evaluation of the dynamic changes due to retrofitting," pp. 119–124, 07 2015.

[33] R. Cantieni, "Experimental methods used in system identification of civil engineering structures," 01 2004.

[34] A. Awadalla, "Developing an open-source frequency domain modal analysis algorithm in python - interim report," School of Mechanical, Aerospace, and Civil Engineering, University of Sheffield, Tech. Rep., 2024.

[35] T. J. Rogers, "Lecture 5 - forced vibration," Lecture Slides for MEC326 Course, Department of Mechanical Engineering, University of Sheffield, 2023, accessed on: Dec. 1st, 2024.

[36] ——, "Lecture 6 - modal properties 2," Lecture Slides for MEC326 Course, Department of Mechanical Engineering, University of Sheffield, 2023, accessed on: Dec. 1st, 2024.

[37] P. Guillaume, P. Verboven, and S. Vanlanduit, "Frequency domain maximum likelihood identification of modal parameters with confidence intervals," 01 1998.

[38] 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," *Proceedings of IMAC*, vol. 21, 01 2003.

[39] B. Peeters, H. Van Der Auweraer, P. Guillaume, and J. Leuridan, "The polymax frequency-domain method: A new standard for modal parameter estimation?" *Shock and Vibration*, vol. 11, no. 3-4, p. 395 – 409, 2004, cited by: 912; All Open Access, Gold Open Access. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-4644323283&doi=10.1155%2f2004%2f523692&partnerID=40&md5=564fa529224241631a3740c34109f97a

[40] B. Peeters, P. Guillaume, H. Van der Auweraer, B. Cauberghe, P. Verboven, and J. Leuridan, “Automotive and aerospace applications of the polymax modal parameter estimation method,” *Proceedings of IMAC 22, the International Modal Analysis Conference*, 01 2004.

[41] B. Peeters and H. Van der Auweraer, “Polymax: a revolution in operational modal analysis,” *1st International Operational Modal Analysis Conference*, 01 2005.

[42] B. Peeters, H. Van der Auweraer, J. Leuridan, and T. Vasel, “Polymax modal parameter estimation: Challenging automotive and aerospace applications,” pp. 1–13+572, 01 2004.