Additively Manufactured Architectured Structures for Controlling Rotordynamics

Zhen Rong Yap

Trinity College

University of Oxford

A thesis submitted for the degree of

Master of Engineering

Trinity 2024

Abstract

Additive manufacturing presents an opportunity to create architectured materials with design parameters with tailorable stiffness and damping. A key application explored are rotordynamic bearing supports, a core component of aerospace engines. Supplemented with analytical models and rotordynamics softwares, a jeffcott rotor rig is used to study the instabilities and behaviour that are affected by different materials including those we printed with laser powder bed fusion (LPBF) in 316SS. Discussion of looseness on the rotordynamic rig is explored by analysis of the FFT spectra. Interpenetrating lattices (IPLs) are presented as to be integrated with the bearing supports and studied in simulation and compressive testing destructively with a hydraulic press and non-destructively with Laser Induced Resonant Acoustic Spectroscopy (LIRAS) to gain data on the mechanical properties, relating them to its design. The findings in this thesis: 1) advise the next steps in integrating the bearing supports and IPLs, 2) verify which models are useful for IPLs, and 3) provide a first look at the ability of LIRAS measurements to study the dynamic response of IPLs.

Thesis Supervisors:

Zachary C. Cordero, Michael P. Moody, Yuanbo Tony Tang

To my loving mother and father.

Acknowledgements

MIT. It has been wonderful.

Finally, I must express my deepest gratitude to my family and friends, whose tireless support, encouragement and sheltering have been my anchor and inspiration throughout this challenging and rewarding journey.

Contents

List of Figures

List of Tables

Engineering Context

1.1 Turbomachinery

Turbomachinery converts the rotational energy of the turbine blades to providing thermal and kinetic energy for the compressed fluid for pressurisation. [1] It is most notably used in aerospace in turbojets for aerogines and turbopumps for rocket engines. However, its usage permeates all aspects of life as turbopumps are also a central piece of vacuum technology that powers electron microscopes, lithography, coating and etching machines (ASML), particle accelerators (CERN) in addition to aerospace systems via 65 years of turbopumps. [2]

Aeroengines

Aeroengines are designed specifically for the unique requirements of aviation, including efficiency at high altitudes, reliability under various atmospheric conditions, and the ability to produce a high thrust-to-weight ratio. Historically, this has increased 7 fold from 1939 [3].

During my visit to the MIT Gas Turbine Laboratory (GTL), I got to see the Power Jets W.1 (Whittle W1) and Junker Juno 004. Engines that powered the British Gloster E.28/39 and the German Messerschmitt Me 262 planes during first introduced during WW2 [4].

Figure 1.1: Power Jets W.1 (Whittle W1) and Junkers Jumo 004 in the MIT Gas Turbine Laboratory.

Junkers Jumo 004's rotordynamics were basic by modern standards. It used simple bearing configurations and an axial flow design that were not optimised for high rotational speeds at the operating temperatures experienced [5]. The Whittle W1 used a centrifugal compressor instead which was simpler mechanically and more robust but experienced greater radial loads and more demand on the bearings [6].
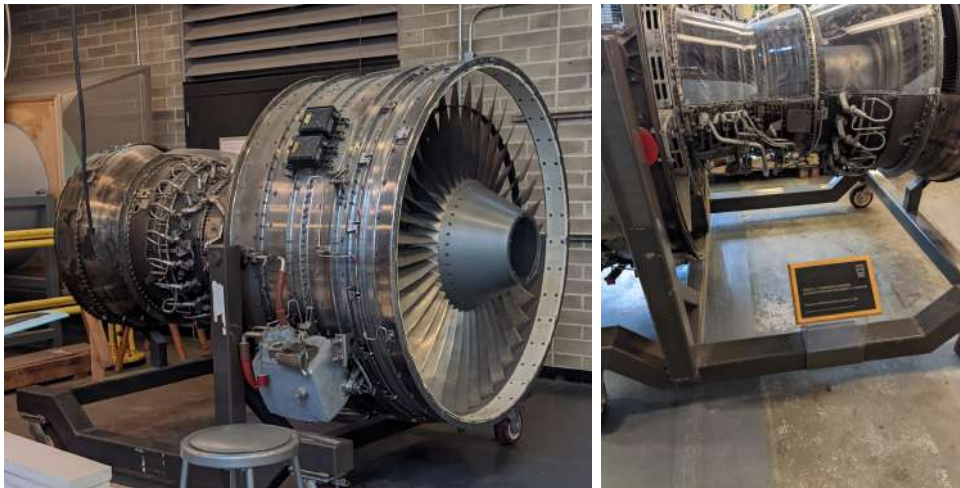
Figure 1.2: CFM56-3 Engine at the MIT Gerard Neumann Hangar.

With several decades of advancements in rotordynamic technology, the CFM56-3 represents a significant evolution in its design. Making use of lighter stronger materials, the rotor is able to operate at higher speeds and under varying thermal loads more reliably. The bearings incorporate advances in lubrication and damping, mitigating vibrations at

criticals and imbalances, reducing wear and tear as well as operational noise [7].

Figure 1.3: Pratt & Whitney PW115 Engine in São José dos Campos, São Paulo, Brazil and SR-71 at the Steven F. Udvar-Hazy Center in Virginia, USA

The Pratt & Whitney 115 redesignated as PW100 and first used by Embraer in the EMB-120 Brasília. [8] The democratisation of the turbojet engine unlocked aviation for developing regions around the world.

The SR-71 used two Pratt & Whitney J58s that has axial-flow turbojets with a static thrust of 32,500 lbf (145kN). [9] At subsonic and transonic speeds, it operates as a turbojet but past mach-2 it operates as a ramjet.

Rocket engines

The first notable usage of turbomachinery in rocket engines was in the V-2 rocket, developed by Germany during World War II and launched in 1944. The shaft spun at 4,000 RPM [10] and used a gas generator cycle due to its simplicity. Mechanical and flow instabilities can cause failures in these systems.

Figure 1.4: V2 rocket in the Imperial War Museum, London, UK and V2 rocket engine section in 33-319 at MIT

The space shuttle main engine had many rotordynamic issues in the form of rapid unscheduled disassemblies (RUDs) as the engineers pushed the limits of turbomachinery. [11] This resulted from operating the high pressure fuel turbopump (HPFTP) too close to its critical modes causing large whirling.

Figure 1.5: Space Shuttle Discovery at the Steven F. Udvar-Hazy Center in Virginia, USA.
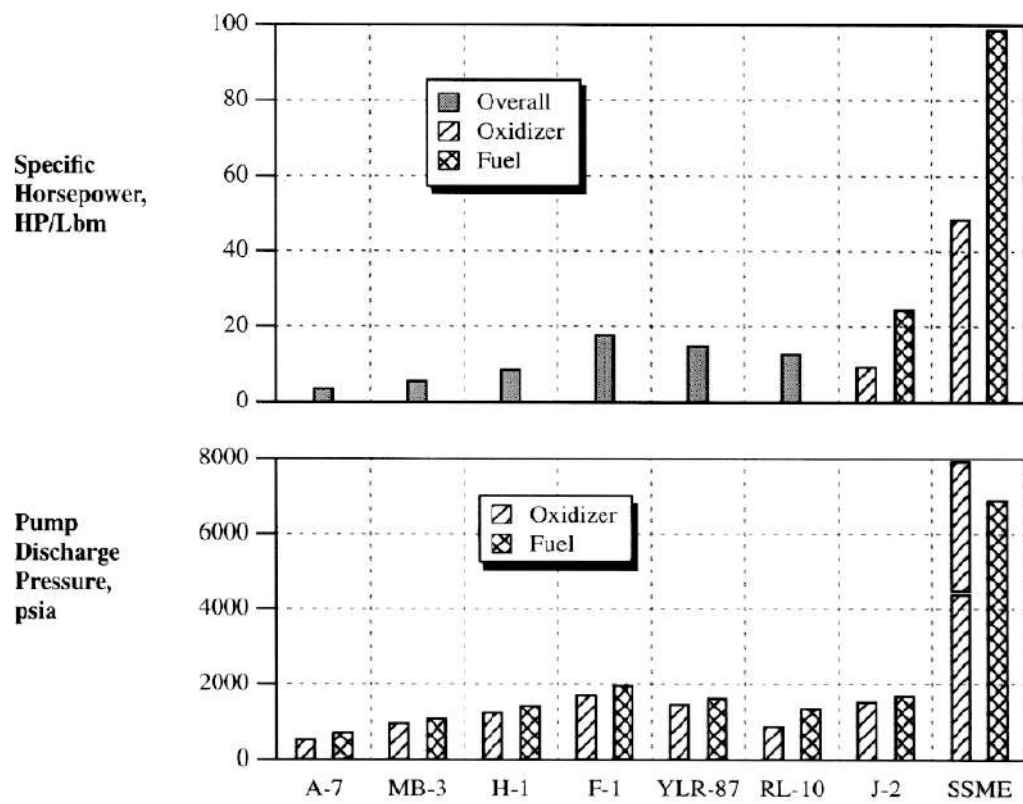
Figure 1.6: Space Shuttle Main Engine Compared to the rest.

The figure about is from lecture notes available on MIT OCW at 16.512 Rocket Propulsion[12].

The Space Shuttle Main Engine (SSME) was first operational in 1981, approximately 17 years after the F-1 and J-2 engines that powered the Saturn V, and the RL-10 that powered the Centaur upper stages [13]. The notable improvements in pump discharge pressure and specific horsepower in the SSME can be attributed to advancements in rotor dynamics. Initially, operating the SSME at full power level (FPL) resulted in failures, which were too close to the critical modes. These issues were mitigated by incorporating squeeze film dampers to modify stiffness and damping, increasing the shaft stiffness, and ensuring proper balancing [14].

Figure 1.7: Falcon 9 at SpaceX, Hawthorne, CA, USA and RD-107 Engine in São José dos Campos, São Paulo, Brazil

Turbopumps are also a main consideration during the design of next generation rockets. Similar to the V-2, the SpaceX Falcon 9 used a gas expansion cycle as it is the simplest to start and most reliable but loses efficiency due to the gas dumped overboard to push the turbine. Going forward, SpaceX's Raptor Engine 2 uses a full flow staged combustion engine [15] that has a dual shaft turbopump setup with fuel rich and oxygen rich environments to drive the movement of fuel and oxidiser into the combustion chamber. Additionally, safety and prioritising reusability in commercial aerospace necessitating fatigue and creep analysis of components through rotordynamics.

By providing tailorable stiffness to these aerospace systems, the critical modes in the rotordynamic response can be engineered for maximum safety and performance during the operation sequence of the engines. With tailorable damping, the vibrations produced by operation near critical modes can be reduced to safe levels.

2

Literature Review

2.1 Turbomachinery

Rotordynamics

Rotordynamics is the study of the dynamic behavior of rotating machinery components, primarily focusing on the vibration characteristics and stability of rotors. Pioneering works by researchers like Kimball [16] and Newkirk [17] laid the foundation for understanding whirl and whip instabilities in rotating shafts. Over the decades, advancements in computational methods and material science have significantly contributed to the field, enhancing predictive models and enabling more complex designs. Jeffcott rotors are the fundamental theoretical model used for rotordynamics [18].
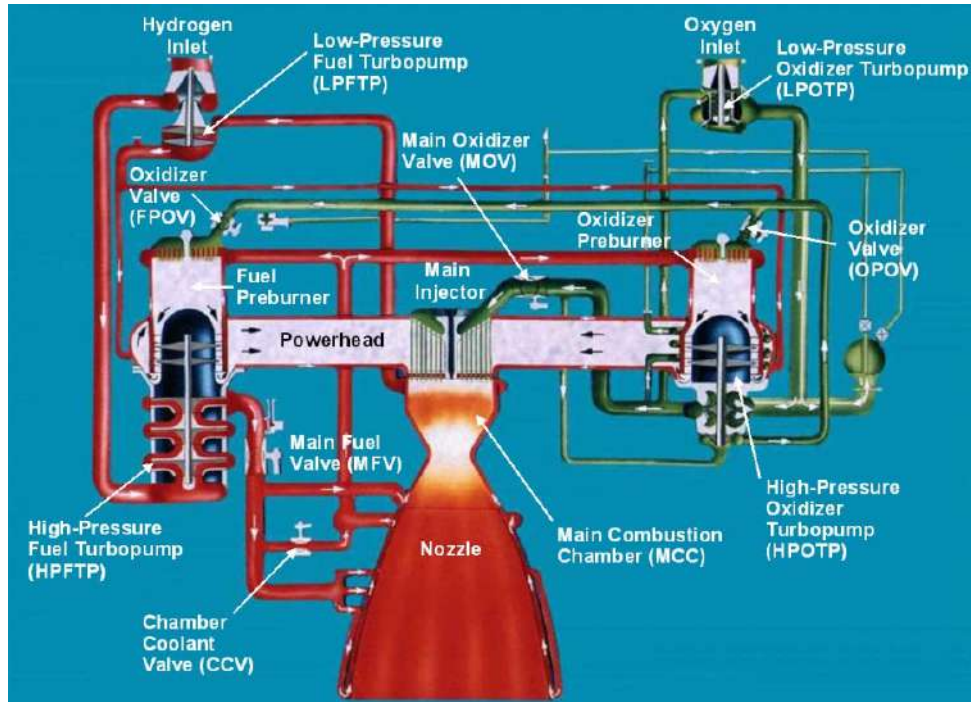
Figure 2.1: Space Shuttle Main Engine Schematic Diagram.

Image from Hooser (NASA MSFC) and Bradley (P&W) in 2011. [11]

Hooser and Bradley list serious turbine blade cracks, subsynchronous vibrations, bearing cage delamination and ball wear on the High Pressure Oxygen Turbopump (HPOTP) as the most salient vulnerabilities.

Whirling phenomena, caused by cross-coupled forces, manifest as precessional motions of a rotor system that can lead to dynamic instabilities and failures, particularly in high-speed turbomachinery like the Space Shuttle Main Engine's (SSME) High Pressure Fuel Turbopump (HPFTP) so it required squeeze film damping. These conditions occur when the rotor speed aligns with the natural frequencies of the oil film or the rotor itself, causing resonance and potentially catastrophic failures. The reusability of the SSME resulted in far longer operating lifetimes at 7.5 hours compared to a few hundred seconds for other turbopumps [19], producing a large amount of flight data that enabled loads and rotordynamic behaviour to be much more well understood, resulting in a predicted reliability of 4 fold by the program's end [11].
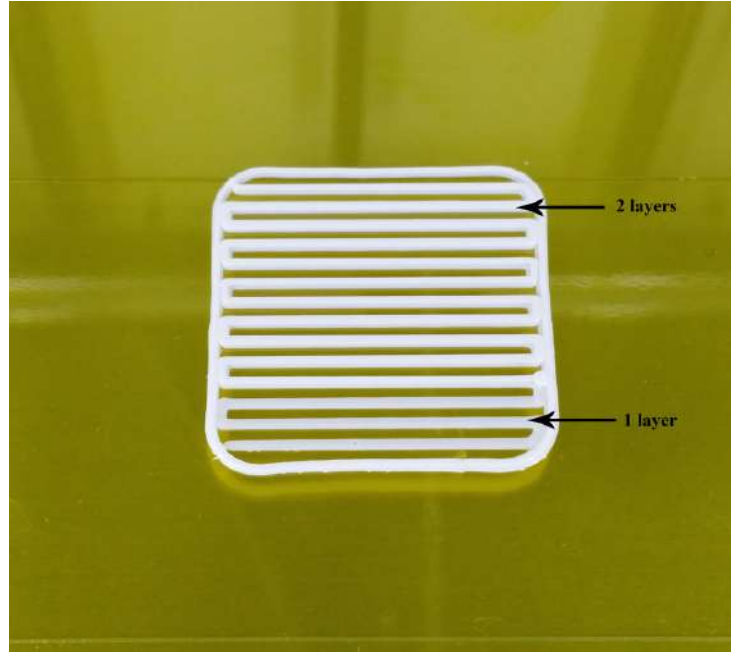
Figure 2.2: Bending dominated structure

Operating Conditions of Bearing Supports in the Space Shuttle Main Engine

Central to its operation were the bearing supports, enabling the engine's rotating components, such as turbopumps, to function under extreme conditions.

The engine's turbopumps and other rotating components could achieve speeds over 30,000 RPM, necessitating bearings and their supports to accommodate these speeds during launch conditions with severe vibration and acoustic stress. By increasing the stiffness in the shaft/bearing system, synchronous vibrations reduced by factors of 2 to 4 [11].

2.2 Metamaterials

From Ashby's Cellular Solids, metamaterials such as foams make good sound absorbers, have good thermal-shock resistance with decreasing density, perfect for damping applications in turbomachinery operating in at extreme temperatures. [20]

Stiffness

The figure above is an example of a bending dominated structure [21]. Compliance in bearing supports are important to controlling the natural frequency of the critical modes and in activating rigid body motion in the shaft movement in a rotordynamic system. Papers from Yu et al (2018) [22], Jiang et al (2022) [23] and Wang et al (2021) [24] all list metamaterial structures that are made up of lattices to maximise compliance. By changing the number of beam layers in a structure, it can be modelled as modifying the stiffness as adding springs in series and parallel.

Damping

Damping in turbopumps is caused by multiple ways such as viscous, structural or hydrodynamic reasons. Fluid-induced forces can couple with the dynamics of the structure resulting in flow induced instabilities (flutter). Squeeze film dampers were used in the SSME to overcome the whirling effects and instabilities. Simulations are commonly used to predict the modes and imbalances to mitigate them, thereby reducing fatigue.

Among these, compliant bearing supports, such as squirrel cages and oil dampers, play a pivotal role in improving machinery stability and performance. Squirrel cages, oil dampers and bearing supports can be directly augmented with metamaterial structures to achieve a more tailored damping response. Dykstra et al. (2023) suggest using buckling to achieve extreme damping while maintaining high stiffness [25] Van Belle et al. (2017) [26] provide methods for studying the dispersion curves of unit cells of metamaterials. However, the simplicity of the method by Garland et al. [27] proposing metamaterial structures with frictional damping designed into the unit cell makes for an easy integration into rotordynamic bearing supports.

Interpenetrating Lattices

The usage of Interpenetrating lattices (IPL) in metallic form has been unlocked by additive manufacturing techniques such as laser powder bed fusion (LPBF) and lattice and

topology optimisation design software such as Ntop. [28] Ntop makes use of Triply Periodic Minimal Surfaces (TPMS), field driven design and generative design for integration of new metamaterial structures with conventional components.

White et al. (2021) showed that metal additively manufactured (AM) IPLs gain improved mechanical performance through increased toughness and tailorable mechanical energy absorption [29]. Taylor et al. from MIT (2021) [30] demonstrated that plastic AM IPLs showed 'increased strain-to-failure and tailorable specific energy absorption'.

The damping properties of additively manufactured lattices are governed by several factors such as strut-based friction, unit cell size, material intrinsic damping, lubrication film damping and interface damping. By deforming the lattice as to cause the struts to touch, damping via strut-based friction is activated as rubbing between the surfaces results in energy dissipation [31]. The same occurs betweeen the lattices and neigbouring components, generally controlled by tightness of fit and surface finish.

2.3 Additive Manufacturing

Additive manufacturing (AM), also known as 3D printing, has revolutionised manufacturing by enabling the creation of complex geometries with internal features impossible with conventional techniques. Additive manufacturing is currently used in rocketry to print components of complex shapes that such as cooling channels within rocket engines. NASA's Gradl and Protz detail these technology advancements [32]. Laser powder bed fusion (LPBF), a prominent AM process, utilises a laser to selectively melt and fuse metallic powder particles layer-by-layer to create a three-dimensional object. Other processes such as Direct Energy Deposition (DED) use plasma arcs to melt powder that deposited by a nozzle. DED is much faster but makes for crude finishes and does not have resolution required for lattices and complex geometries.

Laser Powder Bed Fusion (LPBF)

LPBF unlocks design freedom that allows for tailorable stiffness, strength-to-weight ratios and damping that are unachievable in conventional structures. Versatility with various metallic materials enables the tailored alloys selection that can account for properties suited to aerospace applications[32]. Further still, multi-material prints allow for functionally graded components that can leverage untapped regions of Ashby Plots. This is all made possible with the advent of new softwares such as NTop that allow the design of complex structures with integrated metamaterials as a focus. Other existing softwares such as solidworks and fusion 360 have also introduced topology optimisation features [33] [34].

Controlling the surface roughness of components produced by LPBF can be a challenge. Cao et al demonstrate that scanning speed has positive impacts whereas the layer thickness has a negative influence on the height and go on to suggest optimal parameters for 316L Stainless Steel [35]. Typically, to counter any potential issues such as fatigue crack initiation or distortion caused by porosity, residual stresses and any other defects, techniques such as hot-isostatic pressing (HIP) is common in aerospace centers such as the NASA Jet Propulsion Laboratory as a standard to increase density and for qualification in spacecraft [36]. Cost is a main consideration for the usage of LPBF due to the large startup cost for equipment and cost of materials and supplies for the printer. A standard EOS M100 costs between $100,000 and $250,000 due to varying configurations. [37] Ultimately, if there are components providing functionality important enough for the costs to be borne. LPBF is an excellent manufacturing technique. A section in the appendix details the cost of LPBF bearings.

2.4 Characterisation Methods

Laser-Induced Resonant Acoustic Spectroscopy (LIRAS)

Laser-Induced Resonant Acoustic Spectroscopy (LIRAS) represents a non-contact, non-destructive method for measuring the mechanical properties of materials. The technique

involves the use of laser pulses to generate acoustic waves within a material. The resonant frequencies of these waves are then analyzed to determine the material's stiffness and damping characteristics. A recent paper by Yun et al. [38] explores the dynamical response of lattice structures fabricated out of IP-Dip photoresist provides strong guidance on the LIRAS method and calculations of modulus, speed of sound, dispersion relation and damping. LIRAS offers several advantages over traditional contact-based measurement techniques such as non-destructive measurement, high spatial resolution and rapid measurement.

Traditional Contact-Based Methods

Traditional methods, such as dynamic mechanical analysis (DMA) and modal analysis, require physical contact with the specimen. These methods can provide accurate measurements but may not be suitable for fragile or small-scale specimens due to the risk of damage or alteration of material properties [39].

Other Non-Contact Methods

Other non-contact methods, including ultrasonic testing and digital image correlation (DIC) and Laser Doppler Vibrometry (LDV), also offer advantages in measuring material properties without physical contact [40]. LDV differs to LIRAS in that it measures the movement of the surface while LIRAS induces internal vibrations as such is more tailored for measuring the dynamical response of a surface while LIRAS is tuned to internal properties such as stiffness by attaining an acoustic response. DIC may have limitations in spatial resolution compared to LIRAS.

2.5 Conclusion

It is clear from this literature review that the rotordynamics of aerospace systems can be tuned by changing the stiffness and damping of bearing support structures and that integration with metamaterials can provide tailorable properties. By using a rotordy-

namic rig, the dynamical response of these systems can be studied and models verified. Employing LIRAS presents a promising technique for the non-contact, non-destructive characterization of damping and stiffness in metamaterials of interest.

3

Aims

3.1   Aims

This research aims to design bearing supports that produce tailorable stiffness and damping and relate these properties to the structure of additively manufactured lattices. The objectives of this research are outlined as follows:

1. Assess the rotordynamic behaviour of bearing supports of various materials in relation to additively manufactured components

2. Relate stiffness and damping to the geometry (relative density, unit cell size, etc.) of additively manufactured metal interpenetrating lattices (IPLs).

3. Develop bounds on stiffness and damping of AM IPLs.

4. Integrate AM IPLs with bearing support structures.

1) will be achieved by using a Jeffcott rotor to study the dynamical response, identify critical modes, eccentricity, damping and the presence of whirling motion. 2) and 3) Will be done by analysing the IPL data from the LIRAS setup at the Institut de Physique de Rennes Université de Rennes, France, verifying the simulation models and using them to predict bounds. 4) will be done by integrating solidworks designs with IPLs in Ntop.

<div style="text-align: right;">

4

</div>

Materials and Methods

4.1 Rotordynamics

Experimental Setup

The experiment's goal was to investigate the rotordynamic behaviour of a Jeffcott rotor system by measuring the displacement and rotational frequency under various conditions. The setup involved synchronising x and y axis proximity sensors to monitor the displacement, while a tachometer was employed to measure the rotor's frequency of rotation. Bearing supports were interchangeably used to examine the impact of different materials on the system's dynamics. A 24V power supply was used to power the setup.
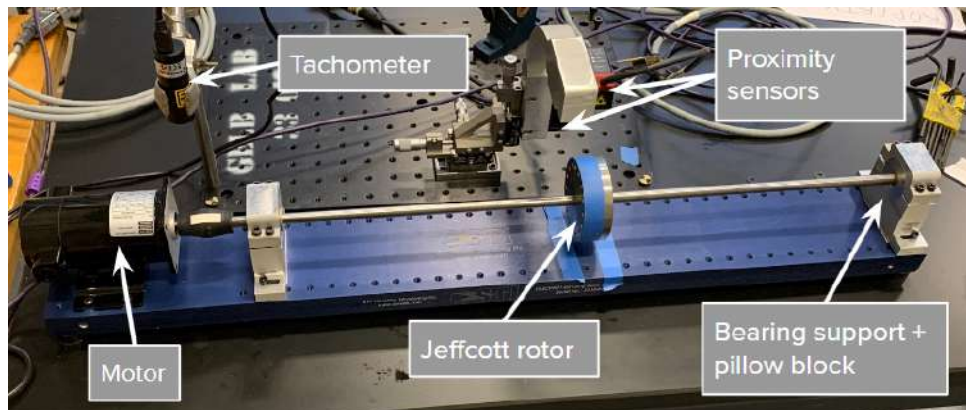
Figure 4.1: Jeffcott RRotordynamic Rig.
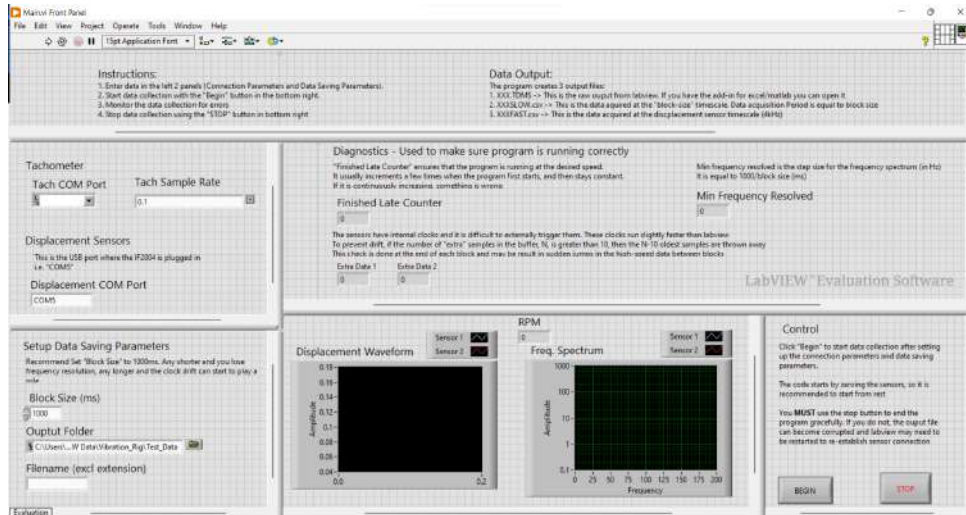
Figure 4.2: Labview Script.

Sensor Synchronisation and Data Acquisition

A LabVIEW script as in was developed by Colin Pavan in the group to synchronise the proximity sensors with the Jeffcott rotor, ensuring accurate measurement of displacements in the x and y axes simultaneously. This script facilitated real-time data acquisition and monitoring, allowing for the precise control and observation of the rotor's dynamics. It also converted the time series data from the tachometer into the frequency spectrum via the use of Fast Fourier Transforms (FFT).

Preparation of Bearing Supports

The experiment required the use of bearing supports made from various materials to study their influence on the rotor system. The supports were designed to be easily interchangeable by attaching or removing screws. This design enabled quick transitions between tests with minimal downtime, ensuring a wide range of material impacts could be assessed efficiently.

Modification for Optical Measurements

To enhance the accuracy of displacement measurements using laser-based proximity sensors, blue matte tape was applied to the rotor's surface. This modification prevented the

laser light from scattering on the aluminium's lustrous metal surface, ensuring that the reflected laser light could be accurately detected by the sensors.

Operational Procedure

The rotor was activated and gradually accelerated from 0 to 8000 RPM. Throughout this process, the proximity sensors measured the displacement, while the tachometer recorded the rotational frequency. These measurements were continuously monitored to identify any instances of instability or significant changes in the rotor's behaviour.

4.2   Additive Manufacturing

Pre-Processing

We utilised computer-aided design (CAD) software such as solidworks and Ntoplogy to create 3D models of the component. The CAD model is converted into a Standard Tessellation Language (STL) file, reducing the model to a mesh of triangles. The STL file is processed using the Magics software by Materialise, generating supports and EOS print is used to interface with the M100 printer, adjusting for any necessary geometric compensations. Overhangs and supports are all considered at this stage for printing laser powder bed fusion (LPBF).

Parameter Setting:

In the EOS software, select the printing parameters:

Powder Loading

We chose 316 Stainless Steel powder from the EOS M100's as opposed to inconel, titanium due to the low cost for prototyping. We carefully loaded the selected metal powder into the printer's supply chamber, adhering to strict safety protocols written up by EOS and the group to prevent exposure to potentially hazardous metal powders.

Table 4.1: Laser and material parameters

Parameter	Value
Infill laser power	77.1 W
Infill scan speed	827 m/s
Powder size	20–65 μm
Hatch spacing	70 μm
Layer thickness	20 μm
Contour laser power	67.2 W
Contour scan speed	600 m/s
Laser beam diameter (spot size)	40 μm
Inert atmosphere	Ar with 0.12% O_2
Recoater type	HSS recoater
Laser type	~1040 nm Yb fiber laser (max beam power: 200 W)

Printing Process and post processing

The EOS M100 spreads a precise layer of metal powder over the build platform, with a laser selectively melting and fusing the powder to form the part, layer by layer.
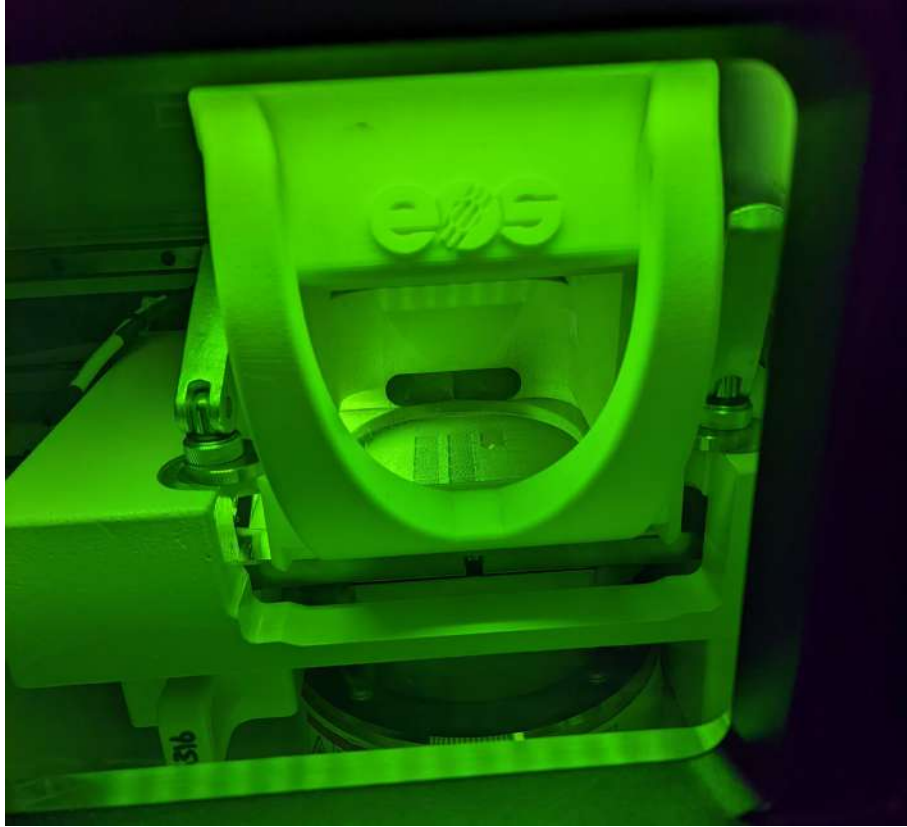
Figure 4.3: EOS M100 during printing 3 rows of lattices.

First, allow the build chamber and components to cool adequately before removal to prevent thermal distortion. The printed components are typically cut from the build plate using wire EDM in the group but I used a hand saw or bandsaw as the large area to cut for my prints would have required 36 hours otherwise. Visual inspection, dimensional accuracy verification with vernier callipers, and mechanical property testing are carried out to ensure the components are printed as expected.

Safety and Maintenance

Adherence to all safety guidelines provided by EOS for the M100 printer and produced by the MIT Aerospace Structures and Materials Laboratory, including the use of protective equipment and maintaining proper ventilation in the printing area.

4.3 Compression Test

The compression properties of the material were characterised using an Instron testing machine. We limited the strain to 0.75 and measured the stresses.
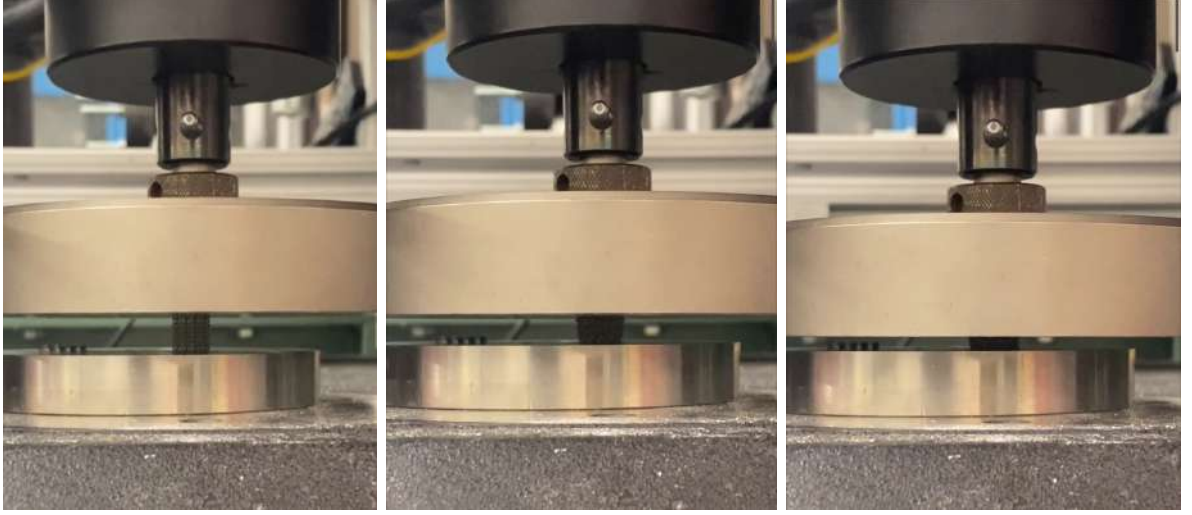
Figure 4.4: Instron test at different stages of compression ending at 0.75 strain.

Testing Machine Setup

The Instron testing machine was calibrated before conducting the compression tests to ensure accuracy in force application and measurement. The machine's grips were adjusted to securely hold the specimens without inducing pre-stress or altering the material properties.

Test Execution

Each specimen was loaded into the Instron machine's grips and aligned to ensure uniaxial tension without bending or twisting forces. The machine was then set to apply a compressive load at a constant strain rate until the specimen reached a strain of 0.75 or failure. The force applied and the corresponding elongation of the specimen were continuously recorded throughout the test.

Safety

During operation, we wore goggles and stood away from the machine. The compression plate was checked to be tight and the arm was fastened with a pin to prevent lateral motion.

Data Acquisition and Analysis

The Instron testing machine provided a detailed record of the force versus displacement data for each specimen. This data was used to calculate the stress-strain relationship, from which the material's compressive strength, yield strength, modulus of elasticity, and elongation at break were derived. The peak stress experienced by the specimens was noted to be just below 14 MPa, indicating the material's maximum compressive capacity.

4.4 Laser-Induced Resonant Acoustic Spectroscopy (LIRAS)

The methodology employed for the non-destructive testing of interpenetrating lattices focused on evaluating their stiffness and damping properties utilised Laser-Induced Resonant Acoustic Spectroscopy (LIRAS). Conducted in collaboration with our French partners, Thomas Pezeril and Ievgeniia Chaban at the Institut de Physique de Rennes Université de Rennes, this experimental approach allowed for the precise characterisation of the mechanical properties of both lattice-structured materials and a chromium sample used as a control for comparative analysis.

Experimental Setup

Sample preparation was requires as the printed lattices with bottom or top plates required grinding and polishing to produce the mirror surface required for LIRAS. The grinding and polishing machine and pads from 150 grit to 1 micrometer diamond suspension were used.
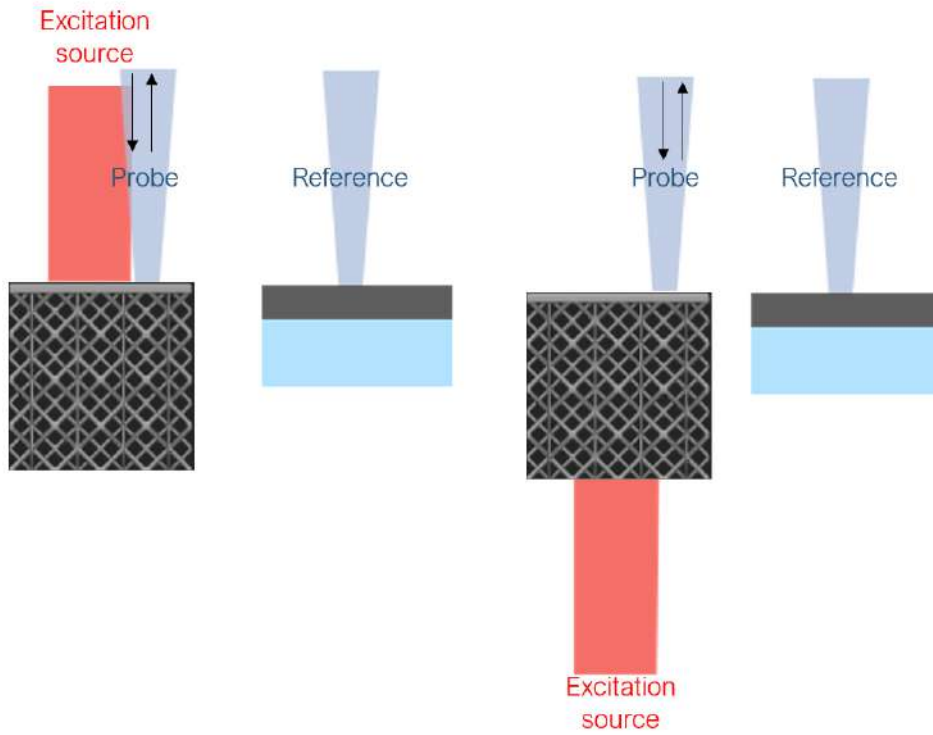
Figure 4.5: LIRAS Setup with the lattice excited from the Front and the Back against a reference.
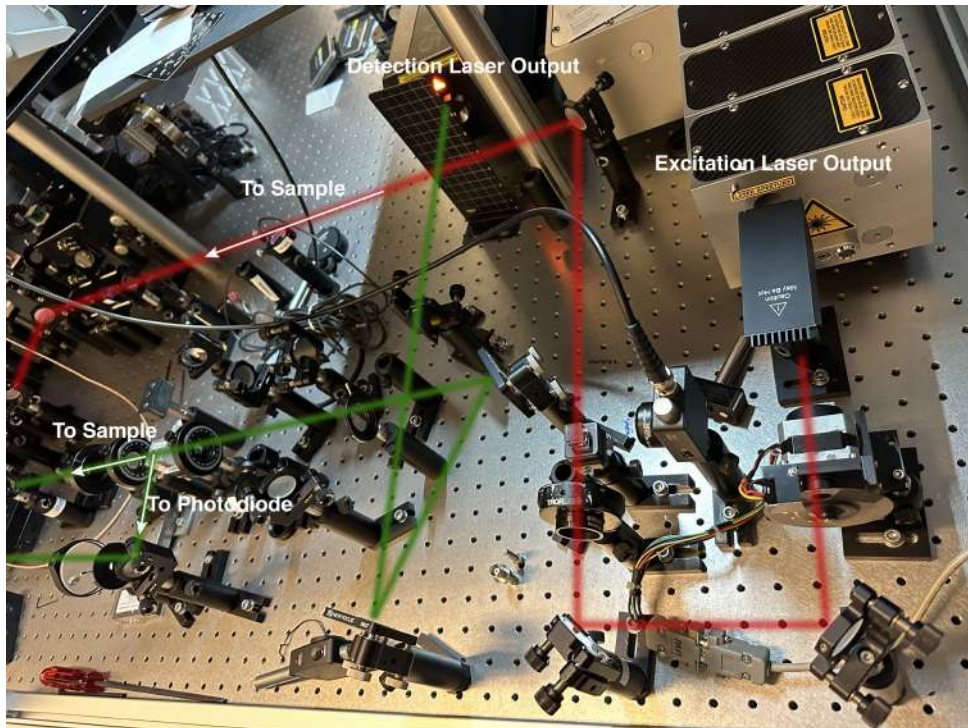
Figure 4.6: LIRAS Optics Setup showing the path of the excitation and detection lasers guided to samples.
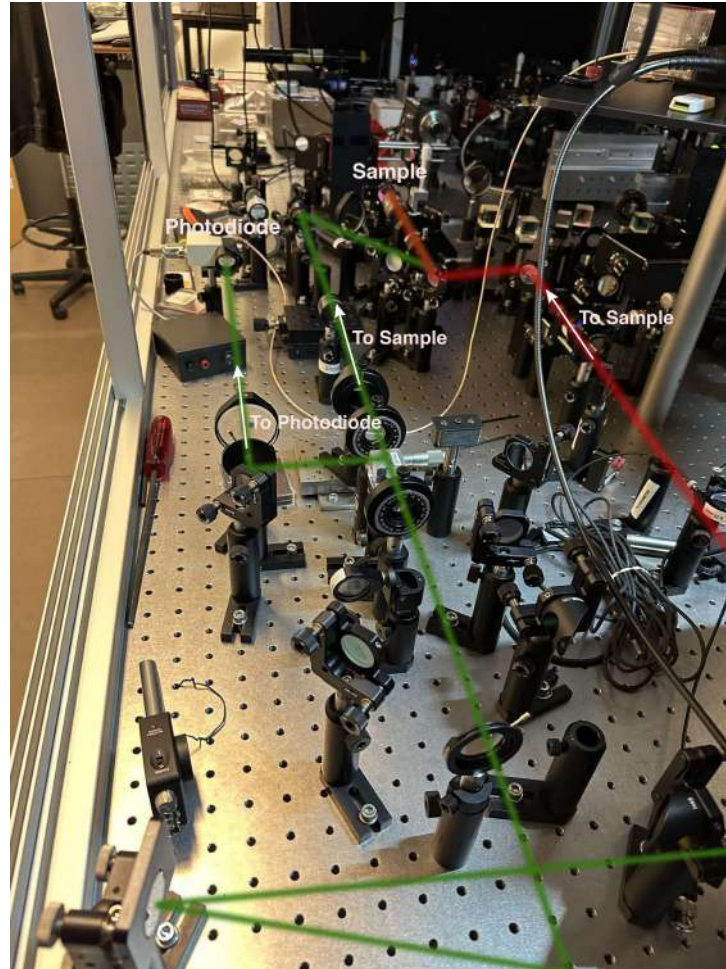
Figure 4.7: LIRAS Optics Setup showing the path of the excitation and detection lasers guided to samples.

The LIRAS setup involved a pulsed laser system to induce acoustic vibrations within the samples. As observed from the figures above, the red lines show the path of the excitation laser getting guided to the sample and the green lines show the path and splitting of detection laser to the sample and to the avalanche photodiode detector. The laser parameters were carefully calibrated to optimise the energy transfer and ensure the generation of clear, measurable acoustic signals without damaging the samples. We used a picosecond pulsed laser to excite the specimens, which was set to emit pulses with a duration of 600 picoseconds at a wavelength of 1064nm and a pulse energy of 6mJ. The pulse energy was variably adjusted, ranging from 20 to 50 µJ, based on specific experimental needs. Additionally, a continuous-wave (CW) detection laser, operating at a wavelength of 532 nm and capable of outputting up to 10 mW, was employed for monitoring the vibrations induced in the specimens. To mitigate the risk of sample damage from excessive laser exposure, the output power of this CW laser was carefully maintained at around 2.5 mW throughout the experiments.

5

Rotordynamics

5.1 Theory

At its most basic level, rotordynamics is concerned with one or more mechanical structures (rotors) supported by bearings and influenced by internal phenomena that rotate around a single axis. This work is mostly concerned with vibrations in rotordynamic systems.

Mass Spring Damper Model: The equation models a damped harmonic oscillator which is used to analyze the dynamic behavior of a mass-spring-damper system. Here, $y(t)$ the displacement of the mass m, $\dot{y}(t)$ the velocity, and $\ddot{y}(t)$ represents the acceleration. The terms involving ζ and ω_0 represent the damping ratio and the natural frequency, respectively, while $F(t)$ is an external force applied to the system.

$$\ddot{y}(t) + 2\zeta\omega_0\dot{y}(t) + \omega_0^2 y(t) = \frac{F(t)}{m} \tag{5.1}$$

Natural or free damped motion of the rotating shaft: These equations extends the model for a natural or free undamped motion of a rotating shaft to include damping. Ω is the angular velocity and r the radial displacement. The system is in equilibrium when the centrifugal force $-m\Omega^2 r$ is balanced by the restoring force kr. The natural frequency of the system Ω_n is defined as $\left(\frac{k}{m}\right)^{\frac{1}{2}}$. The damping force is represented by $c\Omega r$ and $2m\Omega\frac{dr}{dt}$, while ζ is the damping ratio. The exponential term $e^{-\zeta\Omega_n t}$ describes how the amplitude

of the motion decays over time due to damping.

$$-m\Omega^2 r + m\frac{d^2x}{dt^2} + kr = 0 \tag{5.2}$$

$$2m\Omega\frac{dr}{dt} + c\Omega r = 0 \tag{5.3}$$

$$\frac{r}{r_0} = e^{-\zeta\Omega_n t} \tag{5.4}$$

$$\Omega_n = \left(\frac{k}{m}\right)^{\frac{1}{2}} \tag{5.5}$$

$$\zeta = \frac{c}{c_{crit}} = \frac{c}{2m\Omega_n} \tag{5.6}$$

Jeffcott Rotor Equations

The Jeffcott rotor model is a fundamental mechanical system used to analyze the dynamic behavior of rotating machinery.

Mass and Inertia

$$M_{\text{shaft}} = \pi r_{\text{shaft}}^2 L\rho \tag{5.7}$$

$$I_{z,\text{disc}}^A = \frac{\pi(r_{\text{disc}}^4 - r_{\text{shaft}}^4)}{4} \tag{5.8}$$

$$I_{z,\text{shaft}}^A = \frac{\pi r_{\text{shaft}}^4}{4} \tag{5.9}$$

The equations above define the mass and the axial moments of inertia for both the disc and the shaft of the rotor. M_{shaft} represents the mass of the shaft calculated by its volume (area of the cross-section times the length L) and density ρ. $I_{z,\text{disc}}^A$ and $I_{z,\text{shaft}}^A$ represent the axial moments of inertia for the disc and shaft, respectively.

Stiffness and Frequency Calculations

$$K_r = \frac{2 \times 48EI^A_{z,\text{shaft}}}{L^3} \tag{5.10}$$

$$\omega_n = \sqrt{\frac{K_r}{M_{\text{disc}}}} \tag{5.11}$$

$$\omega_c = \sqrt{\frac{K_s}{M_{\text{disc}}}} \tag{5.12}$$

$$\text{RPM}_n = \frac{\omega_n \times 60}{2\pi} \tag{5.13}$$

$$\text{RPM}_c = \frac{\omega_c \times 60}{2\pi} \tag{5.14}$$

This section discusses the stiffness and natural frequencies of the system. K_r is the radial stiffness of the shaft, calculated using the modulus of elasticity E, and ω_n and ω_c are the natural and critical angular frequencies, respectively. The rotational speeds RPM_n and RPM_c are derived from these frequencies.

Dynamic Response and Transmissibility

$$\zeta = \frac{B_r}{2M_{\text{disc}}\omega_n} \tag{5.15}$$

$$\kappa = \frac{K_s}{K_r} \tag{5.16}$$

Damping and stiffness ratios are crucial in understanding the dynamic behavior. ζ is the shaft damping ratio, which is determined by the support damping coefficient B_r which we set to zero here, and κ is the ratio of stiffnesses between the spring and the shaft.

$$r_{b/e} = \frac{\left(\frac{\Omega}{\omega_n}\right)^2}{\sqrt{\left(1 - \left(\frac{\Omega}{\omega_n}\right)^2\right)^2 + (4\zeta\left(\frac{\Omega}{\omega_n}\right)^2)^2}} \tag{5.17}$$

The equation above calculates the ratio of transmissibility at various angular velocities Ω. By substituting constants $c_1, c_2,$ and c_3, we define the dynamic response as follows:

$$\nu = \frac{B_s}{2M_{\text{disc}}\omega_c} \tag{5.18}$$

$$c_1 = 1 - (1 + \kappa)\left(\frac{\Omega}{\omega_c}\right)^2 \tag{5.19}$$

$$c_2 = 1 - \kappa\left(\frac{\Omega}{\omega_c}\right)^2 \tag{5.20}$$

$$c_3 = 2\frac{\Omega}{\omega_c}\nu \tag{5.21}$$

$$r_{a/e} = \frac{\left(\frac{\Omega}{\omega_c}\right)^2}{\sqrt{c_1^2 + (c_3 c_2)^2}} \tag{5.22}$$

$$r_{b/e} = r_{a/e}\sqrt{(1 + \kappa)^2 + (c_3\kappa)^2} \tag{5.23}$$

The constants c_1, c_2, and c_3 are critical for the dynamic response, affecting the amplitude ratios $r_{a/e}$ and $r_{b/e}$ under different operational conditions. ν is the support damping ratio.

The Fast Fourier Transform (FFT): is utilised to efficiently compute the Discrete Fourier Transform (DFT) of the time series rotordynamic amplitude data for analysing the frequency components of a signal. The DFT is represented by the following equation:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-i2\pi kn/N}$$

where $X(k)$ is the value of the frequency domain at frequency bin k, $x(n)$ is the input signal at time n, N is the number of samples.

The Lorentzian curve: This can be fit to peaks to gain information about damping:

$$L(x; A, x_0, \gamma) = \frac{A}{\pi}\left(\frac{\gamma}{(x - x_0)^2 + \gamma^2}\right) \tag{5.24}$$

where A is the amplitude, x_0 is the center frequency of the peak, and γ is the half-width at half-maximum (HWHM).

$$\zeta = \frac{\gamma}{\omega_0} \tag{5.25}$$

ζ is the damping coefficient and ω_0 is the natural frequency. The code for the rotordynamic analysis is contained in the appendix.

5.2 Dyrobes Simulation Software

The Dyrobes software is used carry out rotordynamic analysis, vibration analysis, bearing performance and balancing calculations based in Finite Element Analysis. We have used it to predict instabilities.

Shaft Specifications	
Shaft Length (aluminum bearings)	554 mm
Shaft Length (plastic bearings)	558 mm
Shaft Diameter	9.55 mm
Material	Steel

Table 5.1: Shaft Specifications

Disk Specifications	
Disk Length	25.44 mm
Disk Diameter	76.02 mm
Disk Mass	0.832 kg
Material	Steel

Table 5.2: Disk Specifications

Compliance requirements for bearing housing supports.
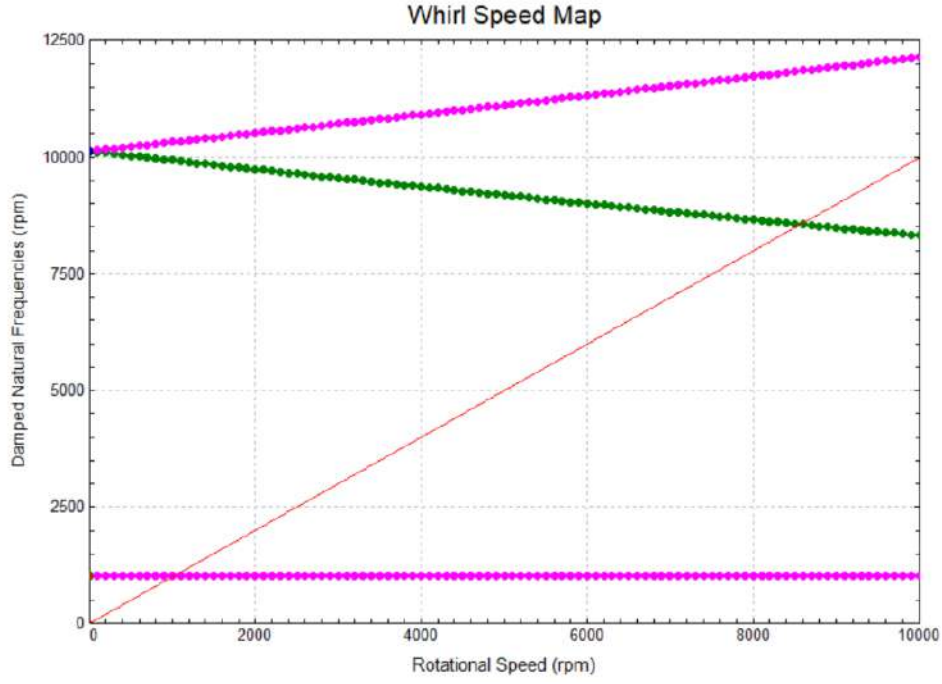
Figure 5.1: Whirl Speed Map.

This figure is a Cambell diagram which shows the critical modes against the rotational speed. First mode expected at 1035 RPM. Designing the startup sequence of a turbomachine is crucial in ensuring safety during operation. Staged combustion cycles are very difficult to start and require alot of testing compared to relatively simpler systems such as a gas expander cycle. As mentioned in the literature review, the engineers of the space shuttle main engine (SSME) were careful to avoid critical modes.
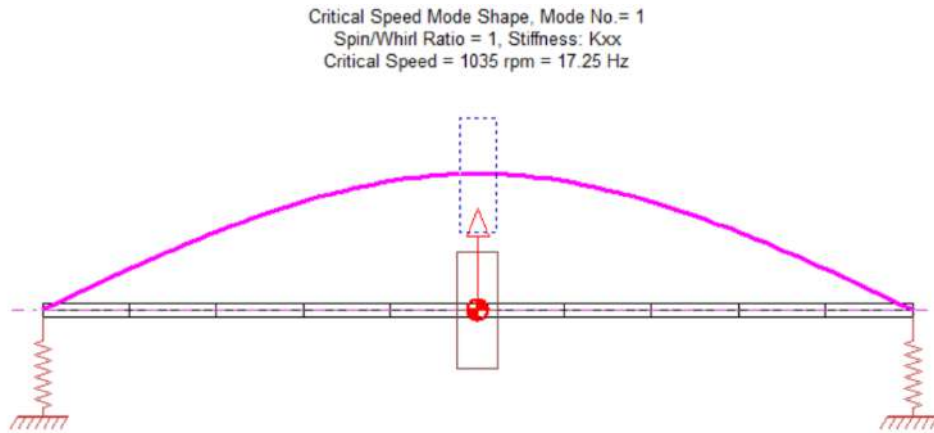
Figure 5.2: Bending Mode.

As observed, the first bending mode at 1035 RPM is very simple and bending from the

center of the beam. Large amplitudes in real life rotordynamic systems can cause failures as turbine blades or rotors rub with the housing.
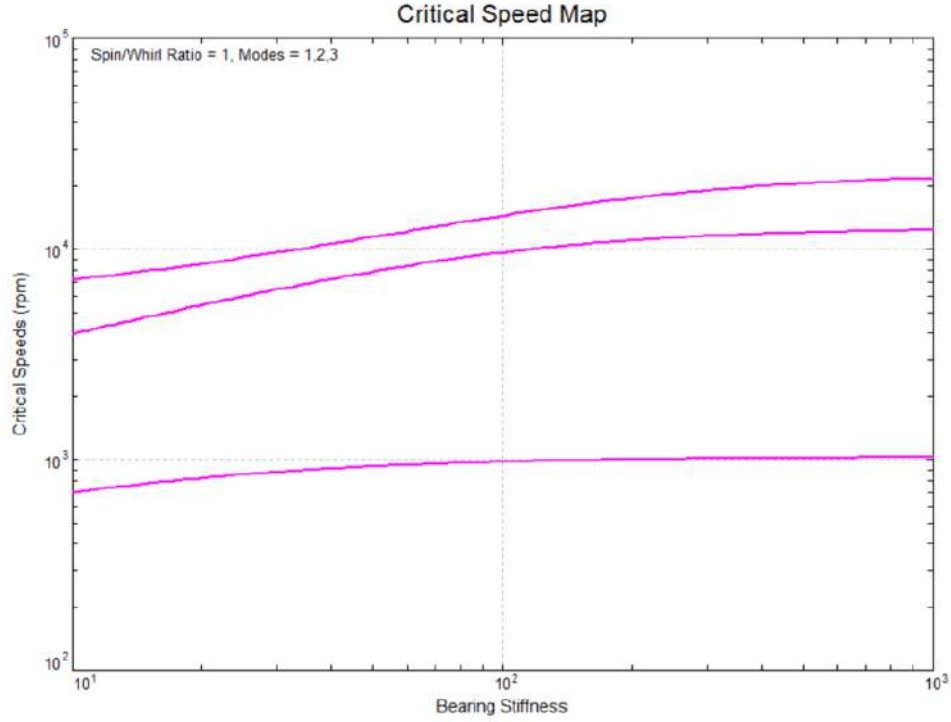
Figure 5.3: Critical Speed Map against Bearing Stiffness.

By changing the bearing stiffness, the critical speeds can also change. Rigid body modes can also be activated where no internal deformation of the shaft occurs, instead the whole system engages in transnational movement. However, as evidenced in the plot, the bearing stiffness must be <300 N/mm to have a significant decrease in the critical speed. However, the shaft is still in the flexural modes at low bearing stiffness. To measure how much force the bearing support undergoes, a laser probe or accelerometer should be used. We did not do this in our experiments.
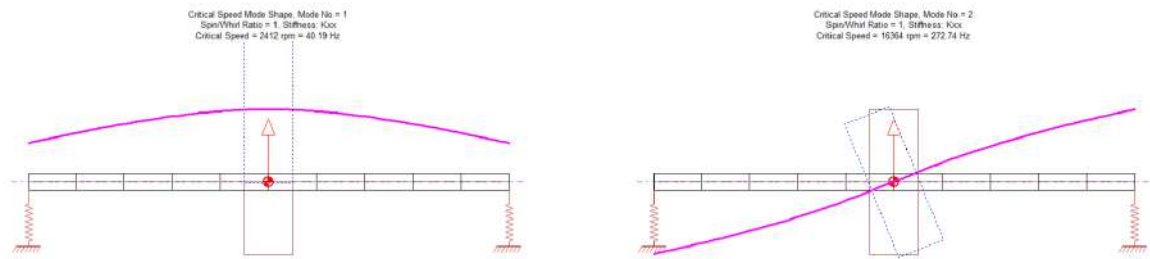
Figure 5.4: a) First and b) Second rigid body modes generated by Dyrobes.
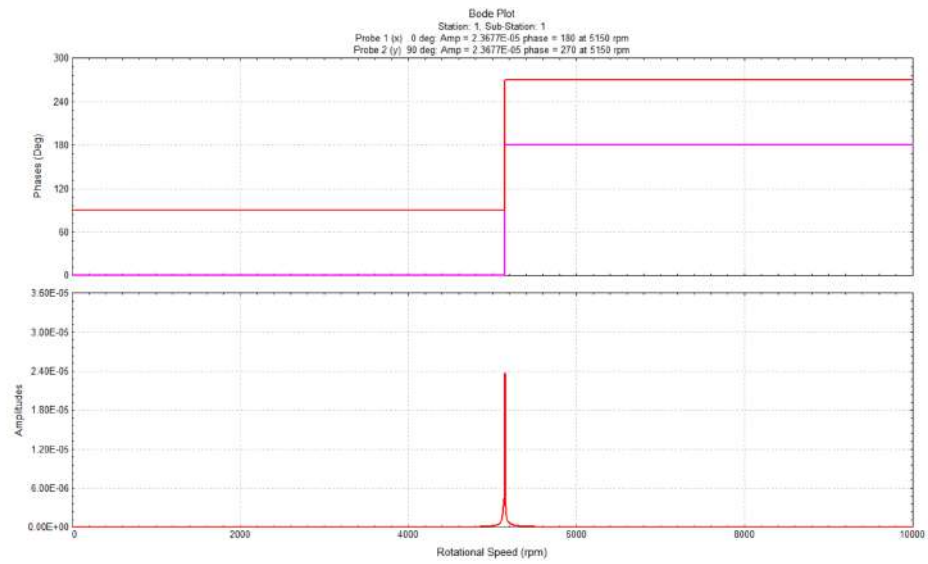
Figure 5.5: Bode Plot.

Dyrobes is also able to generate bode plots that allow you to measure damping from the sharpness of the peaks. The figure above is an example where there is a peak at 5000 RPM. Lorentzian curves can be fit to study the damping through the half-width at half maximum.

5.3 Measurements
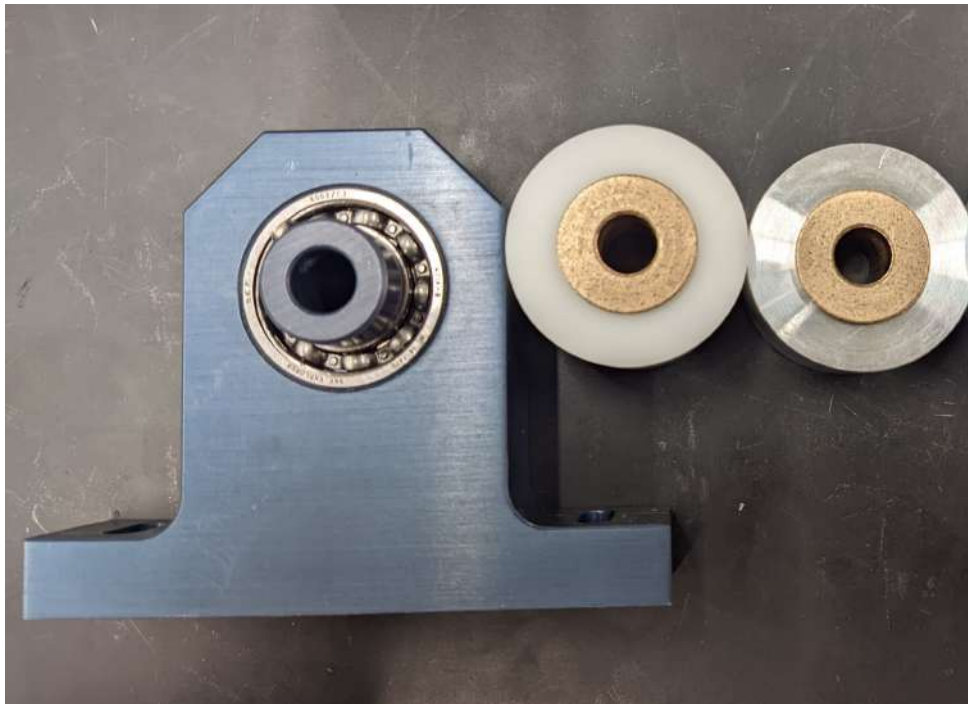
Figure 5.6: Bearing supports of different materials.

Initially during the experiments, the pillow blocks needed to be loosened to align the shaft with the bearing. Otherwise, unnecessary stress is experienced by the shaft and sounds that are described later are heard. An important note is that, the small fluctuations in the rotordynamic data caused large measurements of displacement measured. As such, these large values have to be removed before plotting the data as python understands these values as infinities.
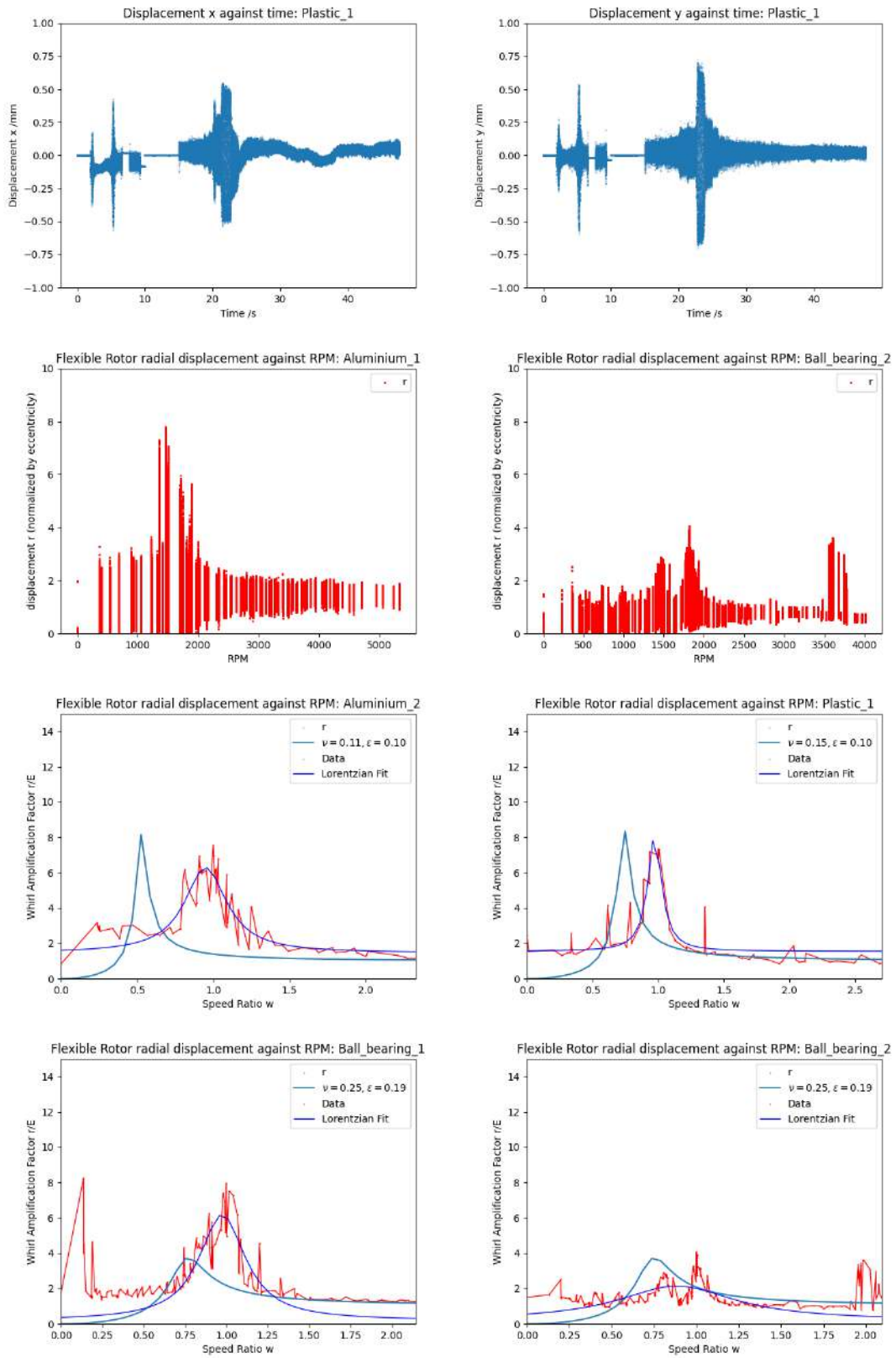
Figure 5.7: Displacement plots for Plastic, Aluminium, and Ball Bearing

36

The figure shows plots for aluminium, plastic and the ball bearing normalised and not normalised by critical RPM. The first two plots show the x and y displacements against time which can be studied to understand the forwards and backwards whirling. Two plots show the scatter of the displacement data in x and y combined for r. The vertical lines observed illustrate the discrete nature of the sweeping which was paused for about a second at a time.

By fitting an analytical flexible jeffcott rotor system to the different response profiles, the eccentricity and the damping ratio can be determined in addition to the damping coefficient from the lorentzian fit. As observed, the peak for the analytical plastic plot although lower RPM than measured has already been modified as the bearing support was behaving as a composite material due to the brass oil-lite fitting and the aluminium pillow block. Initially, by setting the modulus of the material to that of the bearing support, the frequency of the peaks observed from the analytical plots were well below that of the frequency of the critical modes observed. It seemed to behave as if the bearing support was much stiffer. Postulating that the aluminium of the pillow blocks and the brass of the oil-lite bearings also contributed to the stiffness, I used an approach similar to rules of mixture with the different thicknesses of the materials as volume fractions, and achieved predictions much closer to the criticalz modes observed.

$$E_{overall} = \frac{\Sigma_m E_m h_m}{h_{overall}} \tag{5.26}$$

Table 5.3: Material Properties and Bearing Support Data

Property	Aluminium	Plastic	ball Bearing
E (GPa)	94.04	94.04	220
ν	0.1	0.15	0.12
ϵ	0.1	0.1	0.19
Critical RPM	1462	1406	1773
W_c in RPM	1041	1396	1592
W_n in RPM	1517	1517	2320
B_s	18.13	36.49	33.28

Where: E represents Young's Modulus; ϵ denotes Eccentricity; If W_c and W_n are the

critical frequencies and the natural frequencies respectively.

As the model fits the experimental data quite well. We can conclude that using rotordynamics is a reasonable method of measuring the stiffness and the damping of the bearing supports.

From the final plot in the figure 5.8 that is for the ball bearing, it is observed that at twice the the first critical at the speed ratio of 2, there is another peak. This corresponds to a 2nd superharmonic response. Being able to predict and measure these are important in turbomachinery applications [41]. The damping ratio was also larger for this run of the ball bearing data and is likely due to the pillow blocks not being aligned with the shaft fully, resulting in more friction from the rotation.

5.4 Metamaterial Design

The following section details the design of the AM metamaterial bearing support designed in solidworks and tested on the rig.

Figure 5.8: Iterations on bearing support design.

With compliance and damping mechanisms. The first iteration had a large number of springs with small spacing, small number of long fins. There was concern about fins bottoming out or spring beams getting stuck. The second iteration involved large spring

beam spacing, a large number of shorter/thinner fins, thinner outer and inner rings as a result.
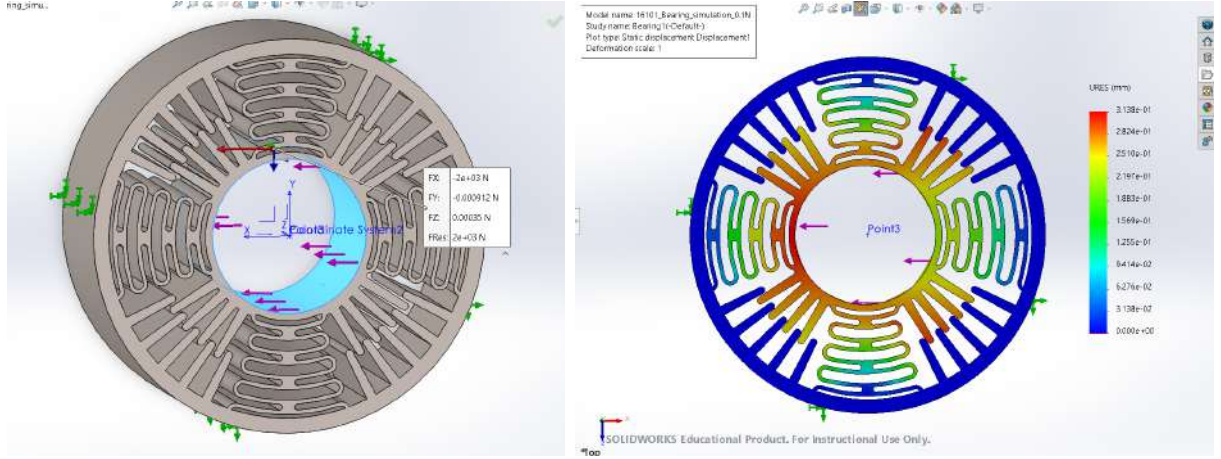
Figure 5.9: FEA of Bearing Support Contours of 100N showing the displacement.

Already from the FEA modelling, it seemed unlikely that the supports would be compliant enough to produce rigid body modes as it required loads of over 100N for any meaningful deformation that would activate the frictional forces which is achievable in real turbomachinery conditions but unlikely in our laboratory setup. However, the print was done so that we could test the rigid body modes as we believed our supports would be compliant enough.

The contouring of pattern and hatch spacing directly affects the surface roughness and controls the frictional damping. Guided by Riju and Arulvel, We used a stripe pattern that maximises the surface roughness for damping as the laser produces 'discontinuities in melting and fusion of the powder particles' [42].
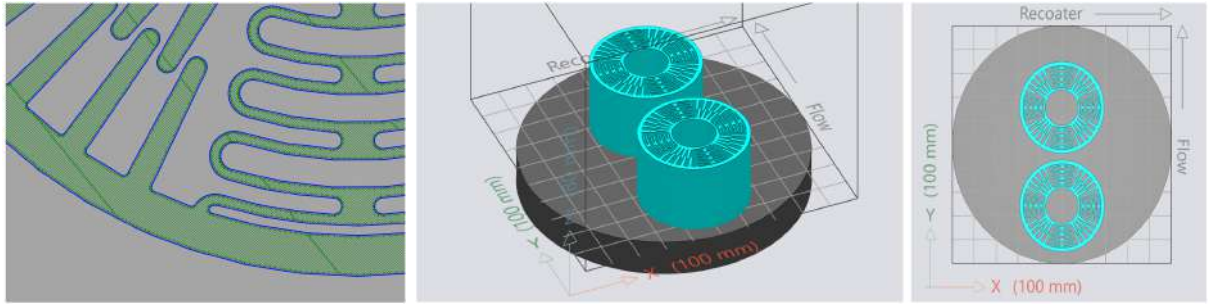
Figure 5.10: Laser contouring and Print Software.

The bearing supports were printed together and took approximately 13 hours to print. The ideal position for prints are in the bottom left quadrant of the build plate. There is maximum argon flow and the powder layer thickness is closest to expected here.
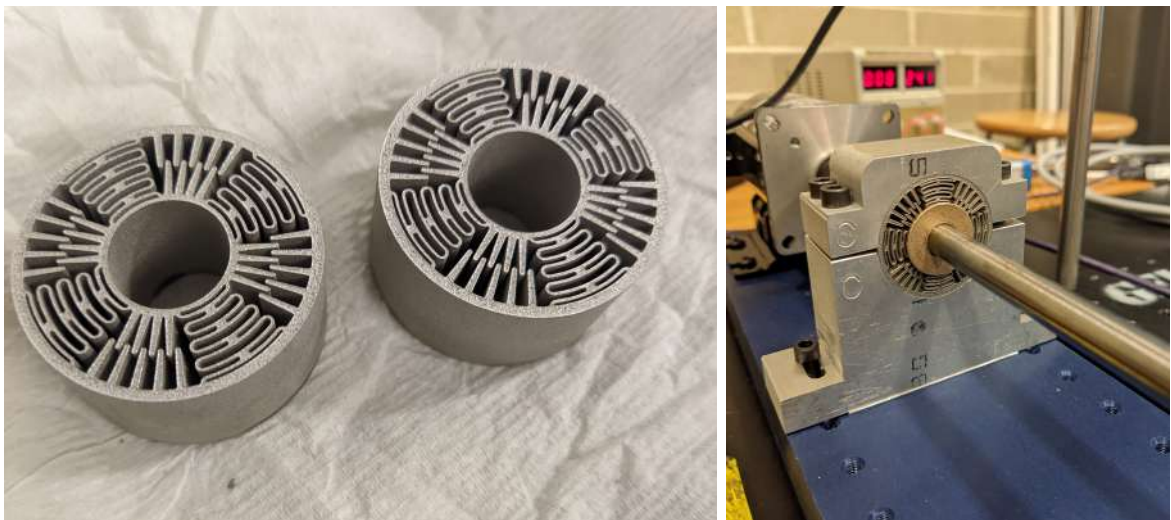
Figure 5.11: LPBF Bearing Supports.

The flashing had to be removed from the bearing supports with a pick. In the figure above, the bearing support is embedded within the pillow block.
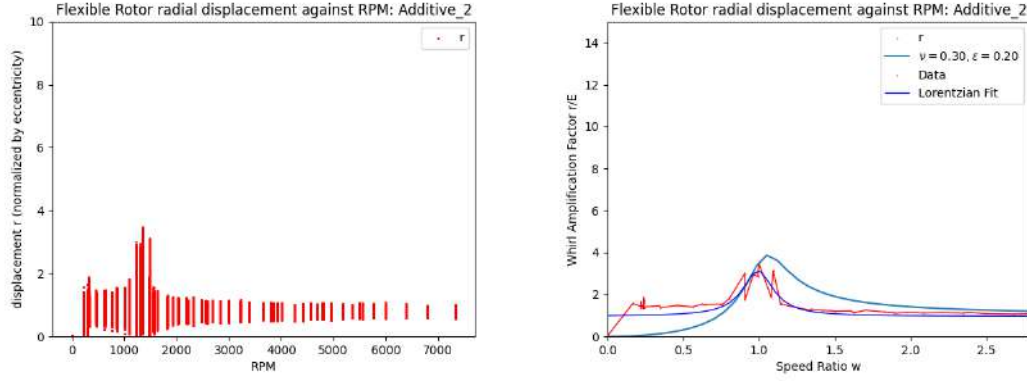
Figure 5.12: Displacement plots for Additive

Table 5.4: Summary of Material Properties, Damping Ratios, and Conditions

Material	Critical RPM	Damping Ratio
Additive	1643.484	0.03600
Additive	1362.203	0.06163
Additive	1474.459	0.04611
Aluminium	1461.625	0.09440
Aluminium	1633.062	0.08881
Aluminium	1634.292	0.05912
Aluminium	1597.251	0.04611
Plastic	1406.336	0.03342
Plastic	1755.216	0.06354
Bearing	1772.737	0.08626
Bearing	1814.718	0.24021

The additive support produced the largest damping ratio, ν, of all the materials with 0.3.

Property	Additive	Aluminium	Plastic	Ball Bearing
E ($\times 10^9$ Pa)	194.0	94.04	94.04	220
ν	0.3	0.1	0.15	0.12
ϵ	0.2	0.1	0.1	0.19
Critical RPM	1643	1462	1406	1773
W_c in RPM	1782	1041	1396	1592
W_n in RPM	2179	1517	1517	2320
B_s	93.14	18.13	36.49	33.28

Table 5.5: Material Properties and Bearing Support Data

Comparing the data, it is clear that a control 316SS bearing support should have been used. It is likely that the increase in damping came from the addition of fingers and spring sections bending at joints and absorbing some of the energy. The support damping ratio B_s shows that the additive support produced a damping coefficient significantly higher than the aluminium, plastic and ball bearing supports.

Looseness

The sounds produced by the looseness of the pillow block to the rig can be measured and automatically analysed for potential failures. The figures above show a fast fourier transform (FFT) spectra of the acoustics recorded using my smartphone originating from the rig. The noises are mostly from the shaft rubbing against the oil-lite brass bearing or from the aluminium pillow block rubbing with the rig. In a) the looseness of the fit between the pillow block and the rig produces sounds that are observed to be of low frequency and of larger amplitude than measured from quickly increasing the RPM. Larger rate of increase of RPM produces sounds of a mostly similar spectrum in b) to that of the low rate just with larger amplitude in c). There was significant background noise during the recordings and loud bangs as others were working in the background. This presents and opportunity for machine learning to detect anomalies.

Discussion

More study should be carried out to understand the critical modes observed for the different materials all cluster around 1400-1800 RPM. Especially for the plastic bearing support
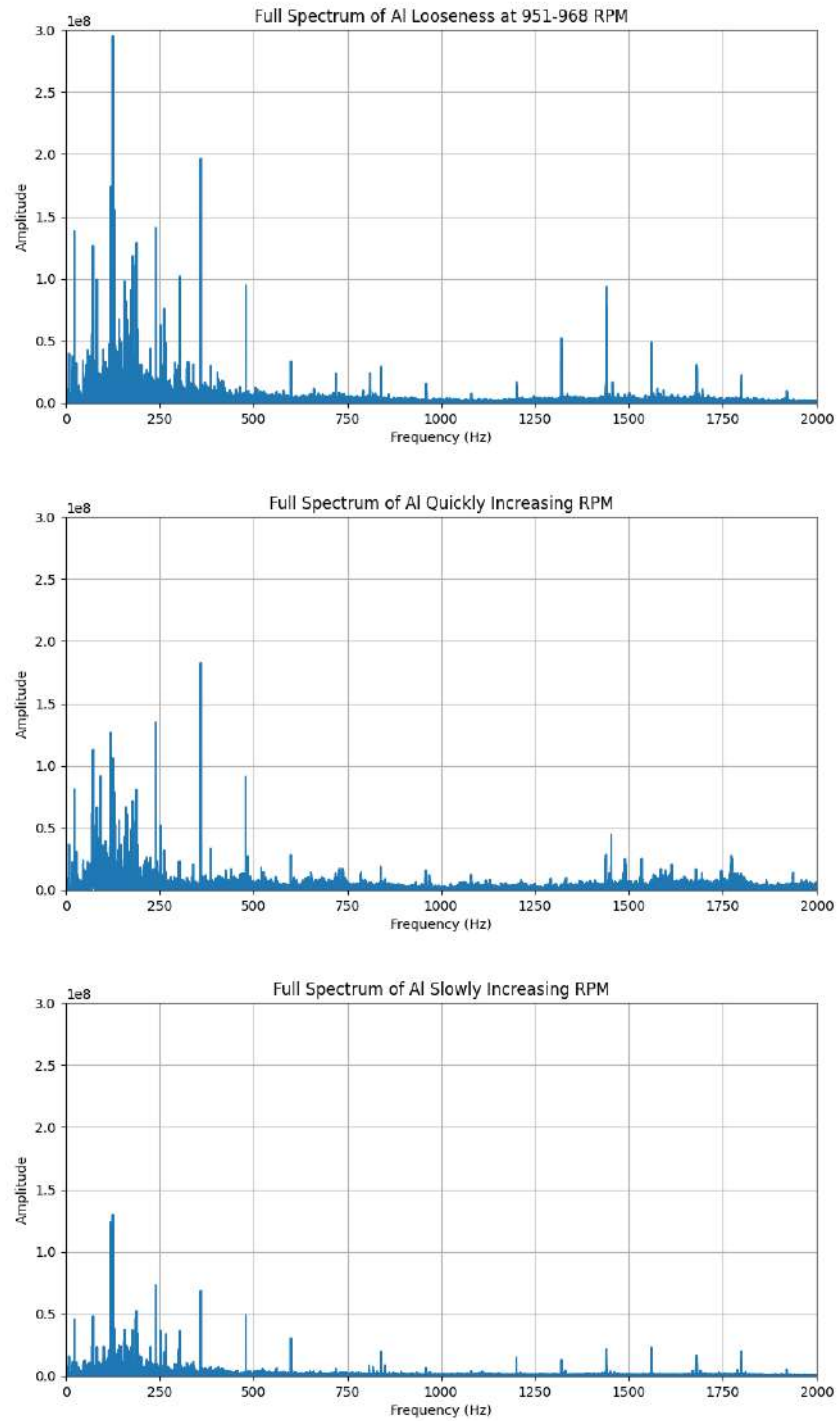
Figure 5.13: FFT Spectrum of Aluminium Looseness Sounds a) Looseness around 951-968RPM b) Quickly increasing the RPM c) Slowly increasing the RPM.

where the peak should have been at 200 RPM without the composite modification to the modulus. Instead of the bode plot, I fitted the Lorentzian curves to the plots of the dynamical responses. It did not show greater damping from the additive support but the bearing damping coefficient fit of the analytical Jeffcott Rotor model indicated significantly higher damping from the additive support. Finally, considering the solidworks simulation indicates that the supports are too stiff. The 3 dimensionality of AM prints should be leveraged to increase compliance further through integration with IPLs.

Errors

For the rotordynamic rig, quantities measured are the time during measurement and displacement in x and y axes. The shape created by the EOS M100 is very accurate due to the laser's accuracy. It had just been refurbished by EOS before the print.

The calculation of the stiffness is most likely to be the largest source of errors. All the length measurements were carried out using vernier callipers 0.05mm. However, since it is hard to measure the modulus of the materials without destructive testing and considering the variety of heat treatments, I took the values of the moduli from literature and assumed a conservative absolute error of 10GPa for further error propagation.

To calculate the uncertainty in Ω_n based on the uncertainties in k and m, we use the error propagation formula in the appendix.

Substituting 10 GPa for σ_k, 0.0005 for σ_m and the values for each material for Ω_n results;

Table 5.6: Table of Critical RPM, Stiffness, and Errors

Critical RPM	Modulus (GPa)	Absolute Error (GPa)	Percentage Error (%)
1782	194	45.9	2.58
1041	94.0	55.3	5.32
1396	94.0	74.2	5.32
1592	220	36.2	2.27

This error is actually quite low and does not affect the critical RPM significantly. As

such, better understanding of the combined stiffness of the bearing structure from the bearing itself, bearing support and pillow block would improve the fit of the analytical curves to the experimental data.

Interpenetrating Lattices

6.1 Bending dominated lattices

Bending dominated IPLs are selected for their compliance to be applied to rotordynamics. The beam-bending regime is valid when the relative density is low $\frac{\rho^*}{\rho} < 0.3$ as from Ashby's Cellular Solids [20].

Relative Density

Relative density of a cellular solid:

$$\rho^* = \frac{\rho}{\rho_s} \tag{6.1}$$

where ρ is the density of the cellular solid, and ρ_s is the density of the solid material as in Ashby's cellular solids.

To calculate the relative density from volumes, t is the thickness of the struts and ρ is the relative density:

$$V_{\text{strut}} = \pi \left(\frac{t}{2}\right)^2 L_{\text{strut}} - \pi \left(\frac{t}{2}\right)^2 t \tag{6.2}$$

The 2nd term above is to remove the overlap of the struts. Overall, the relative density of one lattice can be calculated by:

$$\rho = \frac{4V_{\text{strut}}}{V_{\text{unit cell}}} \tag{6.3}$$

Young's Modulus

Scaling relationship for Young's modulus:

$$E^* = cE_s\rho^{*n} \tag{6.4}$$

where E^* is the lattice's Young's modulus with E_s as the modulus of the bulk material, c is a constant dependent on the cell structure, and n is the scaling exponent and is 2 for bending dominated lattices and 1 for stretching dominated lattices. c is 1 for bending dominated lattices and $\frac{1}{3}$ for stretching dominated lattices.

For BCC lattices, following the approach proposed by Ptochos and Labeas and used in the work and figure below is from Lee et al. (2019) [43] employing Euler-Bernoulli Beam Bending theory derived the equation for the modulus of a BCC unit cell below.
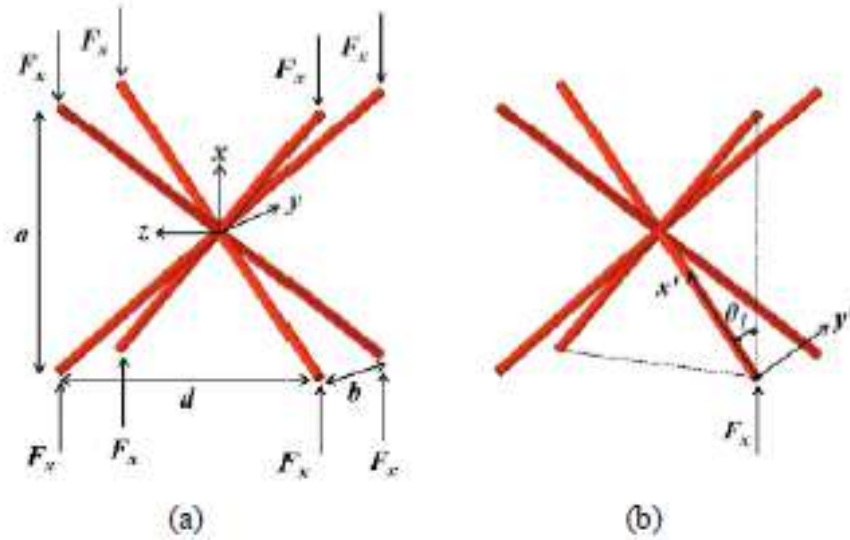
Figure 6.1: Moduli

$$E_x = \frac{4aE_0}{bd\left(\frac{a^2}{\pi r^2\sqrt{a^2+b^2+d^2}} + \frac{(b^2+d^2)\sqrt{a^2+b^2+d^2}}{12\pi r^4}\right)} \tag{6.5}$$

Speed of sound in a lattice

Speed of Sound in a lattice where the thickness of the rods are small compared with the wavelength (0.343m for 1kHz wave) where c is a constant E^* is the effective modulus and ρ^* is the relative density:

$$V_{lattice} = \sqrt{c\frac{E^*}{\rho^*}} \tag{6.6}$$

$$V_{lattice} = \sqrt{c\frac{\rho^{*n}E}{\rho^*\rho}} \tag{6.7}$$

$$V_{lattice} = \sqrt{c\frac{E\rho^{*n-1}}{\rho}} \tag{6.8}$$

6.2   Interpenetrating Lattices

Interpenetrating lattices are one or more lattices that occupy the same volume but do not touch. Combinations of different unit cells can be used such as a FCC and RD unit cell. However, BCC-BCC combinations of unit cells were used due to the simplicity and the 45 degree angles of struts being easy to print as opposed to horizontal struts that cause overhang problems in LPBF are present in Face Centered Cubic (FCC) and Rhombohedral Dodecahedron (RD) unit cells.
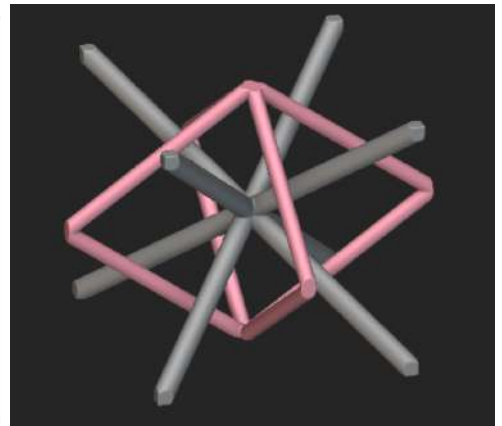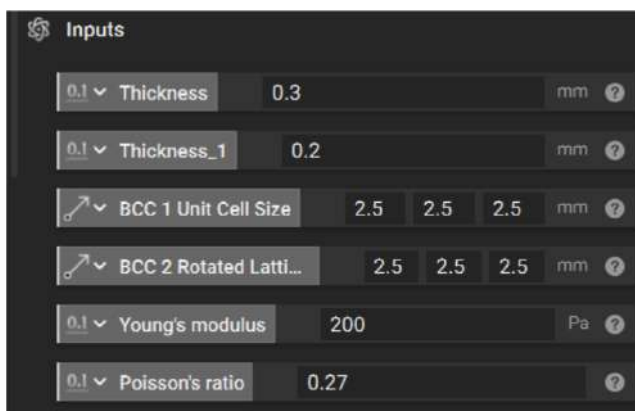
Figure 6.2: Inputs and output from NTop and custom unit cell design.

NTop allows for the creation of custom blocks, I designed the interpenetrating lattice by rotating and scaling a BCC unit cell to fit into the unit cell of another and calling the block in latter blocks as a parameterised unit cell. The 2nd unit cell coloured in red is $\frac{a}{\sqrt{2}}$ along the sides where a is the side length of the overall unit cell but the same in height. The custom blocks are parameterised so that they can be used in lists or subsequent custom blocks.
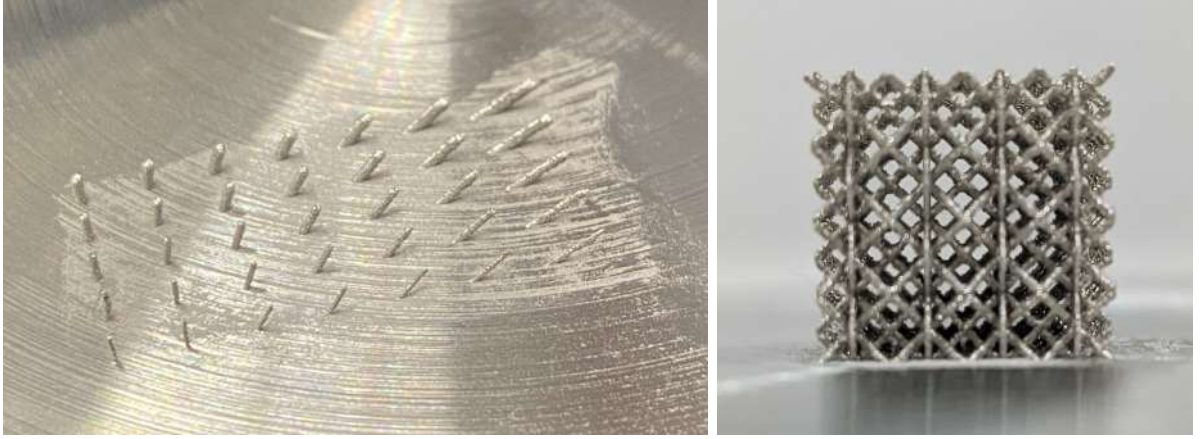
Figure 6.3: Strut parameter sweep and lattice side view.

Making use of the custom functions and lists in NTop, I created a singular strut with parameterised thickness and length and used the lists and vector block functions to create a grid of struts at different angles and thicknesses from 0 to 60 degrees from the vertical and 0.1mm in diameter to 0.35mm. This is helpful to understand what the printer's capability to print overhangs are and how much the z axis of a unit cell can be scaled. This understanding is crucial for printing IPLs that have struts closer together so that only a small load is required to activate frictional damping. I also printed a BCC-BCC IPL on the same build plate to test the capability of the EOS M100 for printing IPLs. The unit cell size referred to from now is the same as the lattice parameter.

6.3   Compression Test

Figure 6.4: First IPL printed, 2.5mm unit cell size and 0.25mm strut thickness before and after compression test.
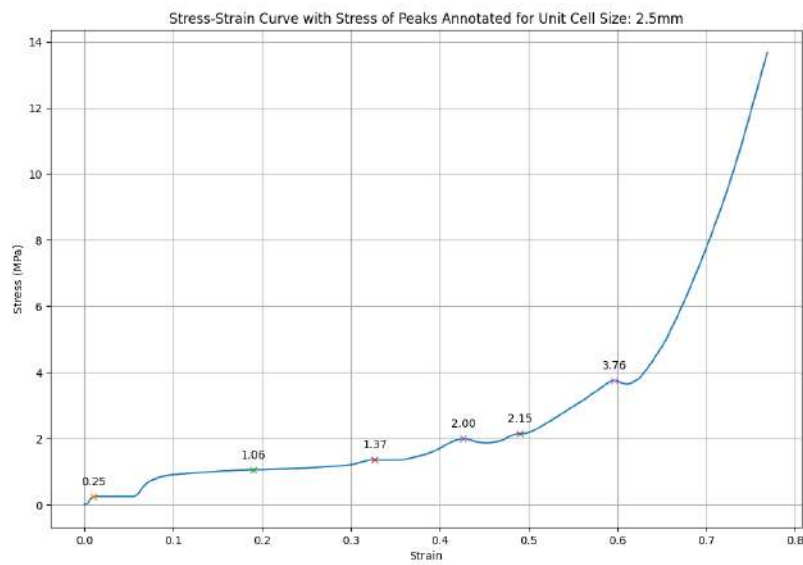
Figure 6.5: Stress Strain Curve from Instron Compression Test. Compressed to 0.75 strain.

From the stress strain curve, the Youngs Modulus can be estimated as 0.025GPa from the first stress annotation of 0.25MPa at 0.01 strain as this is the end of the elastic regime. The value calculated from Ashby's approach in cellular solids of n = 2 combining the two unit cells of relative density 0.05 and 0.06 together is far too high at 2.58 GPa whereas the approach from Lee et al. [43] using beam bending is 0.195 GPa. The modulus from the larger conventional grey unit cell is 0.051 GPa and the red unit cell is 0.138 GPa. As the modulus of the red lattice is much greater but the experimentally measured modulus is lower than either, it is likely some sort of mechanism is causing the smaller unit cell to not be activated. Multiple peaks can be observed in the curve that are likely from the

different layers of lattices buckling.

Abaqus Simulation of Unit Cell

Abaqus is a modelling software that allows for mechanical simulations of meshed objects. Ntop files have to be exported as INP files for Abaqus to inferface well. Making use of symmetry for the quarter and one eight unit cells, the boundary conditions were as follows, the bottom of the unit cell was fixed and the flat surface of the struts that were cut in half were fixed in the plane parallel to them and allowed to move in the z direction freely as U1 and U2 were restricted as seen in the figure below.
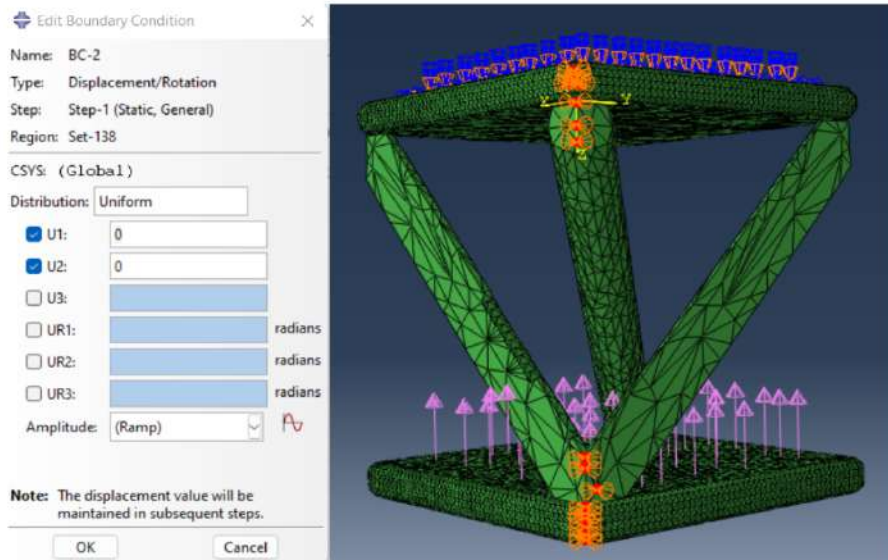
Figure 6.6: Boundary Conditions from Abaqus Simulations
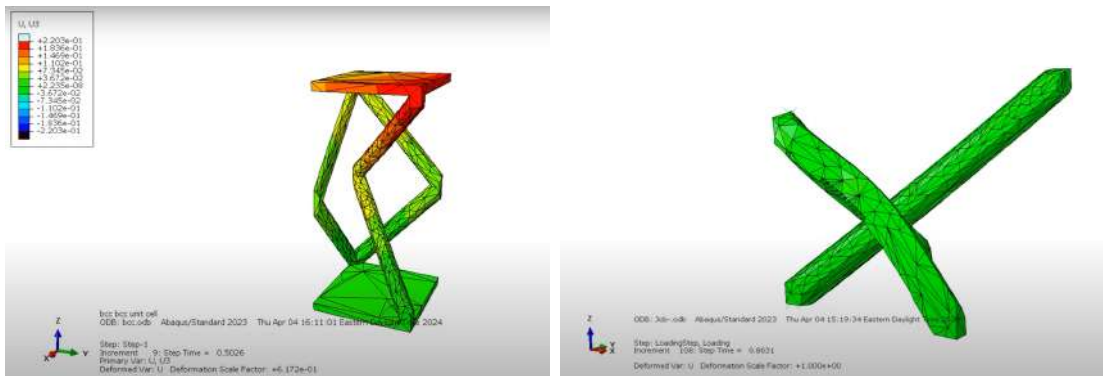
Figure 6.7: Abaqus simulations of a 1/4th and 1/8th of a BCC-BCC IPL unit cell.

Overall, the Young's modulus of the lattices was estimated by simulating a load of 200N

on the top plate and noting the strain of 0.13. The calculated Young's modulus was 0.6 GPa for a 2.5mm unit cell with 0.25mm diameter struts. This is in agreement with the modulus value of 0.057 GPa of just the conventional unit cell from the approach from Lee et al. and the value from the stress strain curve.

As observed in the part a of the figure above, the regions that experienced the most stress besides the top plate was the top strut that goes along the diagonal. This is from the conventional unit cell. This suggest that the initial deformation is focused mainly on the conventional unit cell which explains why the modulus is so much lower than expected from the theoretical calculations

Part b of the figure shows bending in a strut. This is likely due to the meshing of the surface being rather rough and jagged, resulting in a point of weakness and maximum deflection. This can be likened to a divet or cut in the metal in a real sample. And a real sample would be composed of many unit cells, averaging out the defect but a high proportion of these defects can make for relatively poor overall mechanical performance.

NTop Functionality and Prints

This allows grids of lattices parameterised to be formed. In the following image, the unit cell size and strut size are varied. The height of the lattices are also varied so that a dispersion relation can be obtained from LIRAS data. The top and bottom plates are integrated into the custom blocks.
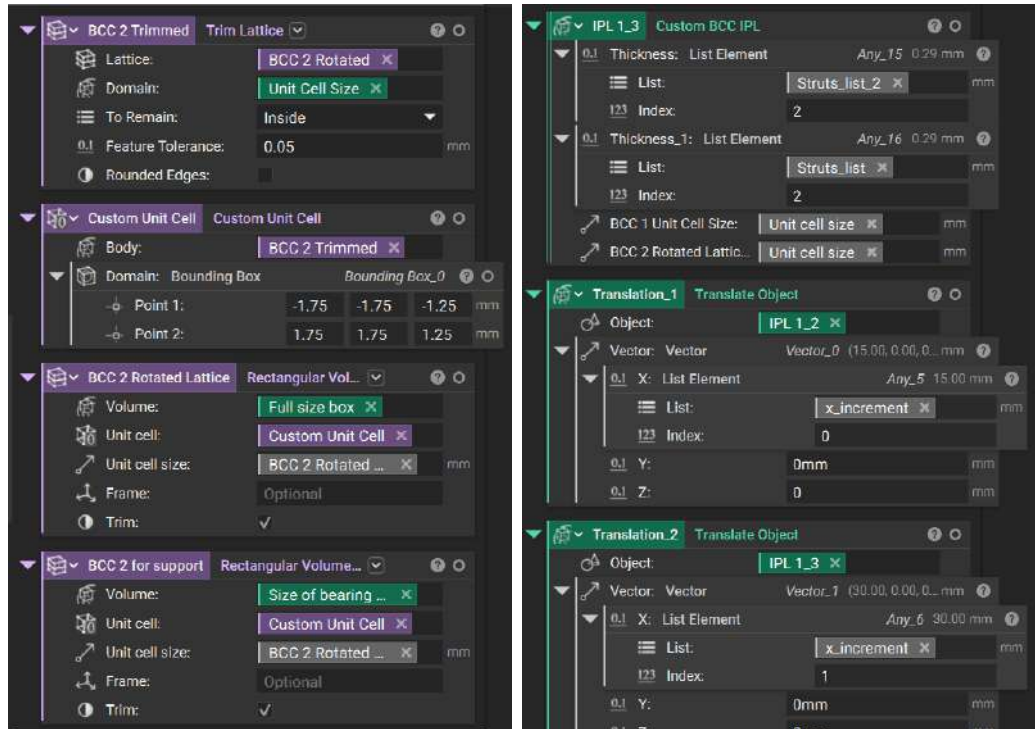
Figure 6.8: Ntop Functions

Making use of the functions within Ntop allowed the parameterisation of unit cell size, strut size, distance from other lattices.
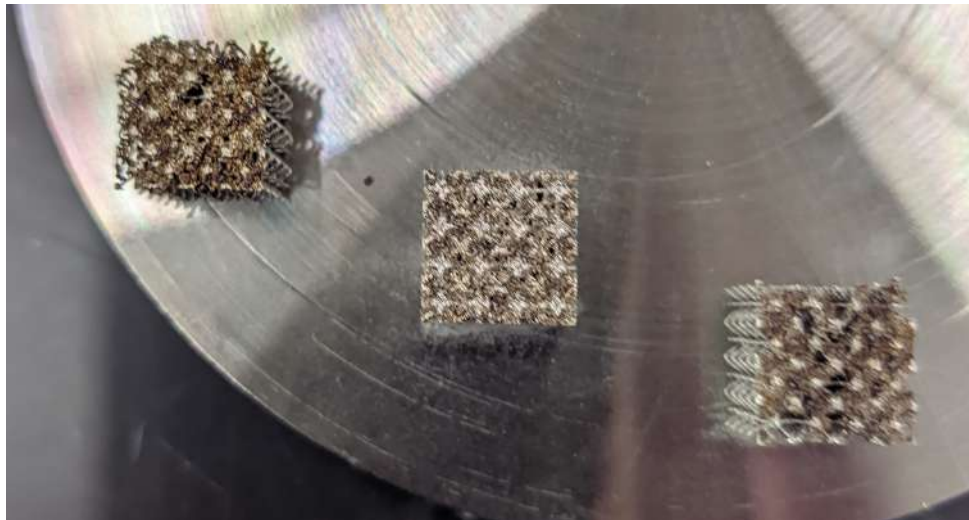
Figure 6.9: Failed top layer print.

Next, we printed more lattices of different unit cell sizes with top plates. However as the unit cell sizes were large enough to create overhangs that were not printable, the top plate failed. A suggestion going forward was to simply continue printing and allow the powder

53

to melt above it and to eventually produce a flat layer.
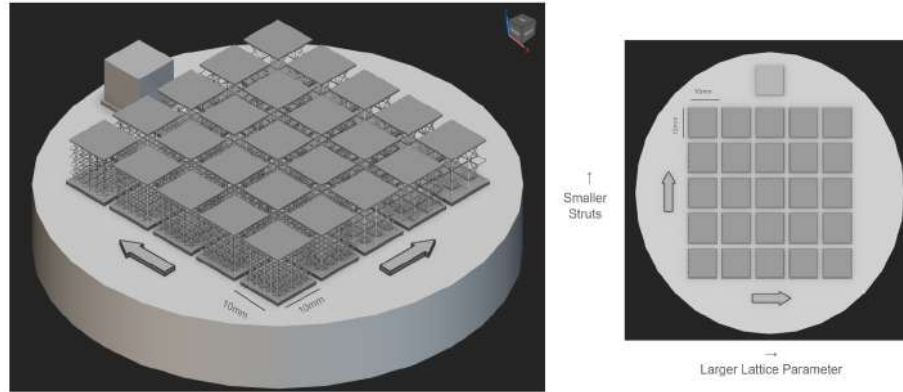
Figure 6.10: Isometric and top view of the build plate.
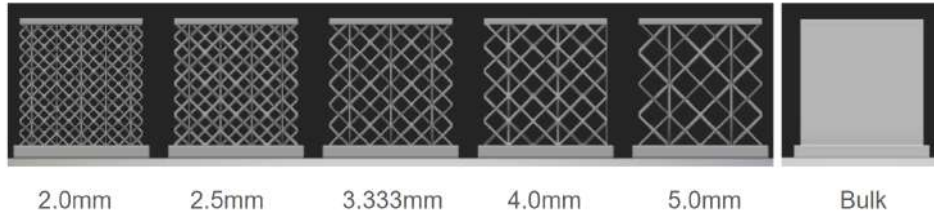
Figure 6.11: Unit cell size sweep.

The minimum size of the struts that could be printed is determined by the laser spot size and NTop. Using the custom lattice block and varying the strut size, the struts almost vanished. The strut size is also affected by the shrinking unit cell sizes. For small unit cell sizes, the struts need to still allow space in between them so that they do not fuse. Ambitiously, I wanted to push the capability of Ntop and print 5 rows of lattices with different heights along with a bulk sample. Figure 6.10 shows a grid of how the lattice parameter and strut size change. Then height of the unit cells were varied so as to be able to plot a dispersion relation for the subsequent LIRAS experiment.
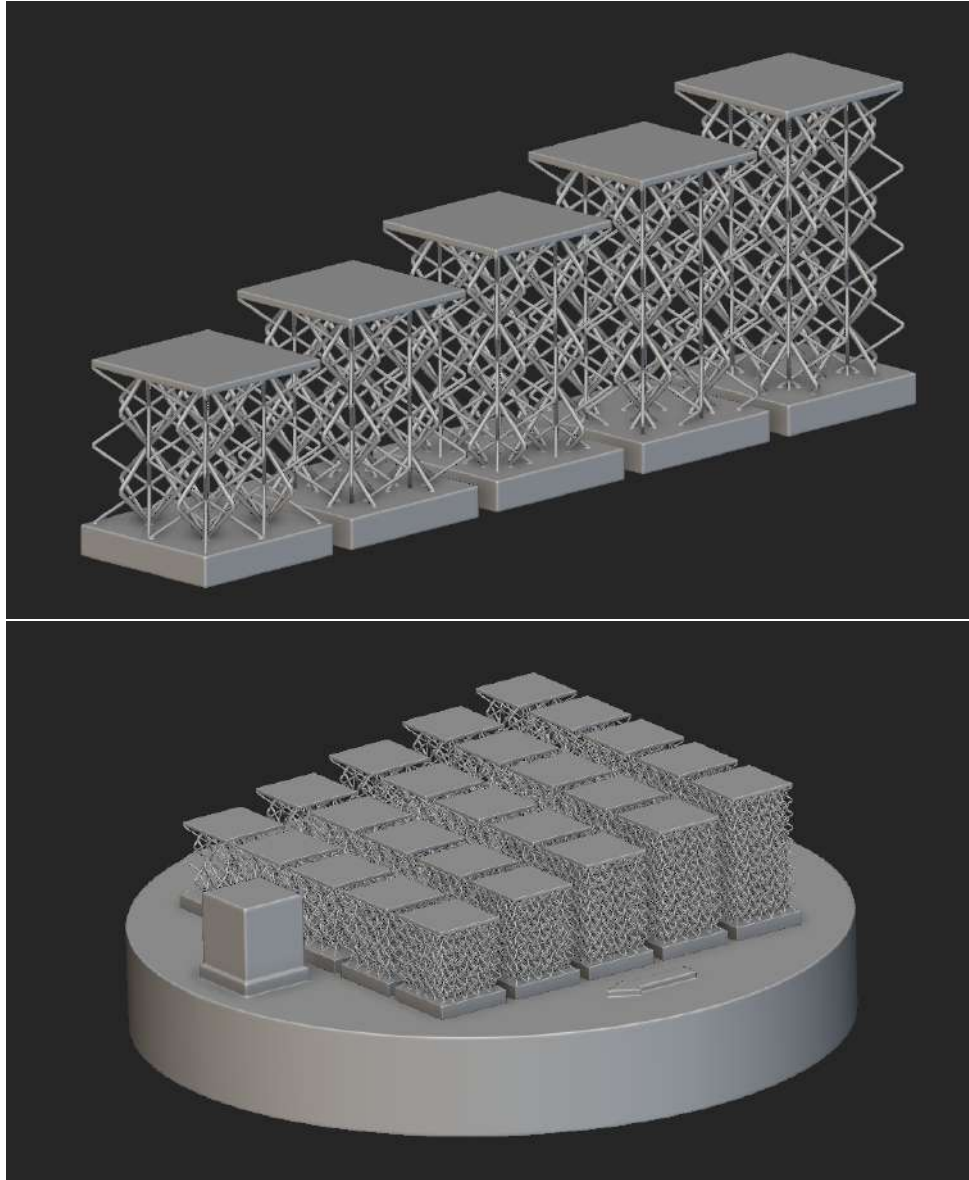
Figure 6.12: Lattices within Ntop.

I also quickly ran into problems during meshing with my RAM usage which is documented in the project management section. Upon revisiting the problem of the overhang, Roger and I were not convinced. The meshing parameters used were 0.05mm for the tolerance and 0.05mm for the minimum feature size.
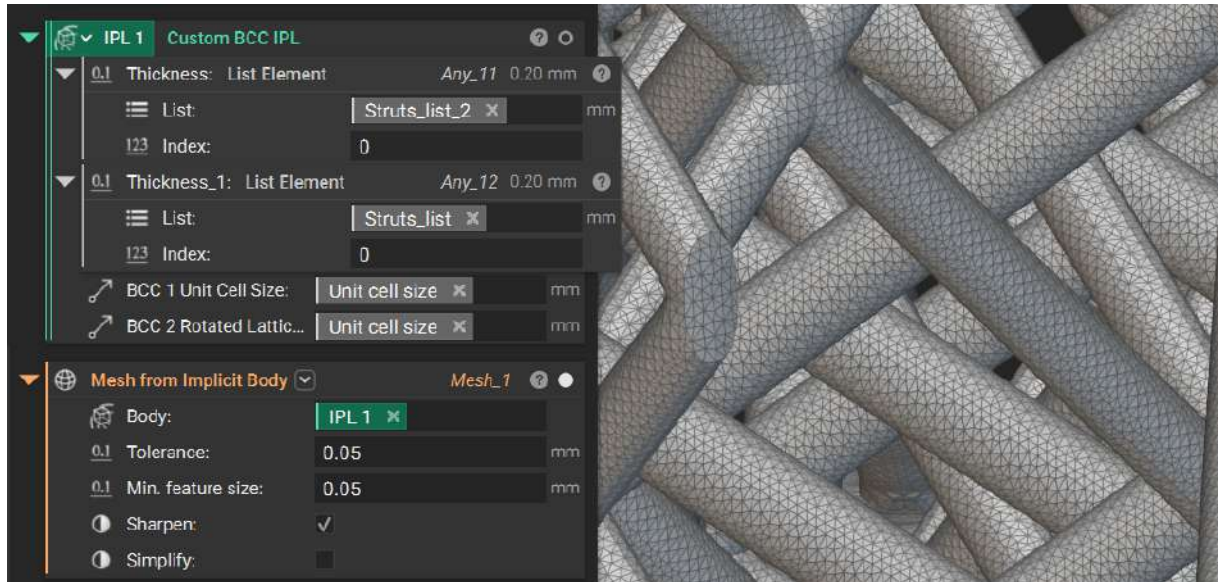
Figure 6.13: Meshing process.

It was estimated that the overhangs to print would be unfeasible. See the appendix for our decision making process to select lattices. Ultimately, we chose to reduce the sizes of the unit cell and only print 3 rows of lattices to maximise the likelihood that the printer would be able to handle the print. As such, the unit cell sizes chosen with overhangs below 2mm were 1.428, 1.666 and 2.0mm and 0.2mm for the strut size.

6.4   Print
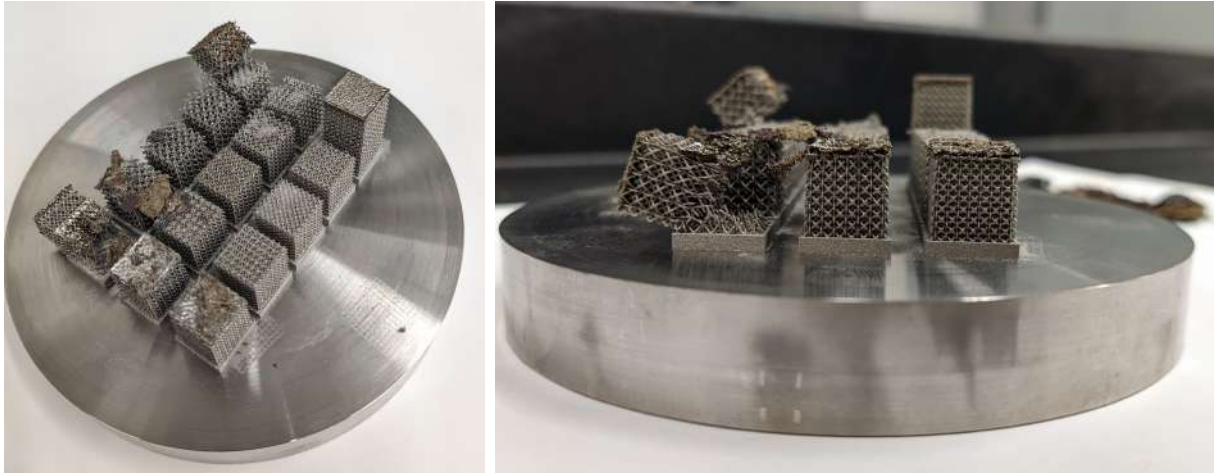
Figure 6.14: Failed print salvaged.

Lattices are labelled A, B or C corresponding to unit cell sizes of 1.428, 1.666 and 2.0mm. The number 1-5 following the letter indicates the height the lattices were supposed to be printed to which ranged from 10 to 18mm with 2mm increments. The print likely failed because the laser is not powerful enough to fully melt all the powder when the print is so large, right where there was supposed to be a top layer, the printer failed to print higher for the rest of the lattices except A5. Top layers were only successful for A1 and A5 and was partially successful for a few others so we likely had overcome the overhang issue. Any others that were printed taller are bent as observed. This is likely due to the recoater blade pushing the lattice over and breaking the strut. And occurred for two lattices in the C row as they were positioned in the rights side of the build plate. Although, material did not continue to fuse, the continual scanning of the laser above the lattices results in heating of the lattice past the initial successful printing. This would result in heat treatment of the lattices that is hard to quantify. The bottom plates were made taller before the print to alllow cutting off otherwise you could have to cut the struts off. This proved helpful as they were repurposed as the mirror surface for LIRAS.

Post-processing

57

Figure 6.15: Lattices ordered by A-C and 2-5.

The lattices were cut off from the build plate with a hacksaw with advise of Todd Billings, the senior instructor of the MIT Aero/Astro machine shop. Not shown in the figure above are A1, B1 and C1 as they were kept for reference.

Figure 6.16: Mirror surface from grinding and polishing top plate.

Next, I brought the samples back to Oxford, grinded and polished the bottom plate from 150 grit to 1 micron diamond suspension for all 15 samples a mirror like finish. This was done under the supervision of Diana Passmore in the Department of Materials Teaching

Laboratories.

Figure 6.17: a) Drying with compressed air and b) labelled in ziploc bags.

After a mirror finish was achieved, the samples were washed with deionised water and dried before placing in labelled Ziploc bags. Based on available space in the box, 12 out of 15 samples were selected to be couriered to the France using the Oxford materials department's courier service. A curious phenomenon was observed as the lattices quickly removed all water once it came into contact with the paper towel. This is likely due to the wicking and capillary effects of the paper in contact with a fully connected body of water held to the struts. The ability of the lattices to hold a fluid such as water with low viscosity provides an opportunity to increase the damping of these metamaterials.

Scanning Electron Microscopy (SEM)
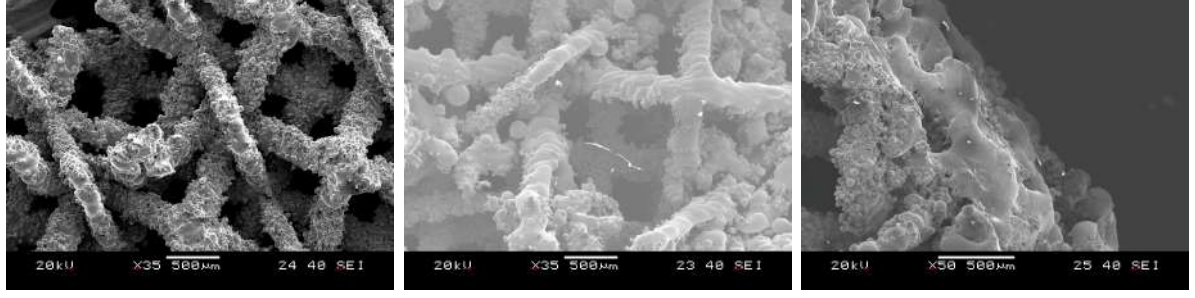
Figure 6.18: SEM images: a) x35 b) x35 c) x50.

Under Diana's supervision, I characterised the lattices under an SEM. The images show that the strut size is correct, that there is some scrap from sawing the lattices off the build plate within the lattice and the top plate of the lattices. The SEM showed the lattices were printed as expected. Surface roughness can be estimated from this.

Converting Stress Strain Curves
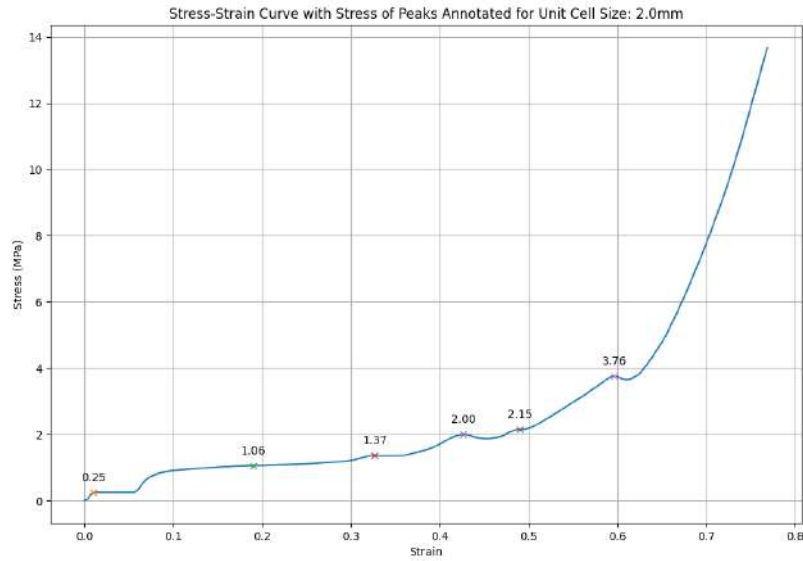
Figure 6.19: Stress Strain curve converted from figure 6.6 using relative density

We can convert the stress-strain curve from the 2.5mm unit cell size and 0.25m strut diameter using relative density factors to that of the lattices we have. This should aid future work in compressing the lattices for further LIRAS experiments. The moduli estimated for the unit cell sizes of 1.428, 1.666 and 2.0mm and 0.2mm strut size are

0.155, 0.066, 0.025GPa. The expected maximum stress is 85, 37.5 and 14 MPa. This information is useful for guiding the compression of the lattices so as to activate friction damping.

Discussion

The printer failing to print the whole lattices and top layers was unfortunate and makes gathering trends on materials properties difficult. However, by salvaging the lattices, we can study the results to guide future prints. For a future print, laser parameters could be varried for printing the top plate overhangs. A more straightforward approach would be to continue to use the bottom plates as the mirror surfaces but make them taller so that they can be easily cut off with a bandsaw. From table of lattice measurements in the appendix, the lattice masses are estimated by Ievgeniia are clustered together for each type of unit cell. The bottom plate masses vary as the grinding and polishing was done manually.

Errors

Any errors from this section are in simulation or length measurements. The meshing is most likely to contribute to errors in simulation as observed in the abaqus figure due to jagged meshing.

7 Laser Induced Resonant Acoustic Spectropscopy (LIRAS)

7.1 LIRAS

The Laser-Induced Resonant Acoustic Spectroscopy (LIRAS) data that was acquired is measured as displacement from pulses of the laser.

The heights of the lattices were measured and the mass of the structures and plates estimated in table 6.3. Following, the paper from Yun et al. [38], these quantities are useful.

$$f = \frac{c}{2H} \tag{7.1}$$

The longitudinal peaks are isolated and a plot of frequency against $k = \frac{1}{4H}$ is created. The speed of sound is used to calculate the modulus of the lattice.

$$E = \rho c^2 \tag{7.2}$$

The Lorentzian curve is once again fit to peaks to gain information about damping:

$$L(x; A, x_0, \gamma) = \frac{A}{\pi} \left(\frac{\gamma}{(x - x_0)^2 + \gamma^2} \right) \tag{7.3}$$

where A is the amplitude x_0 is the center frequency of the peak γ is the half-width at half-maximum (HWHM).

$$\zeta = \frac{\gamma}{\omega_0} \tag{7.4}$$

ζ is the damping coefficient and ω_0 is the natural frequency.

Thomas Pezeril and Ievgeniia Chaban were able to use the lattices to gain displacement data from pump probe laser pulses. Upon converting this data to the FFT spectrum the frequency of the modes are determined and the half width of the peaks are measured by fitting a Lorentzian curve to the peaks. The half width of the peaks are then used to calculate the damping coefficient of the lattices. The height of most of the samples are around 10mm except for one sample which is 18 mm. The samples all have the same strut size but different unit cell size. There were four samples each of unit cell size 1.428 mm, 1.666mm and 2 mm. The lattices were supposed to be of different heights but the print failed at around one cm for all lattices except one A5. The lattices that were supposed to be taller experienced more energy absorption due to the continual operation of the laser on the powder right above the remaining lattice. As such, these have undergone heat treatments that changed the stiffness and damping of the samples. The data mentioned allows for the determination of the speed of sound in the lattices this is an agreement with the theoretical speed of sound calculated using equation.

7.2   Results
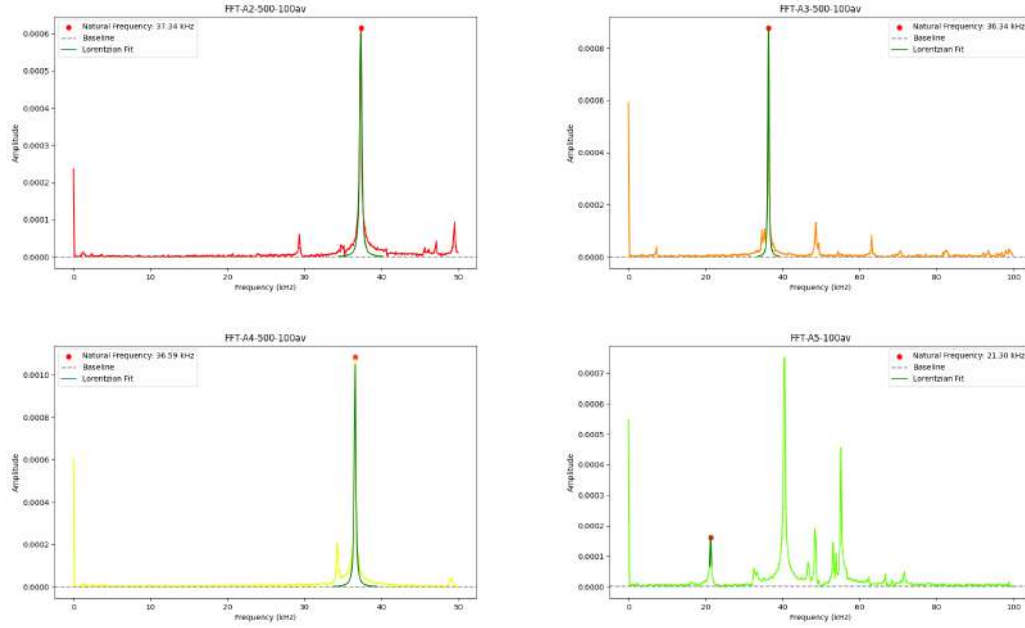
Figure 7.1: LIRAS Plots for series A.
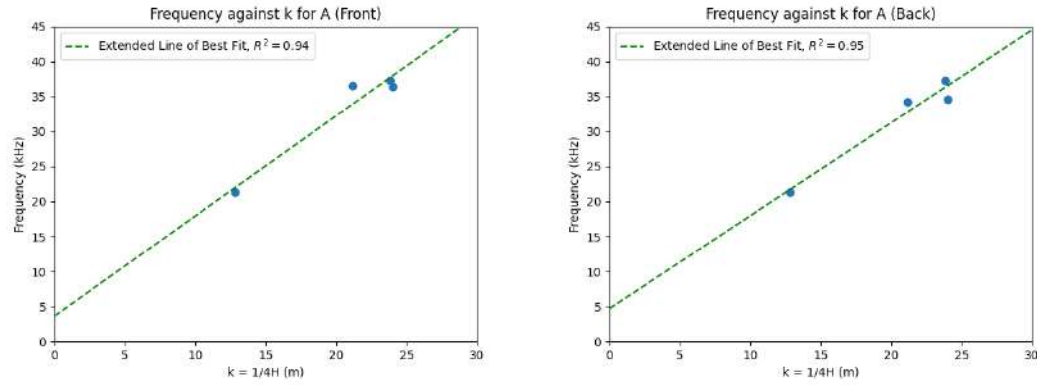
Figure 7.2: Frequency against k for A

The LIRAS Plots for the A labels show natural frequencies corresponding to the longitudianl mode at 36-37kHz for A2-4 and 21.3kHz for A5. Although, the A5 plot also shows a peak at 40kHz, the peaks are supposed to shift with the changes in the lattice height. As such, the other peaks including the one at 40.6kHz are likely other modes that were excited. Recall A5 has a large defect in the form of a groove at the bottom of the plate from my accidental sawing into the lattice. This likely contributed to the additional reflections and deformation from the laser that was detected.
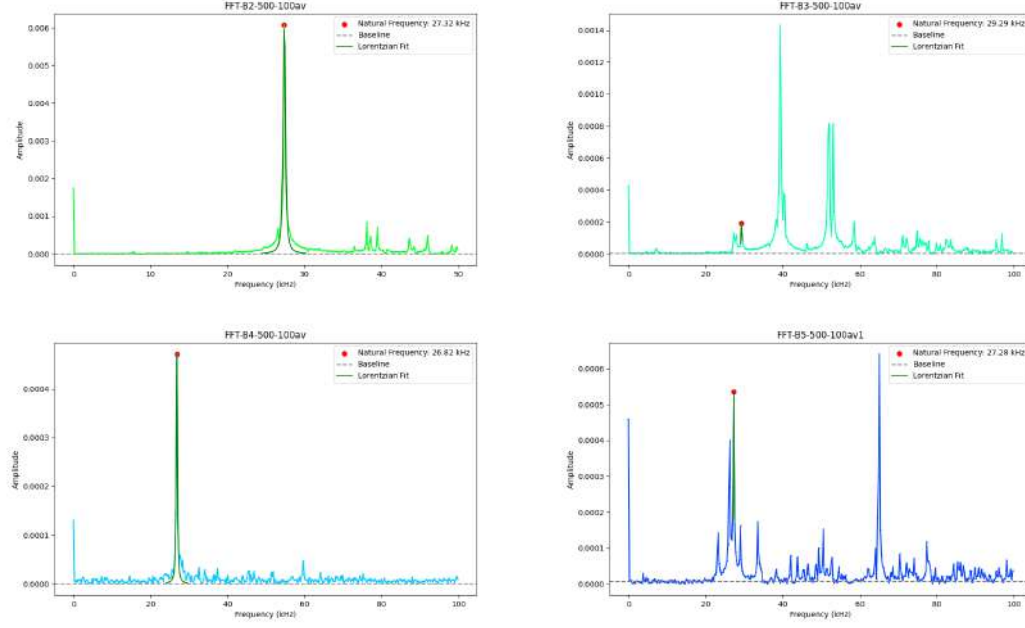
Figure 7.3: LIRAS Plots for series B.
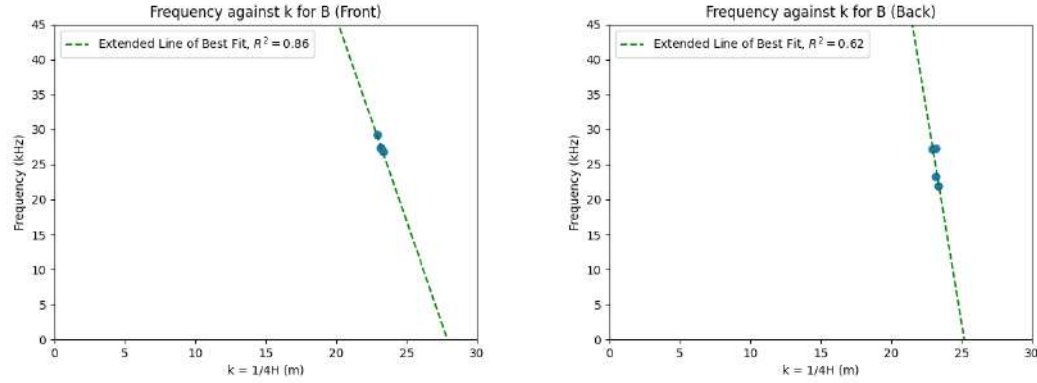
Figure 7.4: Frequency against k for B

The LIRAS plots for the B series of lattices indicate the activation of the longitudinal mode at around 26-29kHz which is where the lorentzian curve is centered and fitted for damping estimation. B2 and B4 show this as their tallest peak whereas B3 and B5 have their tallest peaks at around 52.5kHz and 65kHz. Recall, B5 was one of the lattices that was bent over, allowing other deformation modes to be excited during laser pulsing. So the peak corresponding to 65kHz is likely a bending mode.
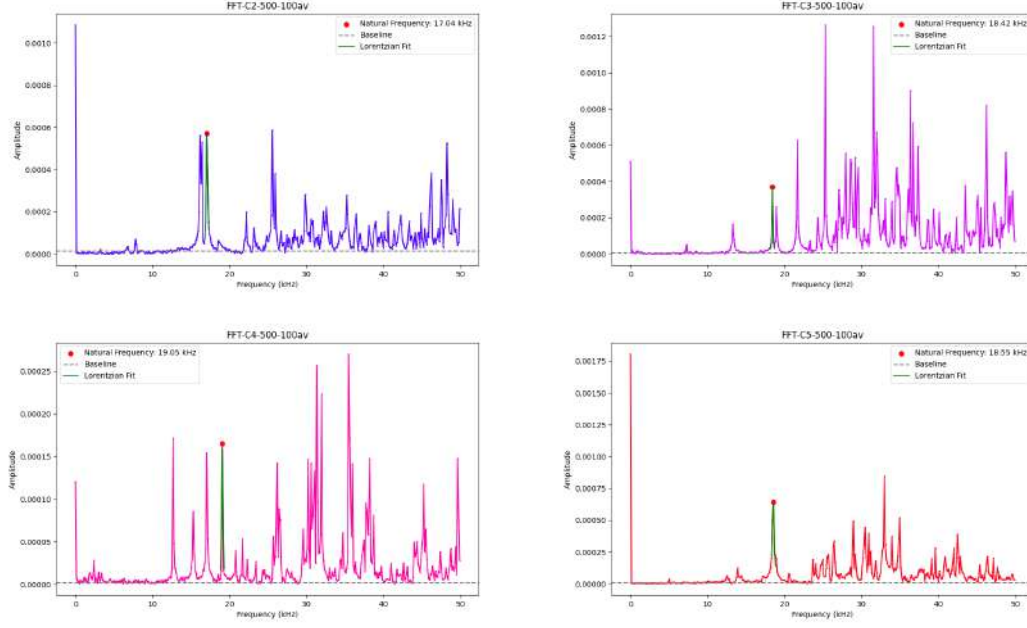
Figure 7.5: LIRAS Plots for series C.
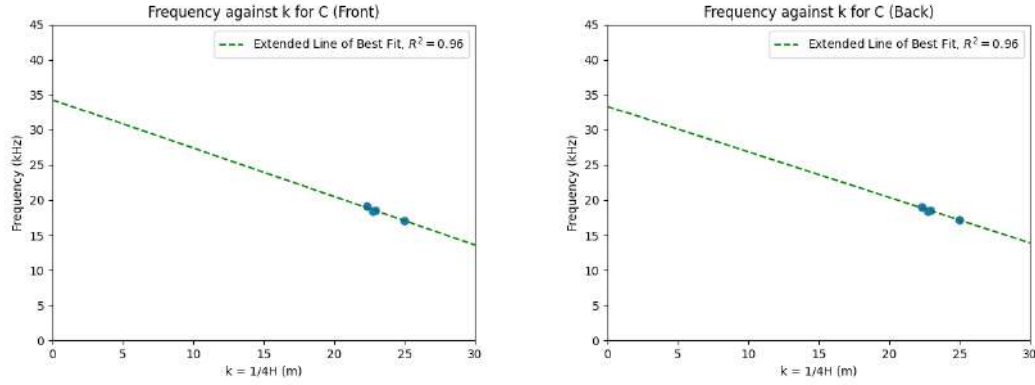
Figure 7.6: Frequency against k for C

The C plots show longitudinal peaks between 17-20kHz. This range of peaks was chosen because it was the set of peaks that was present in all plots and was the lowest for C2 and C5. However, these unit cells in general had lots of peaks suggesting many other deformation modes were excited. This was expected in C4 and C5 as they were bent but is was also observed in C2 and C3.

Comparing the plots for the front and the back of the lattice, the data behaves in a remarkably similar way considering the lack of a mirror finish. This was especially helpful for the lattices that were bent and had activate many other deformation modes. If the

measurement of the dynamical response continues to be reliable from just exciting the lattice without a mirror plate attached is to hold for different unit cell sizes and strut sizes in larger more complicated samples. This opens up the potential of LIRAS to non-destructively characterise qualified and active aerospace components.

The plot of frequency against k shows that the values selected are clustered together, making it hard to confidently deduce anything about the dispersion relation. that they represent the longitudinal mode.

The LIRAS plots where the laser was aimed at the lattice back and not the front (mirror plate) shows remarkably similar results. The dispersion relation for each of the types of lattices indicate this.

From these plots for A, B and C of the frequency of peak against the wavenumber, the plot for A suggests a linear fit and almost goes through the origin. However, due to the failed prints for B and C, the data points are cluster around 22 and 23 m^{-1}. As such, it is difficult to draw conclusions about the behaviour of the modes but they likely also follows a longitudinal mode.

Table 7.1: LIRAS Data of Lattices with their Properties for Front and Back

Lattice	Natural Freq. (kHz)	Unit Cell Size (mm)	Density$_{rel}$	Height (m)	E (GPa)	Speed of Sound (m/s)	Damping Coefficient
A2	37.3	1.42	0.225	0.0105	1.10	783	3.19
A3	36.4	1.42	0.225	0.0104	1.03	756	3.96
A4	36.6	1.42	0.225	0.0118	1.34	863	2.95
A5	21.3	1.42	0.225	0.0195	1.24	831	9.77
B2	27.4	1.66	0.166	0.0108	0.46	591	4.64
B3	29.3	1.66	0.166	0.0109	0.54	639	6.20
B4	26.8	1.66	0.166	0.0107	0.44	574	5.77
B5	27.3	1.66	0.166	0.0108	0.46	589	6.34
C2	17.1	2.00	0.115	0.0100	0.11	342	5.71
C3	18.4	2.00	0.115	0.0110	0.15	405	3.69
C4	19.1	2.00	0.115	0.0112	0.17	427	4.35
C5	18.5	2.00	0.115	0.0109	0.15	403	7.53

Table 7.2: Averaged properties of Sample Types

Sample Type	Unit Cell Size (mm)	E (GPa)	E$_{Theory}$ (GPa)	E*$_{Theory}$ (GPa)	Sound Speed (m/s)	Sound Speed$_{Fit}$ (m/s)
A	1.42	1.18	9.59	0.67	808	781
B	1.66	0.48	5.21	0.36	598	604
C	2.00	0.15	2.53	0.18	394	444

Speed of sound

As the FFT spectrums were very messy for the B and C lattices, manual inspection of the spectrum had to be done to constrain the highest peak search. Typically, across the different lattices, the peak that was most consistently present that was at the lowest frequency gave reasonable E and speed of sound results. A fit of n=2.69 and the constant c = 0.4 was used to match the theoretical values to the experimental values in table 7.2.
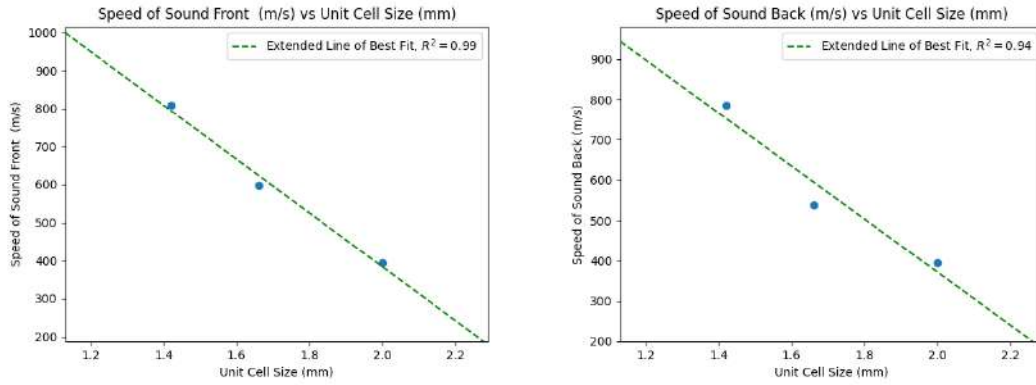
Figure 7.7: Speed of sound (m/s) against Unit Cell Size

From figure 7.7, there is a negative correlation with increasing unit cell size and the speed of sound as it is greater for the lattices with greater densities. As the speed of sound calculation varies is determined by the height and the frequency of the peak. This information is used to determine that the longitudinal mode for A5 is at 21kHz. By using a LINEST function on the data, plotting $\ln \frac{c_{sound}^2 \rho}{E}$ against lnp^*, a fit is achieved with n = 3.14 ± 0.29 and the constant c = 0.619 which errors of a max of 1.0 and a low of 0.35.

Stiffness

Observing table 7.3, the theoretical approach from Cellular Solids [20] of calculating the Youngs Modulus E and experimental value do not agree. The approach used by Lee et al. [43] combined the modulus for each of the lattices in the IPL. That being said, it seemed that the value for the lattices are 76% more, 33% more and 15% less than expected for A, B and C respectively but was in the order of magnitude. The experiment was closest

to theory for the C samples.
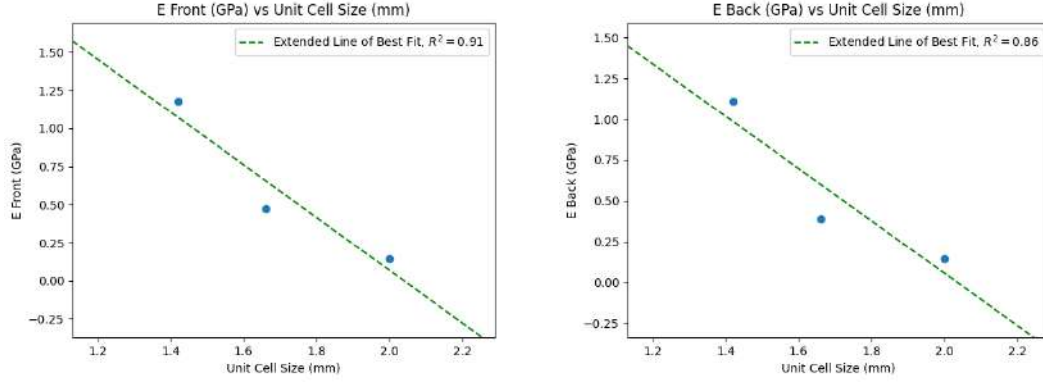
Figure 7.8: E front and back against unit cell size

Other exponents of n were attempted instead of 2 from the approach in Cellular Solids and E = 190 GP, however a good fit was not achieved. Using the LINEST function to fit lnE against $ln\rho$ results in $n = 2.88 \pm 0.18$, $E = 341$ GPa, $E_{max} = 545$ GPa and $E_{min} = 214$ GPa. Since the modulus of steel required for a good fit is much higher than 190 GPa, this method should be discarded.

Damping
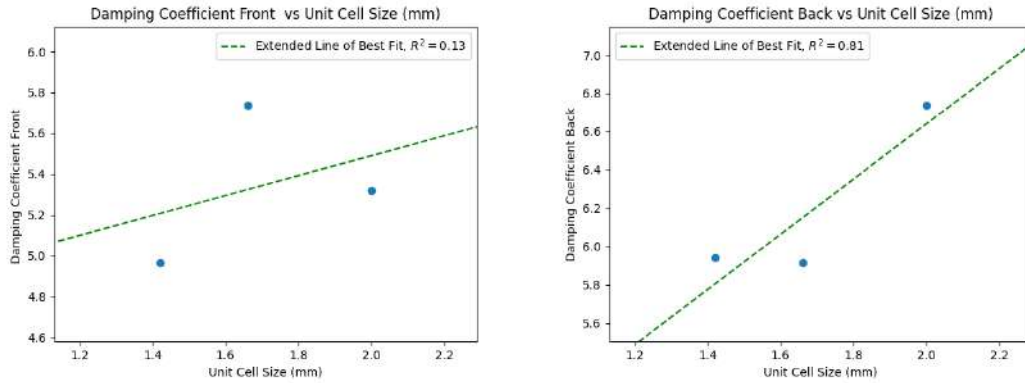
Figure 7.9: Damping front and back against unit cell size

Once again the vibrational mode studied needed to be specified. For A5, the 21kHz peak is used again. This is the first instance of deviation of materials properties between the front and the back samples. The experiment needs to be repeated to verify this as the peaks were very messy. As the lattices are not deformed to have the struts touch to acti-

70

vate frictional damping, the damping is likely related to the relative density of the lattices.

The presence of alternate modes contributing to energy loss in B and C reveals defects within lattices in high fidelity. Its non-destructive nature allows components to be inspected between operation of the rotordynamic system. The FFT spectra of the C lattices are far too messy.

An alternative method to measuring the damping is to fit an exponential decay function to the detected peaks:

$$\phi = A \cdot \exp(-b \cdot t) \tag{7.5}$$
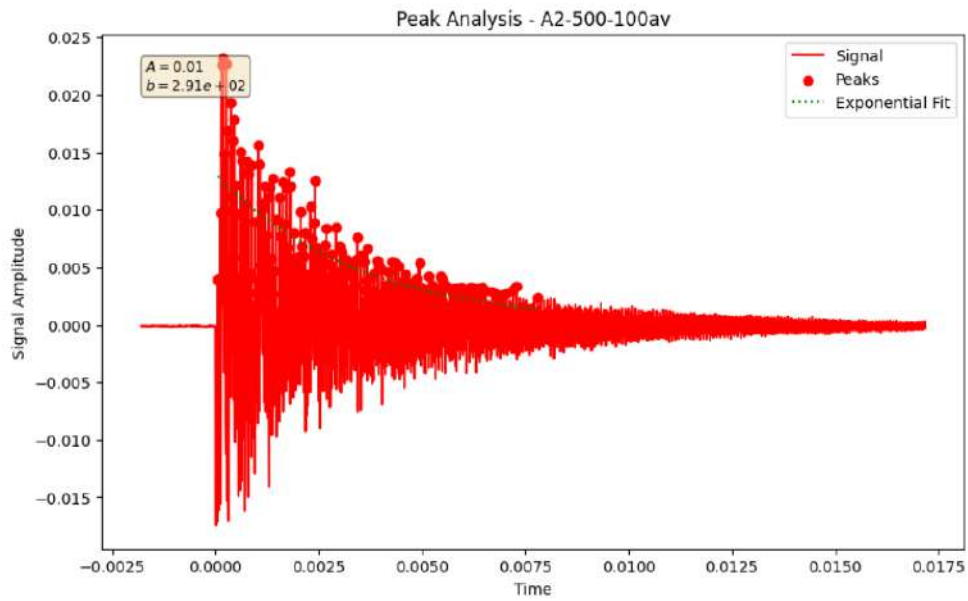
Figure 7.10: A2 damping from time series

As observed from the plot, the damping coefficient is heavily over estimated. This is true for using this method for all the samples the other samples too. So the HWHM method is recommended.

Discussion

In our data, speed of sound and modulus showed a clear trend as the calculations for the speed of sound was mainly based on the frequency of the longitudinal mode and the modulus depends on the speed of sound. The relationship for the damping is unclear. The back data is in the appendix. We used 316SS, 0.2mm strut size, had no lubrication damping. However, there is a small chance that the interface between the lattices and the samples holder in the LIRAS setup caused a small amount of damping.

A weakness of this section is the calculations and simulation for damping. As such, it is hard to interpret the data and select the right model and extraction method. Modifying the LIRAS experiment to measure the stress on the lattices to allow for the calculation of damping capacity would aid in this effort [44]. Bounds on the modulus can range from 0 to E depending on the relative density, it is clear that changing the strut size by small amounts can have dramatic changes to the relative density and the modulus.

It is hard to be sure but there is a slight upward trend of stiffness from A2, B2, C2 lattices against the subsequently numbered lattices. This could be due to a number of factors but it likely has something to do with being exposed to the laser for longer as the values are more than 10% different.

Errors

Errors are most likely to result from the setup of the laser. The half width is in part determined by the amplitude of the response. The time and resolution are likely to also contribute to errors.

8

Integration and future work

8.1 Future Work

Rotordynamics

The stiffening effect of the brass oil-lite fitting and pillow blocks on the bearing supports is a crucial area of study as they directly affect the critical RPM of the critical modes. Effective Young's Moduli needs to be measured for this. More real rotordynamic experiments can be carried out with additional rotors and balancing with focus on understanding the whirling modes and comparison to more dyrobes simulations. Bulk 316ss samples should be used as controls and other materials such as inconel can be analysed too. The fatigue and creep properties are crucial for ensuring reusability and long-life.

Interpenetrating Lattices and LIRAS

The next print should once again involve lattices of different heights to replot the dispersion curve and study the different deformation modes. In-situ LIRAS experiments of deformation to activate frictional damping should be carried out guided by the compressive stress strain data. The effect of the mirror surface should be verified so that the conditions under which LIRAS can produce a clean dynamical response is understood. Damping that arises from other deformation modes should also be estimated and related to lattice geometry.

8.2 Integration

With compliance and damping mechanisms. The first iteration had a large number of springs with small spacing, small number of long fins. There was concern about fins bottoming out or spring beams getting stuck. The second iteration involved large spring beam spacing, a large number of shorter/thinner fins, thinner outer and inner rings as a result.
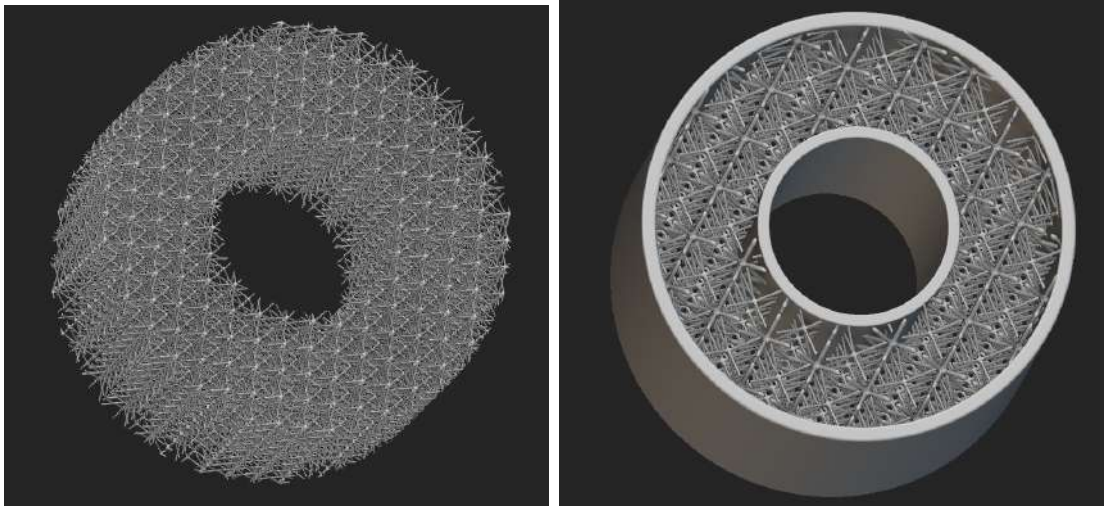
Figure 8.1: Lattices for bearing supports.

Our calculations indicate that a bearing support with internals made up fully by IPLs would be too stiff and not be in the range to produce rigid body modes. As such, a modified design of the meta material bearing support in chapter 5 is suggested.
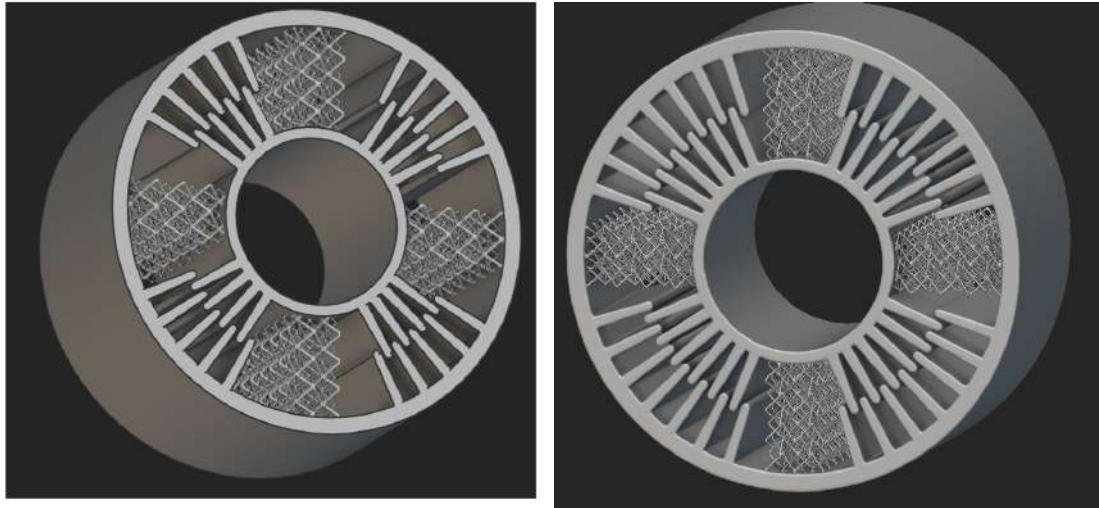
Figure 8.2: Integrated bearing supports.

Design Considerations for Lattice Integration

When integrating AM lattices into bearing supports, the material chosen for the AM lattice must be compatible with the operating environment, including temperature ranges and lubrication interaction. The design must ensure that the lattice-augmented components can withstand the operational loads without compromising the turbomachinery's reliability or lifespan. As from the economics section of the appendix, if the lattices are simpler and can be made more cheaply with sandcasting rather than LPBF if there is a demand for over 8500 bearing supports.

As shown from the LIRAS section, the sample does not require a mirrored surface to study its dynamical response. As such, future additively manufactured IPL-integrated bearing supports should be studied with LIRAS.

Design with plastic 3D printing

The time and effort spent setting up the EOS M100 makes it a time-consuming tool for rapid prototyping. The build space is limited and removing the component from the build plate can be a challenge. I suggest testing designs of lattice structures and bearing supports through plastic 3D printing first. To validate any simulation models and further understand the rotordynamic effects from the bearing supports before moving to a metal

print with gained confidence.

9

Conclusions

9.1 Conclusions

The individual components of the project were successful in demonstrating the feasibility of the method in achieving the goals. However, the damping is yet to be well related to the geometry of the interpenetrating lattices or the metamaterial bearing support structures.

Going back to our initial goals:

1. Assess the rotordynamic behaviour of bearing supports of various materials in relation to additively manufactured components

2. Relate stiffness and damping to the geometry (relative density, unit cell size, etc.) of additively manufactured metal interpenetrating lattices (IPLs).

3. Develop bounds on stiffness and damping of AM IPLs.

4. Integrate AM IPLs with bearing support structures.

We were able to achieve 1), 2) and 3). The integration of bearing supports in Ntop has been successful for 4) but the printing and rotordynamic experiment still needs to be carried out.

The rotordynamic rig was able to measure the response profile of the different bearing support materials consistently and show that the additive supports had greater damping

than the other materials. However, as in the future work section more work need to be done on the looseness of fit. 78

Laser powder bed fusion proved to be an immensely effective technique at fabricating rotordynamic bearing supports and interpenetrating lattices.

A surprising success is the LIRAS section of the project. It proved better than expected in measuring stiffness and damping on the side of the samples without the mirror surface and also quite accurate in predicting the speed of sound.

10

Final Chapter

10.1 Health, safety and risk assessment

Additive manufacturing: The main risk was from using the EOS M100 printer as the powder is hazardous. I had training 3 sessions of training to use the printer and had to wear the appropriate PPE including goggles and N-95 mask or ventilator, workwear coveralls and safety shoes. Gloves were worn and anyone else in the laboratory was instructed to stay far away from anyone preparing powder or cleaning up powder. Hazardous waste was discarded safely in containers. During training, safety procedures were heavily emphasised so that I was able to carry out the process alone later.

Post Processing: Goggles, labcoat and gloves were worn. Gloves were changed when there was any holes from contact with a pad.

Compressive test: Goggles were worn during operation. The test was carefully observed so that excessives loads that could lead to failure of the machine would not occur. My laboratory partner helped setup the Instron machine and informed me of all the safety procedures.

Rotordynamics: Goggles were worn during operation. Looseness of pillow blocks and bearing supports was kept to a minimum to prevent any disassemblies and flying parts. Similarly, excessive vibrations from the rig could cause parts of the table to come loose

and collapse. Care was taken not to stand exactly next to the rig in the axis of the rotor. Any contact with the rig was on parts that were not moving to prevent burns from friction. During training, safety procedures were heavily emphasised so that I was able to carry out the process alone later.

LIRAS: Thomas Pezeril and Ievgeniia Chaban followed their safety procedures.

10.2   Ethical and sustainability considerations

Ethics: Turbomachinery is a technology that has been used for good and bad. The acceleration of the development of this technology occurred during WWII by the Germans and the British. They were used in planes, missiles, ships and trucks that allowed for military strategy, manufacturing and logistics advantages. In present day, advancements in turbomachinery have led to improvements in engines such as the Praff & Whitney F135 used in vertical takeoff in the F35. However, this technology has also been used to achieve incredible feats in aerospace for going to the moon, providing initial energy to propel satellites out of the Heliosphere in manufacturing for turbomolecular vacuum pumps for extremely pure semiconductors and in household vacuums. All of this solidifies turbomachinery as a technology with extreme applications. As such, this technology is heavily export controlled.

Sustainability: Additive manufacturing uses more energy compared to conventional manufacturing processes at every step. However, being able to save the powder that is not lasered by sifting, allows for less material wastage and degradation of properties through the recycling processes. Any scrap created in the machine shop was put through its recycling process. The experiments carried out can be used to test for fatigue and creep, two crucial parts to the reusability of rocket engines and aeroengines, thereby extending the life of large aerospace systems. Improvements in performance can also increase fuel efficiency that result in reduced emissions.

10.3 Project Management

Initial Objectives and Milestones: My project initially focused solely on lattices. However, as my lab partner moved on to other projects, I took on the additional responsibility of designing bearing supports to work with the rotordynamic rig and analysing the rotordynamics. Professor Cordero's 16.101 Topics in Fluids: Additive Manufacturing for Aerospace Engineers was tremendously helpful for this. This broadened the scope of the project to include rotordynamics, but the core objectives of investigating the printed lattices by LIRAS remained largely unchanged.

Throughout the project, I set specific milestones to track progress. These included designing the lattices, printing them, and analysing the data to estimate the Young's modulus. While I achieved some milestones on schedule, there were delays in others due to unforeseen challenges.

Challenges and Back-up Plans: One of the anticipated challenges was navigating the difficulty of interfacing between Abaqus and Ntopology for simulations. However, the most significant obstacle I encountered was the extended downtime of the printer and the limitations of my laptop's processing power. These limitations hindered my ability to print and run simulations efficiently and troubleshoot issues promptly.

When the printer broke down, I continued working on the Abaqus simulations and NTop design. Additionally, to overcome the limitations of my laptop, I borrowed Suhas's computer towards the end of the project, allowing me to rerun Abaqus simulations.
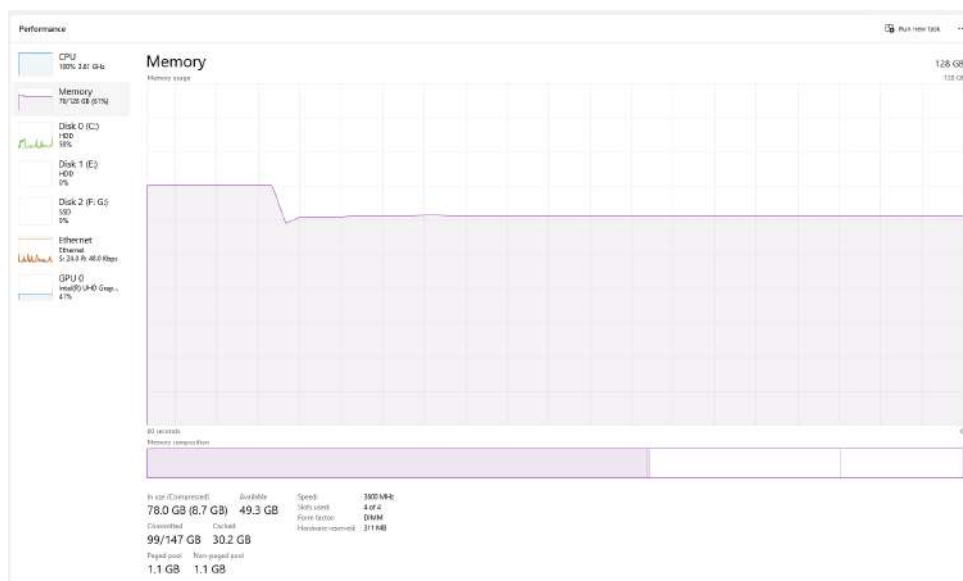
Figure 10.1: RAM usage during operation.

My usage of RAM on my personal computer that only had 12GB of RAM was a hindrance to meshing large grids of lattices. It caused hours long wait times for meshing before crashing. Using Suhas' much more powerful computer with 128GB of RAM. The final meshing of 5 rows of lattices of different heights, a bulk sample and arrows to indicate enlarging parameters required 78GB of RAM to mesh and 8GB of storage for the STL file and was reduced to mere minutes.

While putting the samples back into the ziploc bags after drying, I mixed up a few samples as I was drying the samples on the heating rack at the same time. By inspecting the mass measured and images from Ievgeniia, I was able to remap them back to their proper labels i.e B3 to A2 and A2 to B3.

```
Found 12 CSV files in Front Data.          Found 14 CSV files in Back Data.
Mapped Filenames:                           Mapped Filenames:
A2-500-100av.csv -> B3-500-100av.csv        A2-500-100av-back.csv -> B3-500-100av-back.csv
A3-500-100av.csv -> A3-500-100av.csv        A3-500-back-100av.csv -> A3-500-back-100av.csv
A4-500-100av.csv -> B4-500-100av.csv        A4-500-100av-back.csv -> B4-500-100av-back.csv
A5-100av.csv -> A5-100av.csv                A4-500-100av-back1.csv -> B4-500-100av-back1.csv
B2-500-100av.csv -> B2-500-100av.csv        A5-100av-back.csv -> A5-100av-back.csv
B3-500-100av.csv -> A2-500-100av.csv        B2-500-100av-back.csv -> B2-500-100av-back.csv
B4-500-100av.csv -> A4-500-100av.csv        B3-500-back-100av.csv -> A2-500-back-100av.csv
B5-500-100av1.csv -> C5-500-100av1.csv      B4-500-100av1-back.csv -> A4-500-100av1-back.csv
C2-500-100av.csv -> C2-500-100av.csv        B5-500-100av-back.csv -> C5-500-100av-back.csv
C3-500-100av.csv -> B5-500-100av.csv        B5-500-100av-back1.csv -> C5-500-100av-back1.csv
C4-500-100av.csv -> C4-500-100av.csv        C2-500-100av-back.csv -> C2-500-100av-back.csv
C5-500-100av.csv -> C3-500-100av.csv        C3-500-100av-back.csv -> B5-500-100av-back.csv
                                            C4-500-100av-back.csv -> C4-500-100av-back.csv
                                            C5-500-100av-back.csv -> C3-500-100av-back.csv
```

Figure 10.2: Front and Back Mappings for LIRAS files Successful

As there were multiple sets of files to remap, I wrote a function that mapped the files to indices ranging from 1-12 first to avoid a conflict that arises from having identical files then mapped them to their proper labels.
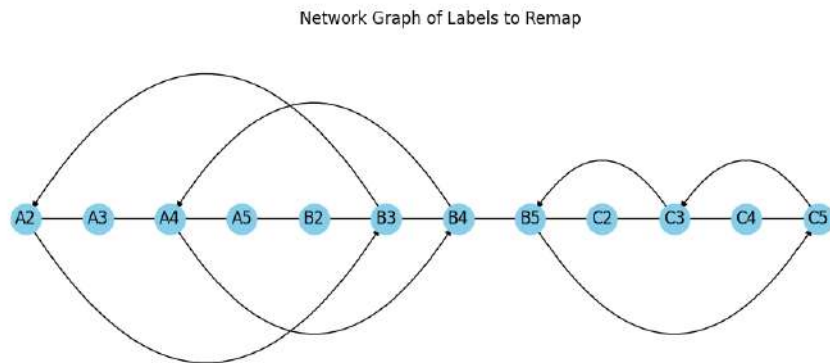
Figure 10.3: Network Graph of Labels of Files to Remap

I was careful to not run the function twice as the files do not map completely back to the previous state.

Resource Planning and Availability: 316 stainless steel was readily available in the lab. Regarding equipment, I had convenient access to most tools and machines within the lab. However, booking time on the SEM and wire EDM equipment proved challenging due to scheduling requirements for training. Instead, I used a hack saw in the Aero/Astro

machine shop to cut samples from the build plate instead of relying on the wire EDM which would have required 36 hours to cut. Additionally, I carried out the grinding and polishing tasks in Trinity Term in the materials teaching labs at Oxford as I was fully trained in metallography and SEM usage.

In terms of human resources, I sought permission and regularly discussed plans with Prof Cordero. I also had frequent meetings with Roger Hou and benefited from the expertise of others, including Tim Stubbs, Matthew Shorter, Colin Pavan, Thomas Pezeril, and Ievgeniia Chaban, for various tasks throughout the project.

Collaboration and Decision Making: While I primarily worked independently, the project required collaboration and support in specific areas. I relied on the supervision and guidance of Roger Hou and Tim Stubbs when using the printer and other equipment. Additionally, I borrowed Matthew Shorter's laptop to use the necessary LabVIEW scripts.

For project direction, I made most decisions in consultation with Prof Cordero, ensuring his approval at each step. I also maintained regular communication with Roger Hou to discuss plans and approaches. The project benefitted from these collaborations, and we often had more frequent meetings when finalizing plans or preparing for printing.

Documentation: Throughout the project, I documented my work by keeping notes within a Google Drive. This record-keeping practice was useful in tracking my progress and referring back to key decisions made along the way in writing my thesis.

Lessons Learned: Reflecting on the project, there are a few key takeaways. In hindsight, it would have been advantageous to prioritize getting the lattices printed sooner. Additionally, advocating for continued access to the printer after the printer mishap would have minimised delays caused by relying on others for printing tasks. Using a more powerful computer sooner would have allowed me to execute my simulations earlier on.

Due to collaborators working on the project as well. They named files relating to aluminium as áluminum' in the American spelling causing great confusion as my code would not understand how to parse and categorise the file. This provided a humorous chuckle upon discovery.

10.4   Project Management Forms

PROJECT MANAGEMENT FORM 1

Part II Project Description Form

After discussion with your supervisor YOU should complete this form and send a copy to the Academic Administrative Office by Friday of 0th week of Michaelmas Term.

Name: Zhen Rong Yap   College: Trinity College

Address for correspondence: 362 Memorial Drive, Amherst Way, Cambridge Massachusetts.

Contact telephone number: +447376278333 / +18577636328

Title of project: Additively Manufactured Interpenetrating Lattices with Tailorable Stiffness and Damping

Supervisor: Prof Zachary Cordero, Prof Michael Moody, Dr. Yuanbo (Tony) Tang

What are the objectives of the project in order of priority?

1) Assess damping behaviour of additively manufactured metal interpenetrating lattices (IPLs)
2) Relate stiffness and damping to geometrical (relative density, unit cell size, etc.) and surface roughness effects
3) Develop bounds on stiffness and damping of AM IPLs

List the major milestones that must be accomplished in order to meet the objectives of the project

1) Print lattices (Oct.)
2) Characterize structure of lattices + base metal (Oct.)
3) Develop computational model of lattice dynamical properties (stiffness and damping) (Oct.-Jan.)
4) Characterize dynamical properties of lattices as functions of temperature, frequency, oscillation amplitude (Nov.-Jan.)
5) Develop experimentally-validated analytical model of dynamical properties (Jan.-Mar.)
6) Predict bounds on lattice properties (Apr.)

Are you working essentially on your own or as part of a team? If you are part of a team what is your role, and to what extent is the success of your project dependent on other members of the team?

With an MIT senior that is doing UROP.

What resources (equipment, materials, technician support etc.) will you need?

316SS. Instruments using EOS M100. LIRAS. Oven for annealing.

Do you require any training to meet your objectives, e.g. in the use of specific experimental equipment or software, and how are you going to obtain that training?

Training on additive manufacturing EOS M100. Trained on data analysis software for the technique.

Complete the following plan for your entire project as you see it now. List each major task down the left hand column, and for each one draw a horizontal line to indicate the period you expect to allocate to it. For example, the final task, writing your thesis, is shown as occupying mid-April to mid-June.

Task	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Print lattices	Y								
Characterize structure of lattices + base	Y	Y							
Develop computational model of lattice dynamical properties (stiffness and damping)	Y	Y	Y	Y					

Characterize dynamical properties of lattices as functions of temperature, frequency, oscillation amplitude		Y	Y	Y					
Develop experimentally-validated analytical model of dynamical properties					Y	Y	Y		
Predict bounds on lattice properties								Y	
Writing up								xxxx	xxxx	xxxx

Has your supervisor completed a *Risk Assessment Form* about your project yet? ☐ Yes   ☒ No

Your signature: Zhen Rong Yap-

Date: 06/10/2023

Your supervisor's signature: Zachary Cordero, Michael Moody, Yuanbo Tang

Date: 05/10/2023, 11/10/2023, 16/10/2023

PROJECT MANAGEMENT FORM 2

1st Part II Project Analysis Form

Complete this form and send a copy to the Academic Administrative Office by Friday of 6th week of Michaelmas Term

Name: Zhen Rong Yap

Title of Project as given in your Project Description:
Additively Manufactured Interpenetrating Lattices with Tailorable Stiffness and Damping

Refer back to the project plan in your Project Description and list the goals you set for this term. Comment briefly on the extent to which you have achieved them.
Goals:
1) Assess damping behaviour of additively manufactured metal interpenetrating lattices (IPLs)
2) Relate stiffness and damping to geometrical (relative density, unit cell size, etc.) and surface roughness effects
3) Develop bounds on stiffness and damping of AM IPLs

Goal (1) has been stalled because of an issue with the metal printer. Goal (2) is in progress.

Identify clearly any difficulties you have encountered. Are they surmountable in the time available?
I was unable to work for the whole month of November because of illness. Extenuating circumstances documents to be submitted to Phillipa and the department.

There have been delays in printing lattices because I accidentally cross-threaded bolt holding the recoater blade. The printer should be fixed by January.

This term, I will focus on computational modelling the dynamical properties of the lattice (stiffness, damping).

State any refinements, modifications or replacements of the original objectives for your Part II project:
Modelling of the lattice stiffness and damping are to be carried out this term.

Are you intending to change the title of your project? If so, state the new title:
No

Have the training needs you identified in the Project Description been met, and have you identified any further training requirements?
Yes.

Tick the appropriate box. Do you have:

Results	None	_Some_	Sufficient
Analysis of results	None	_Some_	Sufficient

Do you have any other comments you wish to make?
Our collaborators in France have been notified about the delays.

After looking at the project plan in your Project Description complete the following project plan for the remainder of your Part II.

Task	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Print lattices				Y	Y				
Characterize structure of lattices + base				Y	Y				
Develop computational model of lattice dynamical properties (stiffness and damping)	Y	Y	Y	Y					
Characterize dynamical properties of lattices as functions of temperature, frequency, oscillation amplitude		Y	Y	Y					

Develop experimentally-validated analytical model of dynamical properties			Y	Y	Y				
Predict bounds on lattice properties						Y			
Writing up							xxxx	xxxx	xxxx

General comments by the supervisor:

Zhen's progress has been slow because of several factors – health issues, equipment delays, and climbing up the learning curve. He needs to develop the ability to solve problems independently, e.g., by reading papers, manuals, online forums, and requesting support and guidance from other people in the research group.
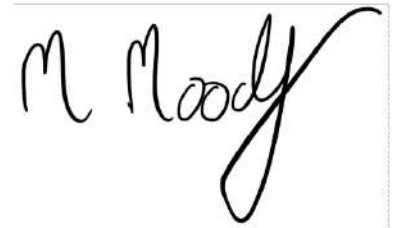
Your signature: *[signature]*
Date: 29/11/2023

Your supervisor's signature: *[signature]*
Date: 11.29.23

19th Nov 2023 *[signature: M Moody]*

PROJECT MANAGEMENT FORM 3

2nd Part II Project Analysis Form

Complete this form and send a copy to the Academic Administrative Office by Friday of 6th week of Hilary Term

Name: Zhen Rong Yap

Title of Project: Additively Manufactured Architectured Structures for Controlling Rotordynamics

Refer back to the project plan you made last term and list the goals you set for this term. Comment briefly on the extent to which you have achieved them.

Goals:
1) Assess damping behaviour of additively manufactured metal interpenetrating lattices (IPLs)
2) Relate stiffness and damping to geometrical (relative density, unit cell size, etc.) and surface roughness effects
3) Develop bounds on stiffness and damping of AM IPLs

Identify clearly any difficulties you have encountered. Are they surmountable in the time available?

N/A

State any refinements, modifications or replacements of the objectives you set for your Part II project:

Bearing supports have been printed that make use of sections that individually contribute to stiffness and damping. They have been tested on a rotordynamic rig.

Are you intending to change the title of your project?

☒ Yes. If so, state the new title: Additively Manufactured Architectured Structures for Controlling Rotordynamics

☒ No

What is the title of the talk you will give to the Department?

Additively Manufactured Architectured Structures for Controlling Rotordynamics

Have all your training needs for this project now been met?

Yes

Tick the appropriate box. Do you have:

Results	☐	None	☒	Some	☐	Sufficient
Analysis of results	☐	None	☒	Some	☐	Sufficient

Do you have any other comments you wish to make?

General comments by the supervisor:

Your signature:

Date: 13/ 3 | 2024

Your supervisor's signature:

Date: 3/13/2024

13/3/2024

Bibliography

[1] Jr. E. Logan. *Handbook of turbomachinery*. 2003.

[2] A. Schopphoff. 65 Years of Turbopumps: How the invention of the turbopump had a major impact on our modern life, 4 2023.

[3] D. R. Ballal and J. Zelina. Progress in Aeroengine Technology (1939-2003), 2004.

[4] R. J. Blackburn. Flight. pages 553–558, 1949.

[5] A. L. Kay. *German Jet Engine and Gas Turbine Development*. The Crowood Press, 2002.

[6] F. Whittle, M. I. E. Mech, and F. Bacon. The early history of the Whittle jet propulsion gas turbine. Technical report, Institution of Mechanical Engineers, 1946.

[7] G. Norris. CFM56: Engine of Change. *Flight International*, 1999.

[8] Embraer. EMB 120 Brasilia, 2011.

[9] K. J. Kloesel, N. A. Ratnayake, and C. M. Clark. A Technology Pathway for Airbreathing, Combined-Cycle, Horizontal Space Launch Through SR-71 Based Trajectory Modeling. Technical report.

[10] H. S. Seifert. Twenty-Five Years of Rocket Development. *Journal of Jet Propulsion*, 25(11):594–603, 11 1955.

[11] K. P. Van Hooser and D. P. Bradley. Space Shuttle Main Engine-The Relentless Pursuit of Improvement. Technical report.

[12] L. Lozano and S. Sabnis. 16.512 Rocket Propulsion Lecture Notes. 2023.

[13] C. J. Meisl. Rocket engine versus jet engine comparison. In *AIAA/ASME/SAE/ASEE 28th Joint Propulsion Conference and Exhibit, 1992*. American Institute of Aeronautics and Astronautics Inc, AIAA, 1992.

[14] D. W. Childs. Rotordynamic Characteristics of the High Pressure Oxygen Turbopump (HPOTP) in the SSME. Technical report, Turbomachinery Laboratories, Texas A&M University, 1984.

[15] C. Latimier. Numerical analysis of Raptor engine's combustion chamber. *HAL*, 2024.

[16] Jr A. L. Kimball. Interim Friction Theory of Shaft Whirling. *General Electric Review*, 27:244–251, 1924.

[17] B. L. Newkirk. Shaft Whipping. *General Electric Review*, 27:169–178, 1924.

[18] A. Preumont. *Solid Mechanics and Its Applications Twelve Lectures on Structural Dynamics*. 2013.

[19] J. W. Kannel and T. Merriman. Final Report on SSME Turbopump Bearing Analytical Study for NASA MSFC. Technical report, 1980.

[20] Lorna J. Gibson, Michael F. Ashby - Cellular Solids_ Structure and Properties (Cambridge Solid State Science Series) -Cambridge University Press (1999).

[21] S. Henshaw. Bed Adhesion and Squish (Accessed October 2023), 2018.

[22] X. Yu, J. Zhou, H. Liang, Z. Jiang, and L. Wu. Mechanical metamaterials associated with stiffness, rigidity and compressibility: A brief review, 5 2018.

[23] H. Jiang, H. Ziegler, Z. Zhang, S. Atre, and Y. Chen. Bending behavior of 3D printed mechanically robust tubular lattice metamaterials. *Additive Manufacturing*, 50, 2 2022.

[24] Y. Wang, Z. Chi, and J. Liu. On buckling behaviors of a typical bending-dominated periodic lattice. *Composite Structures*, 258, 2 2021.

[25] D. M. J. Dykstra, C. Lenting, A. Masurier, and C. Coulais. Buckling Metamaterials for Extreme Vibration Damping. *Advanced Materials*, 35(35), 9 2023.

[26] L. Van Belle, C. Claeys, E. Deckers, and W. Desmet. On the impact of damping on the dispersion curves of a locally resonant metamaterial: Modelling and experimental validation. *Journal of Sound and Vibration*, 409:1–23, 11 2017.

[27] A. P. Garland, K. M. Adstedt, Z. J. Casias, B. C. White, W. M. Mook, B. Kaehr, B. H. Jared, B. T. Lester, N. S. Leathe, E. Schwaller, and B. L. Boyce. Coulombic friction in metamaterials to dissipate mechanical energy. *Extreme Mechanics Letters*, 40, 10 2020.

[28] nTop Inc. nTop Documentation, Release 4.0, Implicit and Other Models, Implicit, 2024.

[29] B. C. White, A. Garland, and B. L. Boyce. Toughening by interpenetrating lattices. *Matter*, 6(2):570–582, 2 2023.

[30] S. V. Taylor, A. R. Moustafa, and Z. C. Cordero. Interpenetrating Lattices with Tailorable Energy Absorption in Tension. *Acta Materialia*, 216, 9 2021.

[31] S. Zhang, F. Yang, P. Li, Y. Bian, J. Zhao, and H. Fan. A topologically gradient body centered lattice design with enhanced stiffness and energy absorption properties. *Engineering Structures*, 263, 7 2022.

[32] P. R. Gradl and C. S. Protz. Technology advancements for channel wall nozzle manufacturing in liquid rocket engines. *Acta Astronautica*, 174:148–158, 9 2020.

[33] Solidworks. Topology Optimization (Accessed October 2023), 2024.

[34] Autodesk. Topology Optimization (Accessed October 2023), 2024.

[35] L. Cao, J. Li, J. Hu, H. Liu, Y. Wu, and Q. Zhou. Optimization of surface roughness and dimensional accuracy in LPBF additive manufacturing. *Optics and Laser Technology*, 142, 10 2021.

[36] Roberts et al. Hot Isostatic Pressing of Ultrasonic Additive Manufacturing Liquid Cold Plate Heat Exchangers. 2021.

[37] EOS M100 pricing. https://pick3dprinter.com/shop/eos-eos-m-100/, 2024.

[38] Y. Kai, S. Dhulipala, R. Sun, J. Lem, W. DeLima, T. Pezeril, and C. M. Portela. Dynamic diagnosis of metamaterials through laser-induced vibrational signatures. *Nature*, 623(7987):514–521, 11 2023.

[39] K. P. Menard and N. Menard. *Dynamic Mechanical Analysis*. CRC Press, 3rd edition edition, 2020.

[40] T. J. Beberniss and D. A. Ehrhardt. High-speed 3D digital image correlation vibration measurement: Recent advancements and noted limitations. *Mechanical Systems and Signal Processing*, 86:35–48, 3 2017.

[41] F. Ehrich. Observations of Subcritical Superharmonic and Chaotic Response in Rotordynamios. *Journal of Vibration and Acoustics*, 1992.

[42] R. Prayer Riju and S. Arulvel. A comparative analysis of compressive and wear failure of laser-powder bed fused Stainless Steel 316L produced with different hatch spacing in stripe and contour patterns. *Engineering Failure Analysis*, 162:108429, 8 2024.

[43] K. W. Lee, S. H. Lee, K. H. Noh, J. Y. Park, Y. J. Cho, and S. H. Kim. Theoretical and numerical analysis of the mechanical responses of BCC and FCC lattice structures. *Journal of Mechanical Science and Technology*, 33(5):2259–2266, 5 2019.

[44] F. Rosa, S. Manzoni, and R. Casati. Damping behavior of 316L lattice structures produced by Selective Laser Melting. *Materials and Design*, 160:1010–1018, 12 2018.

[45] Ziprecruiter. Ziprecruiter, 2024.

Economics

This short section provides an economic comparison between Laser Powder Bed Fusion (LPBF) and sand casting methods for the production of bearings. The analysis is grounded in actual production costs, labor rates, material expenses, and additional operational costs such as tooling and electricity.

$$C_p = \text{material cost} + \text{tooling cost} + \text{equipment cost} + \text{overhead cost} \tag{A.1}$$

$$C_p = \left(\frac{P_m \rho V}{1-f}\right) + \frac{C_T N_T}{n} + \left(\frac{1}{\dot{n}}\left[\frac{C_C}{Lt_{wo}}\right]\right) + \frac{\dot{C}_{oh}}{\dot{n}} \tag{A.2}$$
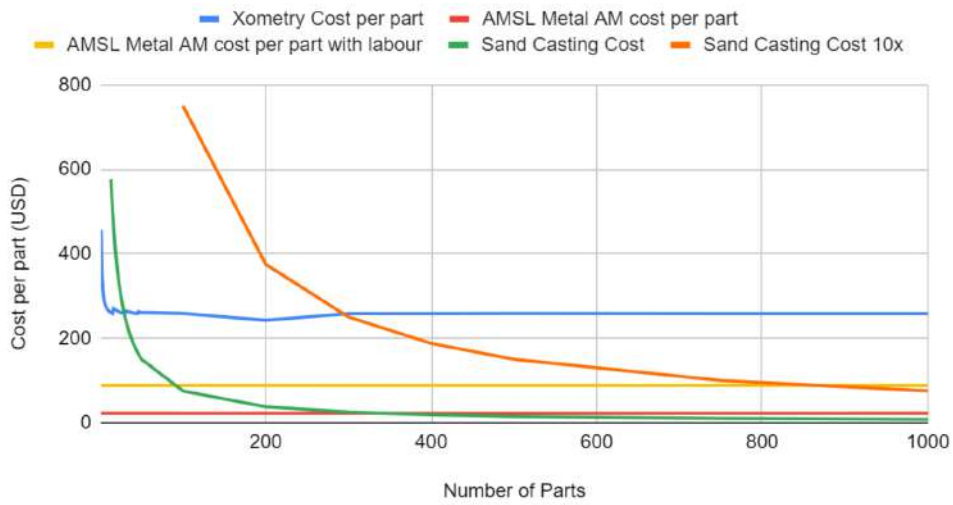
Figure A.1: Cost Curves.

Cost Factor	Cost per Bearing (Single)
Unit Manufacturing Cost	$45.50
Labor Cost (5 hours)	$69.70 - $173.10
Total Cost per Bearing	$115.20 - $218.60
Tooling Cost (initial)	$250,000
Tool Life	10 years
Production Rate per Hour	1/13
Electricity Cost per Hour	$0.27 (1 kW)
Argon Gas Cost per Print	$22.50
Material Cost per kg	$100
Material Cost per Component	$9.51
Scrap Fraction	Negligible
Break-even Point (estimated)	850 bearings

Table A.1: Economic Comparison of LPBF for Bearing Production

The unit cost for manufacturing bearings via LPBF is calculated to be $45.50, with labor costs ranging between $69.70 and $173.10 for five hours of work, based on the hourly wage in Massachusetts as reported by ZipRecruiter. [45] Including labor, the total cost per bearing is $173.62, or $86.81 each when two bearings are produced simultaneously.

The tooling cost (C_t) associated with LPBF is considerable, with an initial investment of $250,000 cite spread over a tool life (n_t) of ten years. The production rate per hour (n_p) is $\frac{1}{13}$, and the electricity cost $(C_{elec,})$ is approximately $0.27 per kWh, assuming a continuous 1 kW consumption per hour.

A significant expense in the LPBF process is the cost of inert argon gas, which can amount to 54% of the bearing production cost, particularly when considering that a $450 tank cite yields roughly 20 prints. However, larger build sizes can amortise this cost more effectively.

The 316 stainless steel powder required for LPBF costs $100 per kg cite, with each component weighing 95.13g, resulting in a material cost of $9.51 per component. The scrap fraction for unused material (f) in LPBF is negligible, which is a notable advantage over traditional manufacturing methods.

Cost Factor	Cost per Bearing (Single)
Unit Manufacturing Cost (material)	$0.36
Total Cost per Bearing	$0.36
Tooling Cost (initial)	High (variable)
Material Cost per kg	$3.75
Material Cost per Component	$0.36
Scrap Fraction	High
Break-even Point (estimated)	8,500+ bearings

Table A.2: Economic Comparison of Sand Casting for Bearing Production

Sand casting presents a different cost structure. While the per-unit material cost is significantly lower, at $0.36 per component, based on the bulk material cost of $3.75 per kg for 316 stainless steel, the overall costs including tooling and R&D can be substantial. The breakeven point for sand casting is around 850 bearings, but this number could realistically be an order of magnitude greater due to other tooling steps and the costs associated with research and development.

Economic Crossover and Considerations

Considering all expenses, the economic crossover point—where the total cost of LPBF equals that of sand casting—occurs at approximately 850 bearings. However, when additional factors such as tooling complexity and the R&D costs associated with sand casting are taken into account, the true breakeven point may be closer to 10 times the initial estimate, highlighting LPBF's potential cost-effectiveness for small to medium production runs.

Conclusion

The analysis suggests that LPBF can be economically viable for the production of bearings, especially in scenarios where small to medium quantities are required, and where the complexity or customization of parts justifies the higher initial costs. As production scales and complexity increases, the cost advantages of LPBF become more pronounced, particularly when considering its negligible scrap rates and the reduced material costs associated with 316 stainless steel.

B

Overhang

It was estimated that overhangs of larger than 2mm from printing the top plate was unfeasible. Two options were explored: Option 1 - Only vary strut thickness and height or Option 2 - Choose smaller unit cells. Increasing the unit cell size increases the unsupported overhang from the top plate. Larger unit cells that have lattice parameters of 3.333mm, 4mm and 5mm exceed our limit.

The image below is a cross section (top view) of the print to view the unit cells where a is the lattice parameter or unit cell size:

a = 3.333mm | Overhang = 2.4mm

a = 4mm | Overhang = 2.8mm

a = 5mm | Overhang = 3.5mm
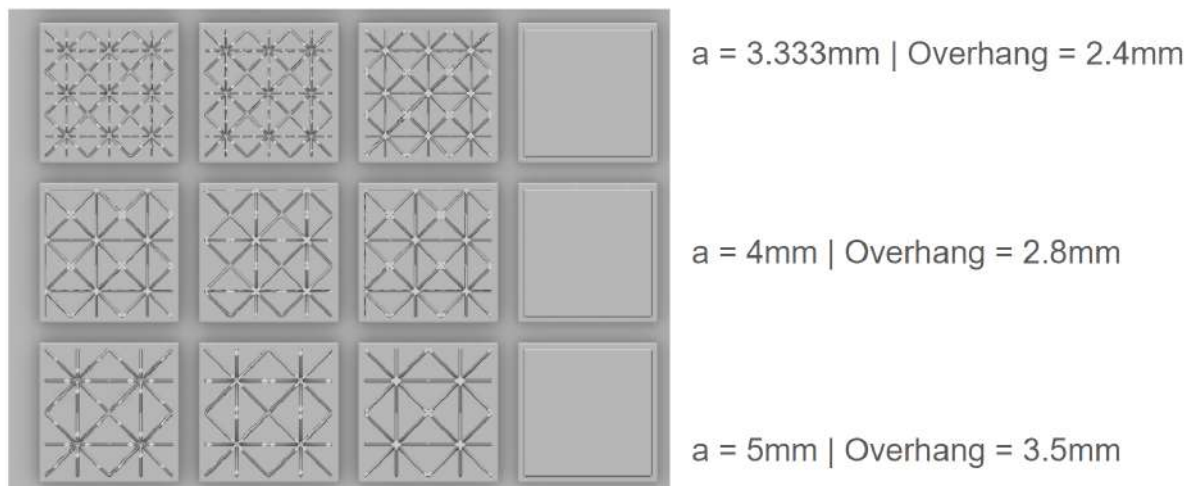
Figure B.1: Unit cells overhang.

Below is a cross section that is $\frac{3}{4}$ of the way up the largest unit cell (5mm) can decrease the overhang slightly but would still be too large than the estimated 2mm for which the

printer is capable. These unit cell sizes were chosen as they provide a range of 1% to 9% in relative density for the purpose of the experiment.
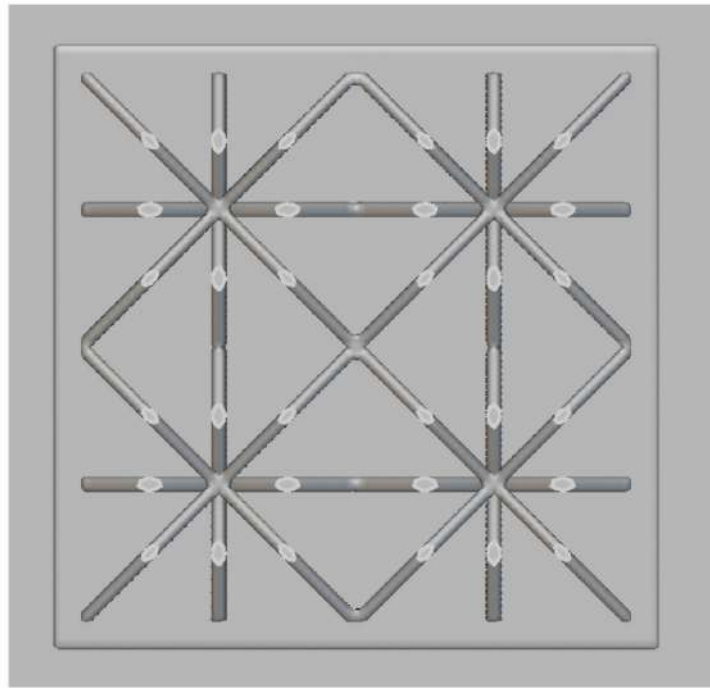
Figure B.2: Three Quarters Cross section view in Ntop.

Here is the table of relative densities for each strut thickness and unit cell size.

Strut Thickness / Unit Cell Size	2.0 mm	2.5 mm	3.0 mm	4.0 mm	5.0 mm
0.15 mm	0.096	0.039	0.019	0.006	0.002
0.2 mm	0.301	0.124	0.060	0.019	0.008
0.25 mm	0.729	0.301	0.146	0.046	0.019
0.29 mm	1.311	0.542	0.263	0.084	0.034
0.35 mm	2.748	1.142	0.555	0.177	0.073

Table B.1: Relative Density vs. Strut Thickness and Unit Cell Size

Option1: I had planned to use 0.25mm as the strut size but have included the other strut sizes here to show that we could also vary the strut size to change the relative density while keeping the unit cell size the same. This would help us get past the overhang issue and produce a similar range of relative densities. Look to the 2.5mm column to see a potential range of relative densities.

Here is the table of relative densities for each strut thickness and smaller unit cell sizes:

Strut Thickness / Unit Cell Size	1.2 mm	1.4 mm	1.7 mm	2.0 mm	2.5 mm
0.15 mm	0.620	0.365	0.198	0.096	0.039
0.2 mm	1.933	1.142	0.620	0.301	0.124
0.25 mm	4.633	2.748	1.499	0.729	0.301
0.29 mm	8.247	4.910	2.687	1.311	0.542
0.35 mm	16.985	10.179	5.603	2.748	1.142

Table B.2: Relative Density vs. Strut Thickness and Unit Cell Size

Option 2: We could decrease the size of the unit cells we print. So instead of 3.3333mm, 4mm and 5mm as the lattice parameters we could look at smaller ones. Look to the 0.25mm row to see the larger range in relative density from 0.34 to 0.05 with this option. Here is the updated table below. We considered other options that involve tapering the size of the struts to become larger as it gets to the top plate however that might affect the results of the stiffness and damping measurements.

Error for rotordynamics

Error propagaton

Given uncertainties σ_k in k and σ_m in m, the uncertainty in Ω_n (denoted as σ_{Ω_n}) can be estimated by:

$$\sigma_{\Omega_n}^2 = \left(\frac{\partial \Omega_n}{\partial k}\right)^2 \sigma_k^2 + \left(\frac{\partial \Omega_n}{\partial m}\right)^2 \sigma_m^2$$

The partial derivatives of Ω_n with respect to k and m are calculated as follows:

$$\frac{\partial \Omega_n}{\partial k} = \frac{1}{2}\left(\frac{k}{m}\right)^{-1/2}\frac{1}{m} = \frac{1}{2}\frac{\Omega_n}{k}$$

$$\frac{\partial \Omega_n}{\partial m} = -\frac{1}{2}\left(\frac{k}{m}\right)^{-1/2}\frac{k}{m^2} = -\frac{1}{2}\frac{\Omega_n}{m}$$

Substituting these into the error propagation formula, we get:

$$\sigma_{\Omega_n} = \sqrt{\left(\frac{1}{2}\frac{\Omega_n}{k}\right)^2 \sigma_k^2 + \left(-\frac{1}{2}\frac{\Omega_n}{m}\right)^2 \sigma_m^2}$$

$$\sigma_{\Omega_n} = \frac{\Omega_n}{2}\sqrt{\left(\frac{\sigma_k}{k}\right)^2 + \left(\frac{\sigma_m}{m}\right)^2}$$

Error from tachometer

Synchronisation: Comparing the x and y data indicates drift signals clock overtime as the displacement sensors were synchronised by an internal clock. Colin Pavan suggested

to use TTL pulses powered by the computer ports to send trigger pulses to both of sensors for synchronisation. A Microepsilon optoNCDT 1420s is suggested for future experiments as it utilises an alternate way of synchronisation that is more robust. As the current tachometer is only updating at 4Hz, a high frequency sampling would also improve the accuracy of the experiment.

Lattice measurements

Lattice	Unit Cell Size (mm)	Relative Density	Height (m)	Total Weight (g)	Bottom Plate	Structure Weight
A2	1.42	0.11	0.0105	2.55	11 x 11 x 0.61 mm ∼ 0.59 g	1.96
A3	1.42	0.11	0.0104	2.44	11 x 11 x 0.6 mm ∼ 0.58 g	1.86
A4	1.42	0.11	0.0118	2.58	11 x 11 x 0.7 mm ∼ 0.67 g	1.92
A5	1.42	0.11	0.0195	4.99	11 x 11 x 0.64 mm ∼ 0.62 g	4.26
B2	1.66	0.08	0.0108	2.15	11 x 11 x 0.56 mm ∼ 0.54 g	1.61
B3	1.66	0.08	0.0109	2.17	11 x 11 x 0.57 mm ∼ 0.55 g	1.62
B4	1.66	0.08	0.0107	2.18	11 x 11 x 0.55 mm ∼ 0.53 g	1.65
B5	1.66	0.08	0.0108	2.29	11 x 11 x 0.69 mm ∼ 0.66 g	1.63
C2	2.00	0.05	0.0100	0.97	11 x 11 x 0.16 mm ∼ 0.15 g	0.82
C3	2.00	0.05	0.0110	1.62	11 x 11 x 0.82 mm ∼ 0.79 g	0.83
C4	2.00	0.05	0.0112	1.71	11 x 11 x 0.87 mm ∼ 0.84 g	0.87
C5	2.00	0.05	0.0109	1.36	11 x 11 x 0.55 mm ∼ 0.53 g	0.83

Table D.1: Properties of Lattice Structures as measured by Ievgeniia Chaban

Additional LIRAS data for lattices from the back

Table E.1: Properties of lattices measured from the back

Lattice	Natural Frequency (kHz)	Unit Cell Size (mm)	Relative Density	Height (m)	E (GPa)	Speed of Sound (m/s)	Damping Coefficient
A2	37.2696	1.42	0.225	0.0105	1.10	783	3.91
A3	34.5776	1.42	0.225	0.0104	0.93	719	5.19
A4	34.2302	1.42	0.225	0.0118	1.17	808	4.52
A5	21.3312	1.42	0.225	0.0195	1.24	832	10.15
B2	27.3253	1.66	0.166	0.0108	0.46	590	5.77
B3	27.1693	1.66	0.166	0.0109	0.46	592	4.76
B4	21.9298	1.66	0.166	0.0107	0.29	469	4.56
B5	23.3191	1.66	0.166	0.0108	0.34	504	8.57
C2	17.1816	2.00	0.115	0.0100	0.11	344	7.29
C3	18.4149	2.00	0.115	0.0110	0.15	405	4.89
C4	19.0507	2.00	0.115	0.0112	0.17	427	6.92
C5	18.5070	2.00	0.115	0.0109	0.15	403	7.85

Code

Code

This appendix contains code for the sections pertaining to rotordynamics, sound analysis and LIRAS.

Rotordynamics code

Zhen Rong Yap (2024)

```python
[5]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import math
```

```python
[6]: import os
     from scipy.optimize import minimize
     import sys

     def find_all_paths(directory):
         paths = []
         for root, dirs, files in os.walk(directory):
             for name in files:
                 paths.append(name)
             for name in dirs:
                 paths.append(name)
         return paths

     class data_file:
         def __init__(self, directory_path, path):
             self.path = path
             self.directory_path = directory_path
             self.full_path = os.path.join(directory_path, path)
             self.plots_directory = os.path.join('Plots', directory_path)
             self.x_lim_min = 0
             self.x_lim_max = 3800
             self.y_lim_min = -1
             self.y_lim_max = 1
             self.show_plots = True  # Attribute to control plot display
             self.save_plots = False
             self.print_bool = False
             self.clean_bool = True

             # Constants for calculations
             self.M_disc = 0.832
             self.Br = 0
             self.L = 560/1000
             self.E = 2e11
```

```python
        self.r_shaft = 9.55/(2*1000)
        self.rho = 7900
        self.r_disc = 76.02/(2*1000)
        self.eccentricity = 0.25 # estimated from the data

        self.nu = 10 # damping
        self.ro = 38.5826 / 2000   # [mm]
        self.brass_OD = 15.875 / 2000   # [mm]
        self.ri = 9/2000
        self.A = 0.00955 * 0.019   # [m^2] Cross-sectional area of the bearing
        self.L_bearing = 0.019   # [m] bearing length
        self.hole_base = 5.3252/100
        self.block = self.hole_base-self.ro/1000
        self.E_al = 70e9
        self.E_brass = 90e9
        self.brass_thickness = self.brass_OD-self.ri
        self.normalise_rpm_by_max_displacement = True
        self.normalise_eccentricity = True

        if 'additive' in self.path.lower():
            self.E_only = 190e9
            self.d = 19E-3
            self.E = self.block*self.E_al+self.brass_thickness*self.E_brass+self.
→E_only
        elif 'aluminium' in self.path.lower():
            # self.E = 90e9*0.1+70e9*0.9   # Aluminium
            self.E_only = 90e9   # Aluminium
            self.d = 27E-3
            self.E = self.block*self.E_al+self.brass_thickness*self.E_brass+self.
→E_only
        elif 'plastic' in self.path.lower():
            # self.E = 10e9*0.1+70e9*0.9   # Plastic
            self.E_only = 90e9   # Plastic
            self.d = 15E-3
            self.E = self.block*self.E_al+self.brass_thickness*self.E_brass+self.
→E_only
        elif 'ball_bearing' in self.path.lower():
            self.E = 220e9   # High carbon chromium bearing steel
            self.d = 27E-3
        # Constants for flexible support calculations
        print('E:', self.E)

        # Calculations
        self.calculate_dynamics()   # Existing calculations
        self.calc_stiffness()
        self.jeffcott_rotor_model_flexible()
```

```python
    def read(self):
        if self.full_path.endswith('.csv'):
            self.data = pd.read_csv(self.full_path)
            # print('Read as .csv')
        elif self.full_path.endswith('.xlsx'):
            self.data = pd.read_excel(self.full_path)
            # print('Read as .xlsx')
        else:
            print('Provide file as .csv or .xlsx (excel format). ', self.path,'␣
↪is not a readable format.')

    def calculate_dynamics(self):
        self.M_shaft = math.pi * self.r_shaft**2 * self.L * self.rho
        self.I_z_disc_A = math.pi * (self.r_disc**4 - self.r_shaft**4) / 4
        self.I_z_shaft_A = math.pi * (self.r_shaft**4) / 4
        self.Kr = 2* 48 * self.E * (self.I_z_shaft_A) / self.L**3
        self.Wn = math.sqrt(self.Kr / self.M_disc)
        self.RPM_n = self.Wn * 60 / (2 * math.pi)
        self.zeta = self.Br / (2 * self.M_disc * self.Wn)
        self.W = np.arange(0, 5000, 10)
        self.rb_div_e = (self.W/self.Wn)**2 / np.sqrt((1 - (self.W/self.
↪Wn)**2)**2 + 2 * self.zeta * (self.W/self.Wn)**2)

    def print_rigid_dynamics(self):
        print('For the rigid bearing supports')
        print('Wn: ', self.Wn)
        print('RPM:', self.Wn * 60 / (2 * math.pi))

    def calc_stiffness(self):
        h = 0.25/1000
        L = 10/1000
        OD = 38.5826/(2*1000)
        ID = 15.875/(2*1000)
        I = L*h**3/12
        self.E_f = 48*self.E*I*2*(OD-self.ri*2)/(4*math.pi*(self.d)*L**3*self.
↪ri*2)
        A = 2*math.pi*ID/2*self.d # 2 bearing supports
        self.Ks = A*self.E_f/self.d

    def jeffcott_rotor_model_flexible(self):
        self.Wc = np.sqrt(self.Ks / self.M_disc)
        self.RPM_c = self.Wc * 60 / (2 * math.pi)
        self.kappa = self.Ks / self.Kr
        self.c1 = 1 - (1 + self.kappa) * (self.W / self.Wc) ** 2
        self.c2 = 1 - (self.W / self.Wc) ** 2 * self.kappa
        self.c3 = 2 * self.W / self.Wc * self.nu
```

```python
        self.ra_div_e = (self.W / self.Wc) ** 2 / np.sqrt(self.c1 ** 2 + (self.
 →c3 * self.c2) ** 2)
        self.rb_div_e = self.ra_div_e * np.sqrt((1 + self.kappa) ** 2 + (self.c3
 →* self.kappa) ** 2)

    def print_flexible_dynamics(self):
        print('For the flexible bearing supports')
        print(f'Wc: {self.Wc}, RPM: {self.RPM_c}')
        print(f'Wn: {self.Wn}, RPM: {self.RPM_n}')

    def plot_response_flexible(self, axis, fig_num, xlabel, ylabel, xlimits=[0,
 →5], ylimits=[0, 15]):
        if self.clean_bool:
            plt.figure(fig_num)
            # plt.plot(self.W/self.Wc, self.rb_div_e)
            plt.plot(self.W/(self.rpm_factor*(2*math.pi)/60), self.rb_div_e,
 →label=r'$\nu = {:.2f}$'.format(self.nu))
            # plt.plot(self.W/(2*math.pi)*60, self.rb_div_e)
            title_suffix = 'r'
            if self.normalise_rpm_by_max_displacement:
                normalised = 'normalised_by_RPM'
            else:
                normalised = ''
            plt.title(f'Flexible Rotor radial displacement against RPM: ' +
 →(self.path).split('.')[0])
            plt.xlabel(xlabel)
            plt.ylabel(ylabel)
            plt.legend()
            # plt.xlim(xlimits)
            # plt.ylim(ylimits)
            if self.save_plots:
                plt.savefig(os.path.join(self.plots_directory, f"{self.path.
 →split('.')[0]}_{title_suffix}_flexible_response_{normalised}.png"))
        plt.show()

    def plot_r_clean_scatter_max(self, axis='r'):

        filtered_data = self.data[(self.data[self.data.columns[0]].between(-100,
 →100)) &
                         (self.data[self.data.columns[1]].between(-100, 100))]

        # Replace the original DataFrame with the filtered DataFrame
        self.data = filtered_data

        # Optionally, reset the index if you want a clean, consecutive index
 →after filtering
```

```python
        self.data.reset_index(drop=True, inplace=True)

        if self.print_bool:
            print("Max value before squaring:", np.max(self.data[self.data.
↪columns[0]]))
            print("Min value before squaring:", np.min(self.data[self.data.
↪columns[0]]))
            print("Max value after squaring:", np.max(np.square(self.data[self.
↪data.columns[0]])))
            print("Min value after squaring:", np.min(np.square(self.data[self.
↪data.columns[0]])))

            # Also, directly check for any infinities or NaNs
            print("Infinities in data:", np.isinf(self.data[self.data.
↪columns[0]]).any())
            print("NaNs in data:", np.isnan(self.data[self.data.columns[0]]).
↪any())

        """Plots a clean scatter diagram showing max for either x or y␣
↪displacement, with optional normalizations."""
        self.data['r'] = (np.sqrt(np.square(np.abs(self.data[self.data.
↪columns[0]]))+np.square(np.abs(self.data[self.data.columns[1]]))))
        # Normalize by eccentricity if required
        if self.normalise_eccentricity:
            factor = self.eccentricity
            # print(factor)
            ylabel = f'displacement {axis} (normalized by eccentricity)'
        else:
            factor = 1
            ylabel = f'displacement {axis}'

        # Normalize RPM by the index at max displacement if required
        if self.normalise_rpm_by_max_displacement:
            # Filter 'r' where 'rpm' is between 800 and 2000
            # Assume 'rpm' is the column name; replace with self.data.
↪columns[index] if it's not named
            condition = (self.data[self.data.columns[2]] >= 800) & (self.
↪data[self.data.columns[2]] <= 2000)
            filtered_r = self.data.loc[condition, 'r']

            # Check if there are any entries after filtering
            if not filtered_r.empty:
                # Find the index of the max value in 'r' within the filtered␣
↪range
                self.rpm_factor = self.data[self.data.columns[2]][filtered_r.
↪idxmax()]
```

```python
        else:
            # Handle case where no data falls within the range
            self.rpm_factor = 1  # Default or error handling
    else:
        self.rpm_factor = 1

    print('Crtical RPM: ', self.rpm_factor)

    self.data = self.data.dropna()
    grouped_data = self.data.groupby(self.data.columns[2])
    max_min_data = grouped_data['r'].agg(['max', 'min']).
→interpolate(method='linear')

    if self.clean_bool:
        plt.scatter(max_min_data.index / self.rpm_factor,␣
→max_min_data['max'] / factor, color='red', label='r', s=0.1)
        plt.plot(max_min_data.index / self.rpm_factor, max_min_data['max'] /␣
→factor, color='red', linewidth=1)
    else:
        plt.scatter(self.data[self.data.columns[2]] / self.rpm_factor, self.
→data['r'] / factor, color='red', label='r', s=1)
        # plt.plot(self.data[self.data.columns[2]] / self.rpm_factor, self.
→data['r'] / factor, color='red', linewidth=1)

    plt.xlabel('Normalized RPM' if self.normalise_rpm_by_max_displacement␣
→else 'RPM')
    plt.ylabel(ylabel)
    title_suffix = 'r'
    plt.title(f'Flexible Rotor radial displacement against RPM: ' + (self.
→path).split('.')[0])
    if self.clean_bool:
        plt.xlim([self.x_lim_min / self.rpm_factor, self.x_lim_max / self.
→rpm_factor])
        plt.ylim([0, 15])
    else:
        plt.ylim([0,10])
    plt.legend()
    if self.normalise_rpm_by_max_displacement:
        normalised = 'normalised_by_RPM'
    else:
        normalised = ''
    if self.show_plots:
        if self.clean_bool:
            plt.ylim([0,15])
        else:
            plt.ylim([0,10])
```

```
        plt.show()
    if self.save_plots:
        plt.savefig(os.path.join(self.plots_directory, f"{self.path.split('.
→')[0]}_{title_suffix}_flexible_response_{normalised}.png"))


def print_bs(self):
    print('Bs is: ', self.nu*2*self.M_disc*self.Wc)
```

```
[7]: def run_all(material_index, directory_path_list, nu_list, eccentricity_list,␣
     →show_plots_bool=False, save_plots_bool=True, norm_bool=True,␣
     →max_data_bool=True):
         data_object_list = []

         all_paths = find_all_paths(directory_path_list[material_index])
         # print("All paths in the folder:")

         for path in all_paths:
             data_object_list.append(data_file(directory_path_list[material_index],␣
     →path))

         # for data_object in data_object_list:
         #     print(data_object.path)

         def data_print(data_object_list, index):
             data_object = data_object_list[index]
             data_object.read()
             data_object.show_plots = True
             # data_object.plot_x_clean_scatter_eccentricity()
             data_object.plot_x_clean_scatter_eccentricity()
             # data_object.plot_y_clean_scatter()

         def data_operate(data_object_list, axis, index, nu_list, eccentricity_list):
             data_object = data_object_list[index]
             data_object.read()

             if axis == 'r':
                 data_object.nu = nu_list
                 data_object.eccentricity = eccentricity_list

             data_object.calculate_dynamics()
             data_object.calc_stiffness()
             data_object.jeffcott_rotor_model_flexible()
             data_object.show_plots = show_plots_bool
             data_object.save_plots = save_plots_bool
             data_object.clean_bool = max_data_bool
             # data_object.normalise_eccentricity = norm_bool
             data_object.normalise_rpm_by_max_displacement = norm_bool
```

```python
        data_object.plot_r_clean_scatter_max(axis=axis)
        # data_object.plot_r_clean_scatter()
        data_object.plot_response_flexible(axis=axis, fig_num=1, xlabel='Speed␣
 ↪Ratio w', ylabel='Whirl Amplification Factor r/E')

        print('Nu:', nu_list)
        print('Eccentricity:', eccentricity_list)

        data_object.jeffcott_rotor_model_flexible()
        data_object.print_flexible_dynamics()
        data_object.print_bs()

    for i in range(len(all_paths)):
        data_operate(data_object_list, 'r', i, nu_list[material_index],␣
 ↪eccentricity_list[material_index])

class RedirectStdoutToFile:
    def __init__(self, file_path):
        self.file_path = file_path

    def __enter__(self):
        self.old_stdout = sys.stdout
        sys.stdout = open(self.file_path, 'w')
        return self

    def __exit__(self, exc_type, exc_value, traceback):
        sys.stdout.close()
        sys.stdout = self.old_stdout
```

```python
directory_path_list = ['Additive', 'Aluminium', 'Plastic', 'Bearing']
nu_list = [0.3, 0.11, 0.15, 0.25]
eccentricity_list = [0.2, 0.1, 0.1, 0.19]

file_path = 'output.txt'  # The path for the output file
with RedirectStdoutToFile(file_path):
    for i in range(len(directory_path_list)):
        #replace this here with a function that opens a .txt files clear it all␣
 ↪then writes all outputs
        print(directory_path_list[i])
        run_all(i, directory_path_list, nu_list, eccentricity_list,␣
 ↪show_plots_bool=False, save_plots_bool=True, norm_bool=True,␣
 ↪max_data_bool=True)
print(f"Output for iteration {i} has been written to {file_path}")
```

Sound analysis Code

Zhen Rong Yap (2024)

```
[ ]: pip install pydub
```

```python
[ ]: import os
     import numpy as np
     from pydub import AudioSegment
     from scipy.fft import fft, fftfreq
     import matplotlib.pyplot as plt

     def read_mp3(file_path, duration=None):
         """Read an MP3 file and convert it to a numpy array of samples."""
         audio = AudioSegment.from_mp3(file_path)
         if duration:
             audio = audio[5:duration * 1000]
         # Convert stereo to mono by averaging left and right channels
         if audio.channels == 2:
             audio = audio.set_channels(1)
         samples = np.array(audio.get_array_of_samples())
         return samples, audio.frame_rate

     def perform_fft(samples, sample_rate):
         """Perform FFT on samples and return frequencies and their amplitudes."""
         n = len(samples)
         yf = fft(samples)
         xf = fftfreq(n, 1 / sample_rate)
         return xf[:n//2], np.abs(yf[:n//2])

     def plot_spectrum(frequencies, amplitudes, title, plot_path):
         """Plot the amplitude spectrum of the frequencies."""
         plt.figure(figsize=(10, 5))
         plt.plot(frequencies, amplitudes)
         plt.title(title)
         plt.xlabel('Frequency (Hz)')
         plt.ylabel('Amplitude')
         plt.grid(True)
         plt.xlim(0, 2000)
         plt.ylim(0, 3E8)
         plt.savefig(plot_path)
         plt.close()
```

```python
        print(f"Plot saved as {plot_path}")

def analyze_directory(directory, plot_directory, initial_seconds=10):
    """Analyze all MP3 files in the directory and generate plots."""
    # Create plot directory if it doesn't exist
    os.makedirs(plot_directory, exist_ok=True)
    for file in os.listdir(directory):
        if file.endswith('.mp3'):
            file_path = os.path.join(directory, file)
            print(f"Analyzing {file}...")
            full_samples, sample_rate = read_mp3(file_path)
            full_freqs, full_amplitudes = perform_fft(full_samples, sample_rate)

            # Read initial segment
            init_samples, _ = read_mp3(file_path, duration=initial_seconds)
            init_freqs, init_amplitudes = perform_fft(init_samples, sample_rate)
            # Compute difference in amplitudes and adjust frequencies
            min_length = min(len(full_amplitudes), len(init_amplitudes))
            diff_amplitudes = full_amplitudes[:min_length] - init_amplitudes[:
↪min_length]
            adjusted_freqs = full_freqs[:min_length]  # Adjust frequency array␣
↪to match amplitude length

            base_filename = os.path.splitext(os.path.basename(file_path))[0]
            plot_spectrum(full_freqs, full_amplitudes, f'Full Spectrum of␣
↪{base_filename}', os.path.join(plot_directory, f"{base_filename}_full_spectrum.
↪png"))
            plot_spectrum(init_freqs, init_amplitudes, f'Initial␣
↪{initial_seconds} Seconds Spectrum of {base_filename}', os.path.
↪join(plot_directory, f"{base_filename}_initial_spectrum.png"))
            plot_spectrum(adjusted_freqs, diff_amplitudes, f'Difference Spectrum␣
↪of {base_filename}', os.path.join(plot_directory,␣
↪f"{base_filename}_diff_spectrum.png"))

# Define directories
sound_directory = 'Sound'
plot_directory = 'Plots/Sound_full'

# Run the analysis on the 'Sound' directory
analyze_directory(sound_directory, plot_directory)
```

IPL Calculations

Zhen Rong Yap (2024)

```
[ ]:   # Calculating Young's modulus from Abaqus
       force = 200
       a = 5E-3
       Area = a**2
       Stress = force/Area
       Max_strain = 0.1298
       E = Stress/Max_strain
       print("{:e}Pa".format(E))
       print(str(E/1E9)+' GPa')
```

```
[85]:  #IPL struts
       import math
       import numpy as np

       def bcc_strut_density(a, h, t):
           """
           Calculate the density of a BCC unit cell with strut thickness t, accounting␣
       ↪for overlap.

           :param a: lattice parameter, or the edge length of the BCC unitK cell.
           :param t: thickness of the struts.
           :return: density of the unit cell struts.
           """
           # Each space diagonal can be calculated as sqrt(3) * a.
           length_strut = math.sqrt(a**2+a**2+h**2)

           # Calculate the volume of one strut.
           volume_strut = math.pi * (t / 2)**2 * length_strut  # volume = pi * r^2 * h

           # Calculate the overlapped volume at the center of the unit cell.
           overlap_volume = math.pi * (t / 2)**2 * t

           # Calculate the total volume of struts in the unit cell minus the overlapped␣
       ↪volume.
           total_volume_struts = 4 * volume_strut - overlap_volume

           # Calculate the volume of the unit cell.
           volume_unit_cell = a**3
```

```python
        # Calculate the density of the struts in the unit cell.
        density = total_volume_struts / volume_unit_cell

        return density
```

```python
# Getting strut thickness from relative density required
import math

def bcc_strut_thickness(a, relative_density):
    """
    Calculate the strut thickness given relative density of a BCC unit cell.

    :param a: lattice parameter, or the edge length of the BCC unit cell.
    :param relative_density: relative density of the unit cell struts.
    :return: strut thickness.
    """

    # Calculate the length of one strut (space diagonal).
    length_strut = math.sqrt(3) * a

    # Calculate the volume of the unit cell.
    volume_unit_cell = a ** 3

    # Calculate the total volume of struts in the unit cell.
    total_volume_struts = relative_density * volume_unit_cell

    # Solve for t
    t = 2 * math.sqrt((total_volume_struts) / (4 * math.pi * length_strut))

    return t

# Example usage:
a = 2.5  # Suppose the lattice parameter a is 1
relative_density = 0.034825  # Suppose the relative density is 0.1
t = bcc_strut_thickness(a, relative_density)
t
```

```python
# FCC struts
import math

def fcc_strut_density(a, t):
    """
    Calculate the density of an FCC unit cell with strut thickness t.

    :param a: lattice parameter, or the edge length of the FCC unit cell.
    :param t: thickness of the struts.
```

```
        :return: density of the unit cell struts.
        """

        # The FCC unit cell contains 12 equal edge struts.
        # Each strut can be considered as a cylinder.

        # Calculate the length of one strut (which is equal to the lattice␣
    ↪parameter).
        length_strut = a

        # Calculate the volume of one strut.
        volume_strut = math.pi * (t / 2)**2 * length_strut  # volume = pi * r^2 * h

        # Calculate the total volume of struts in the unit cell.
        total_volume_struts = 12 * volume_strut

        # Calculate the volume of the unit cell.
        volume_unit_cell = a**3

        # Calculate the density of the struts in the unit cell.
        density = total_volume_struts / volume_unit_cell

        return density

# Example usage:
a = 1  # Suppose the lattice parameter a is 1
t = 0.1  # Suppose the strut thickness t is 0.1
density = fcc_strut_density(a, t)
print(f"Density of the FCC unit cell struts: {density:.6f}")
```

```
[88]: # Stiffness from Paper (Theoretical and numerical analysis of the mechanical␣
      ↪responses of BCC and FCC lattice structures)
      import math
      def stiffness_equation(a, b, d, r, Eo):
        stiffness = 4*a*Eo/(b*d*((a**2)/(math.pi*r**2*math.
      ↪sqrt(a**2+b**2+d**2))+((b**2+d**2)*math.sqrt(a**2+b**2+d**2)/(12*math.
      ↪pi*r**4))))
        return stiffness
```

```
[ ]: # Calculating density, modulus, speed of sound
     import numpy as np
     import pandas as pd

     # unit_cell_size = [1.42857, 1.66666, 2.0, 2.5, 3.333, 4, 5]
     # strut_radius = [0.075, 0.1, 0.125, 0.145, 0.175]

     unit_cell_size = [1.42857, 1.66666, 2.0, 2.5]
```

```python
strut_radius = [0.1]

Eo = 210E9  # Example value for Eo
rho_316 =  7980
n=3.8


# The columns will be 'Strut Thickness (2x) mm'
index = ['{:.3g} mm'.format(2*r) for r in strut_radius]
# Formatting the index to include 'mm' for unit cell size
columns = ['{:.1f} mm'.format(a) for a in unit_cell_size]

# Initialize an empty DataFrame with specified columns and index
df = pd.DataFrame(index=index, columns=columns)
density_df = pd.DataFrame(index=index, columns=columns)
E_df = pd.DataFrame(index=index, columns=columns)
speed_sound_df = pd.DataFrame(index=index, columns=columns)

# Fill the DataFrame with stiffness values
for a in unit_cell_size:
    for r in strut_radius:
        E_beam_bending_1 = stiffness_equation(a, a, a, r, Eo)
        E_beam_bending_2 = stiffness_equation(a, a/math.sqrt(2), a/math.sqrt(2),␣
 ↪r, Eo)
        E_beam_bending = E_beam_bending_1 + E_beam_bending_2
        # Accessing cells by the formatted string of the unit cell size and␣
 ↪strut thickness
        df.loc['{:.3g} mm'.format(2*r), '{:.1f} mm'.format(a)] = '{:.3f}'.
 ↪format(E_beam_bending/1E9)

        density_1 = bcc_strut_density(a, a, 2*r)
        density_2 = 0.5*bcc_strut_density(a/math.sqrt(2), a, 2*r)
        density = density_1 + density_2
        density_df.loc['{:.3g} mm'.format(2*r), '{:.1f} mm'.format(a)] = '{:.
 ↪3f}'.format(density)
        E_df.loc['{:.3g} mm'.format(2*r), '{:.1f} mm'.format(a)] = '{:.3f}'.
 ↪format(density**3*Eo/1E9)
        speed_sound_df.loc['{:.3g} mm'.format(2*r), '{:.1f} mm'.format(a)] = '{:.
 ↪3f}'.format(math.sqrt(2*Eo*density**n/(density*rho_316)))


# Apply formatting to avoid standard scientific notation
df = df.applymap(lambda x: '{:.2f}'.format(float(x)))
density_df = density_df.applymap(lambda x: '{:.3f}'.format(float(x)))
E_df = E_df.applymap(lambda x: '{:.3f}'.format(float(x)))
speed_sound_df = speed_sound_df.applymap(lambda x: '{:.2f}'.format(float(x)))
```

```
print('Relative Density df (Strut thickness (row), Unit Cell Size (column))')
print(density_df)
print(' ')
print('E effective (GPa) from Cellular Solids')
print(E_df)
print(' ')
print('E effective (GPa) from Beam Bending')
print(df)
print(' ')
print('Sound speed df')
print(speed_sound_df)
```

```
# Modulus varying with strut thickness and unit cell size

# Fill the DataFrame with stiffness values
m_list = np.linspace(2.65, 2.8, 16)
for m in m_list:
  for a in unit_cell_size:
      for r in strut_radius:
        E_beam_bending_1 = stiffness_equation(a, a, a, r, Eo)
        E_beam_bending_2 = stiffness_equation(a, a/math.sqrt(2), a/math.sqrt(2),
  ↪r, Eo)
        E_beam_bending = E_beam_bending_1 + E_beam_bending_2
        # Accessing cells by the formatted string of the unit cell size and
  ↪strut thickness
        df.loc['{:.3g} mm'.format(2*r), '{:.1f} mm'.format(a)] = '{:.3f}'.
  ↪format(E_beam_bending/1E9)

        density_1 = bcc_strut_density(a, a, 2*r)
        density_2 = 0.5*bcc_strut_density(a/math.sqrt(2), a, 2*r)
        density = density_1 + density_2
        density_df.loc['{:.3g} mm'.format(2*r), '{:.1f} mm'.format(a)] = '{:.
  ↪3f}'.format(density)
        E_df.loc['{:.3g} mm'.format(2*r), '{:.1f} mm'.format(a)] = '{:.3f}'.
  ↪format(density**3*Eo/1E9)
        speed_sound_df.loc['{:.3g} mm'.format(2*r), '{:.1f} mm'.format(a)] = '{:.
  ↪3f}'.format(math.sqrt(2*Eo*density**n/(density*rho_316)))

  E_df = E_df.applymap(lambda x: '{:.2f}'.format(float(x)))
  print('E effective from Cellular Solids (GPa) and n:', m)
  print(E_df)
  print('')
```

Compression Test Analysis Code

Zhen Rong Yap (2024)

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

path = 'Compression Data 14.05.2024.xlsx'
df = pd.read_excel(path)
print(df.head)
```

```python
from scipy.signal import find_peaks

# Calculate the gradient between each point
stress = 'Stress (MPa)'
strain = 'Strain'
df['Gradient'] = np.gradient(df[stress], df[strain])
interval = 0.0025


def plot(unit_cell_size, rho, df, peaks):
    # Plot the data
    plt.figure(figsize=(12, 8))
    plt.plot(df['Strain'], df['Stress (MPa)'], label='Stress-Strain Curve')

    # Create a list of (peak index, peak strain, peak stress) tuples
    peak_list = [(i, df['Strain'][i], df['Stress (MPa)'][i]) for i in peaks]
    # Sort peaks by strain for easier processing
    peak_list.sort(key=lambda x: x[1])

    # Group peaks that are within 0.05 of each other and select the highest␣
↪within each group
    selected_peaks = []
    current_group = []

    for peak in peak_list:
        if not current_group or peak[1] - current_group[-1][1] <= 0.05:
            current_group.append(peak)
        else:
            # Select the peak with the maximum stress from the current group
            highest_peak = max(current_group, key=lambda x: x[2])
```

```python
            selected_peaks.append(highest_peak)
            current_group = [peak]

    # Don't forget to check the last group
    if current_group:
        highest_peak = max(current_group, key=lambda x: x[2])
        selected_peaks.append(highest_peak)

    # Annotate the selected peaks
    for peak in selected_peaks:
        peak_index, peak_strain, peak_stress = peak
        plt.plot(peak_strain, peak_stress, 'x', label=f'Peak Stress:␣
↪{peak_stress:.6f}')
        plt.annotate(f'{peak_stress:.2f}',
                     (peak_strain, peak_stress),
                     textcoords="offset points",
                     xytext=(0, 10),
                     ha='center')

    plt.xlabel('Strain')
    plt.ylabel('Stress (MPa)')
    plt.title('Stress-Strain Curve with Stress of Peaks Annotated for '+'Unit␣
↪Cell Size: '+str(unit_cell_size)+'mm')
    plt.grid(True)
    plt.savefig('S-S-curve-peaks-annotated_'+str(unit_cell_size)+'mm.png')
    plt.show()

peaks, _ = find_peaks(df['Stress (MPa)'])
unit_cell_size_list = [1.428, 1.666, 2.0, 2.5]
rho_list = [0.107, 0.078, 0.054, 0.054]
n = 2.69

for i in range(len(unit_cell_size_list)):
    new_df = df.copy(deep=True)
    new_df['Stress (MPa)'] = new_df['Stress (MPa)'].values*rho_list[i]**n/
↪rho_list[-1]**n
    # df['Strain']
    plot(unit_cell_size_list[i], rho_list[i], new_df, peaks)
```

LIRAS code

Zhen Rong Yap (2024)

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.signal import find_peaks
from scipy.fft import fft
from scipy.optimize import curve_fit
import os

# Close all existing figures
plt.close('all')

which_bool = 'Front'
# Create necessary directories
plots_dir = which_bool + ' Plots'
data_dir = which_bool + ' Data'
if not os.path.exists(plots_dir):
    os.makedirs(plots_dir)

# List CSV files in the data directory
file_list = [f for f in os.listdir(data_dir) if f.endswith('.csv')]
m = len(file_list)

# Color mapping for plotting
colors = plt.cm.hsv(np.linspace(0, 1, m))

# Initialize DataFrame to store peak results
peak_results = pd.DataFrame(columns=['File', 'Natural Frequency (kHz)',
 'Amplitude', 'HWHM', 'Damping Coefficient'])

# Define the Lorentzian function without baseline parameter
def lorentzian(x, A, x0, gamma):
    return (A / np.pi) * (gamma / ((x - x0)**2 + gamma**2))

# Process each file
for ii in range(m):
    filename = file_list[ii]
    data = pd.read_csv(os.path.join(data_dir, filename))
    t = data.iloc[1200:, 0].to_numpy() * 1e3  # Time in seconds
```

```python
    s = data.iloc[1200:, 1].to_numpy()  # Signal

    # Compute time increment and frequency range
    N = len(t)
    dt = np.mean(np.diff(t))
    freqs = np.fft.fftfreq(N, dt)[:N//2]  # Frequency in kHz

    # FFT computation and magnitude calculation
    FFT_result = fft(s)
    FFT_result_half = np.abs(FFT_result[:N//2]) * 2 / N

    # Determine frequency range based on file name
    if 'A' in filename:
        min_freq_threshold = 20   # kHz
        max_freq_threshold = 37.5  # kHz
    elif 'B' in filename:
        min_freq_threshold = 20   # kHz
        max_freq_threshold = 30   # kHz
    elif 'C' in filename:
        min_freq_threshold = 15   # kHz
        max_freq_threshold = 20   # kHz
    else:
        min_freq_threshold = 10   # Default minimum if unspecified
        max_freq_threshold = 39   # Default maximum if unspecified

    # Apply frequency limits
    valid_freqs = (freqs > min_freq_threshold) & (freqs < max_freq_threshold)
    freqs_limited = freqs[valid_freqs]
    FFT_result_half_limited = FFT_result_half[valid_freqs]

    if not freqs_limited.size:  # Check if the filtered frequencies array is
→empty
        print(f"No valid frequencies found in the range for file {filename}")
        continue

    # Determine a fixed baseline
    baseline = np.min(FFT_result_half_limited)

    # Find the natural frequency
    natural_freq_index = np.argmax(FFT_result_half_limited)
    natural_frequency = freqs_limited[natural_freq_index]

    plt.figure(figsize=(10, 6))
    plt.plot(freqs, FFT_result_half, color=colors[ii], marker='o', markersize=1)
    plt.scatter(natural_frequency, FFT_result_half_limited[natural_freq_index],
→color='red', label='Natural Frequency: {:.2f} kHz'.format(natural_frequency))
    plt.axhline(y=baseline, color='gray', linestyle='--', label='Baseline')
```

```python
    plt.xlabel('Frequency (kHz)')
    plt.ylabel('Amplitude')
    title = 'FFT-' + filename.replace('.csv', '')
    plt.title(title)

    # Find peaks with stringent criteria within the valid frequency range
    peaks, properties = find_peaks(FFT_result_half_limited, height=np.
→max(FFT_result_half_limited) * 0.1, distance=20)
    peak_freqs = freqs_limited[peaks]

    # Find the peak closest to the natural frequency
    peak_distances = np.abs(peak_freqs - natural_frequency)
    closest_peak_idx = np.argmin(peak_distances)
    closest_peak_freq = peak_freqs[closest_peak_idx]

    # Fit Lorentzian only to the closest peak
    peak_width = 0.3
    peak_range = (freqs_limited > closest_peak_freq - peak_width) &
→(freqs_limited < closest_peak_freq + peak_width)
    # Adjusting initial guess for amplitude, possibly reducing it
    initial_amplitude = max(FFT_result_half_limited[peak_range] - baseline)
    initial_guess = [initial_amplitude, closest_peak_freq, 0.1]
    initial_guess = [FFT_result_half_limited[peaks][closest_peak_idx] -
→baseline, closest_peak_freq, 0.1]
    bounds = ([0, closest_peak_freq - 0.1, 0], [initial_amplitude,
→closest_peak_freq + 0.1, 1])  # Constrain amplitude and width

    try:
        popt, pcov = curve_fit(lorentzian, freqs_limited[peak_range],
→FFT_result_half_limited[peak_range] - baseline, p0=initial_guess,
→bounds=bounds)
        lorentzian_freqs = np.linspace(0.8*freqs_limited[peak_range][0], 1.
→2*freqs_limited[peak_range][0], 1000)  # Fine grid across the observed
→frequency range
        lorentzian_curve = lorentzian(lorentzian_freqs, *popt) + baseline  #
→Calculate the Lorentzian across this fine grid
        # plt.plot(lorentzian_freqs, lorentzian_curve, 'g', linewidth=2,
→label='Lorentzian Fit')
        plt.plot(freqs_limited[peak_range],
→lorentzian(freqs_limited[peak_range], *popt) + baseline, color='green',
→label='Lorentzian Fit')
        # Store peak data including HWHM and Baseline
        new_row = pd.DataFrame({
            'File': [filename.replace('.csv', '')],
            'Natural Frequency (kHz)': [popt[1]],
            'Amplitude': [popt[0]],
```

```python
            'HWHM': [popt[2]],
            'Damping Coefficient': [popt[2] / natural_frequency*1E3]
        })
        peak_results = pd.concat([peak_results, new_row], ignore_index=True)
    except RuntimeError:
        print("Fit failed for file", filename)

    plt.legend()
    plt.savefig(os.path.join(plots_dir, title + '_lorentzian_fit.png'),␣
 ↪format='png')
    plt.show()

# Display results table
print(peak_results)

# Optionally, save the peak results to a CSV file
peak_results.to_csv(os.path.join(plots_dir,␣
 ↪'peak_analysis_results_lorentzian_natural_frequency.csv'), index=False)
```

LIRAS Analysis Code

Zhen Rong Yap (2024)

```python
[1]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     from scipy.signal import find_peaks
     from scipy.fft import fft
     from scipy.optimize import curve_fit
     import os

     data = pd.read_csv('Front_Back_E_C_Damp.csv')
```

```python
[ ]: data['k'] = 1/(4*data['Height (m)'])
```

```python
[ ]: samples_list = ['A', 'B', 'C']
     no_of_samples = 4
     which_bool = 'Front'

     for i in range(len(samples_list)):
         start_index = 4 * (i)
         end_index = 4 * (i+1)

         x = data['k'].iloc[start_index:end_index]
         y = data['Natural Frequency '+which_bool+' (kHz)'].iloc[start_index:
      ↪end_index]
         coefficients = np.polyfit(x, y, 1)
         trendline = np.poly1d(coefficients)
         r_squared = np.corrcoef(x, y)[0, 1]**2
         plt.plot(np.linspace(0, 100), trendline(np.linspace(0, 100)), 'g--',␣
      ↪label=f'Extended Line of Best Fit, $R^2={r_squared:.2f}$')

         plt.scatter(x, y, color='blue')


         plt.scatter(x, y)
         plt.title('Frequency against k for '+samples_list[i]+' ('+which_bool+')')
         plt.xlabel('k = 1/4H (m)')
         plt.ylabel('Frequency (kHz)')
         plt.ylim([0, 45])
         plt.xlim([0, 30])
```

```
    plt.legend()
    plt.savefig('Frequency against k for '+samples_list[i]+' ('+which_bool+')'+'.
 ↪png')
    plt.show()
```

```
data_averaged = data.copy(deep=True)
group_index = data_averaged.index // 4
data_averaged = data_averaged.drop(columns=data_averaged.columns[0])

# Calculate the average of each group for every column
data_averaged = data_averaged.groupby(group_index).mean()
lattice_types = data['Lattice'].iloc[::4].apply(lambda x: x[0])

# Assign these types to the averaged DataFrame
data_averaged['Lattice Type'] = lattice_types.values
cols = ['Lattice Type'] + [col for col in data_averaged if col != 'Lattice Type']
data_averaged = data_averaged[cols]
data_averaged
```

```
for i, column in enumerate(data_averaged.columns):
    if column not in ['Unit Cell Size (mm)', 'Natural Frequency Front (kHz)',
 ↪'Height (m)', 'Relative Density', 'Natural Frequency Back (kHz)', 'Lattice
 ↪Type', 'k']:

        x = data_averaged['Unit Cell Size (mm)']
        y = data_averaged[column]
        coefficients = np.polyfit(x, y, 1)
        trendline = np.poly1d(coefficients)
        r_squared = np.corrcoef(x, y)[0, 1]**2
        plt.plot(np.linspace(0, 100), trendline(np.linspace(0, 100)), 'g--',
 ↪label=f'Extended Line of Best Fit, $R^2={r_squared:.2f}$')

        plt.scatter(x, y)
        plt.title(f'{column} vs Unit Cell Size (mm)')
        plt.xlabel('Unit Cell Size (mm)')
        plt.ylabel(column)
        padding = 0.5
        padding_x = (x.max() - x.min()) * padding  # 5% padding
        padding_y = (y.max() - y.min()) * padding  # 5% padding
        plt.xlim([x.min() - padding_x, x.max() + padding_x])
        plt.ylim([y.min() - padding_y, y.max() + padding_y])
        plt.legend()
        safe_column_name = column.replace('/', ' per ').replace(' ', '_').
 ↪replace('(', '').replace(')', '')
        plt.savefig(f'{safe_column_name} vs Unit Cell Size.png')
        plt.show()
```