THE MACHMOS NODE
A Low-Cost Open-Access Vibro-Acoustic Sensor Node for Machine Condition Monitoring in the Age of Artificial Intelligence

SIMON JONATHAN TREMBLAY

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Master of Applied Sciences in Mechanical Engineering

Department of Mechanical Engineering
Faculty of Engineering
University of Ottawa

Abstract

Machine Condition Monitoring and Prognostics and Health Management are long-standing industrial fields that have not yet fully benefited from the widespread adoption of artificial intelligence. Research in these areas is hindered by the limited availability of open-access machine health data required to train artificial intelligence models. This work proposes that by making machine monitoring more affordable, the acquisition of large volumes of machine health data across industrial systems can be significantly expanded. To this end, this thesis describes the design, construction, and qualification of the MachMoS Node, an open-access vibro-acoustic and thermal sensor. The machine monitoring system (MachMoS) node developed in this work is cost-effective yet meets performance requirements for industrial condition monitoring applications.

The thesis includes a review of relevant international standards in condition monitoring, prior research, and fundamental principles of sensors commonly used in this field. The methodology details the design and validation of the MachMoS Node against high-performance industrial sensors. The development process encompassed benchmarking against existing commercial sensors, defining requirements and specifications, component selection, electronic circuit and printed circuit board design, and mechanical enclosure design following Design for Manufacturing and Assembly principles to allow production by research institutions or basic machine shops, as Moreover, simulations addressing acoustic, thermal, and power-management considerations were conducted. These analyses ensure over one year of sensor operation between charges, predicted enclosure-induced resonance in agreement with experimental measurements, and provided a thermal model that allows sensor readings to be compensated for, enabling more accurate machine temperature estimations.

The resulting device is a rechargeable, wireless, vibro-acoustic and thermal monitoring node. Its performance was validated through laboratory testing, including accelerometer and acoustic frequency-response analysis, temperature calibration in an ice bath, an ingress protection assessment, and evaluation of energy usage, charging circuitry, wireless range, and external sensor connectivity. Future work will include comprehensive electromagnetic interference and compatibility testing, extended temperature evaluation, further cost optimisation, code optimisations, and long-term industrial deployment as part of a complete monitoring system.

Acknowledgements

I want to thank my partner Justine, as well as my family and friends for their unrelenting support and patience throughout this last (and perhaps final) portion of my academic career. A special thanks to my feline friends Ace and Moona for your love and everyday entertaining silliness.

Dr. Patrick Dumond for your guidance over the past years: from your 1st year Engineering Mechanics lectures, to your advisement many years later through my graduate studies, thank you. May you never stop working to improve our engineering programs and for providing students with alternative paths to success in the field.

Thank you to the uOttawa Supermileage Competitive Team, the staff of the Centre for Entrepreneurship and Engineering Design: Jason, Alex and of course Justine, whom I consider my mentors, for everything you have taught me of the world of design and manufacturing. A special thanks to Sharnado, Bruce, and now Sharkira, the projects on which I have spent countless (and I mean countless) hours, but which have also taught me so much.

Thank you, Elie, Zitai, Vincent, Saif, Sameed for your work on the code used in the MachMoS node and gateway devices, without which this project would not have been possible.

To the young bright minds of uOttawa and Algonquin College whom I've had the pleasure to teach. I hope my passion for the fields of manufacturing and engineering was contagious.

Finally, a *peekaboo!* and a cheek pinch for my niece Aria: thank you for your smiles and for your big eyes which day after day absorb more of this fascinating world. You remind me the importance of admitting ignorance and keeping an open mind as we navigate through our busy and admittedly needlessly convoluted lives.

Table of Contents

List of Figures

List of Appendix Figures

List of Tables

List of Appendix Tables

Nomenclature

2p2s	2-parallel, 2-series
3v3	3.3 Volts
AC	Alternating Current
ADC	Analog to Digital Conver.ter/sion
ADC Code	Digital binary output from an ADC
SAR	Successive Approximation Register
ΔΣ	Delta-Sigma Modulator
AI	Artificial Intelligence
AMPRES	Resultant Amplitude
BOM	Bill of Materials
CAD	
ECAD	Electrical Computer-Aided Design (software)
MCAD	Mechanical Computer-Aided Design (software)
CAM	Computer-Aided Manufacturing
CDF	Cumulative Distribution Function
CNC	Computer Numerical Control
Communication Protocol	
BLE	Bluetoooth Low Energy
I^2C	Inter-Integrated Circuit
I^2S	Inter-IC Sound
SPI	Serial Peripheral Interface
USB	Universal Serial Bus
Wi-Fi™	IEEE 802.11
COTS	Commercial Off the Shelf
CPU	Central Processing Unit
CVD	Callendar Van-Dusen (Equation)
DC	Direct Current
DDL	Dumond Design Lab
DFMA	Design for Manufacturing and Assembly
Distributors/Manufacturers	
Digi-Key	Global distributor of electronic components
JLCPCB	Accessible Chinese manufacturer of PCBs
McMaster-Carr, McM-C	Large American hardware supplier
DMA	Direct Memory Access
DoF	Degrees of Freedom
DSP	Discrete Signal Processing
DUT	Device Under Test

EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EMPF	Effective Mass Participation Factor
Filters	
BP	Band Pass
FIR	Finite Impulse Response
HP	High Pass
LP	Low Pass
Fourier Transform	
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
FT	Fourier Transform
IFFT	Inverse Fast Fourier Transform
IFT	Inverse Fourier Transform
FPS	Frames per Second
GPIO	General Purpose Input/Output
IC	Integrated Circuit
ICES	Interference-Causing Equipment Standard
ID	Inner Diameter
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IO	Input/Output
IoT	Internet of Things
IP	International Protection (Rating)
IR	Infrared
ISO	International Organization for Standardization
KPI	Key Performance Indicator
LoS	Line of Sight
MCM	Machine Condition Monitoring
MCU	Microcontroller Unit
MEMS	Microelectromechanical System(s)
MUT	Machine Under Test
NiMH	Nickel Metal-Hydride
OD	Outer Diameter
ODR	Output Data Rate
PCB	Printed Circuit Board
PDF	Probability Density Function
p-FET	P-channel MOSFET

PFM	Pulse-Frequency Modulation
PHM	Prognostics and Health Management
pk-pk	Peak to Peak
Polymers	
ABS	Acrylonitrile Butadiene Styrene
ASA	Acrylonitrile Styrene Acrylate
Buna-N	Copolymer of butadiene and acrylonitrile
Nylon	Aliphatic Polyamide
PET-G	Polyethylene Terephtalate Glycol
PLA	Polylactic Acid
R_a	Roughness Average
RF	Radio Frequency
RH	Relative Humidity
RMS	Root-Mean Squared
RSS	Radio Standards Specification
RTD	Resistance Temperature Detector
RUL	Remaining Useful Life
SME	Small and Medium-Sized Enterprise
SPS	Samples per Second
Threads	
NPT	National Pipe Taper
UNC	United National Coarse
UNF	United National Fine
VFD	Variable Frequency Drive
Voltage Regulator	
LDO	Low Dropout
SMPS	Switch Mode Power Supply
w.r.t.	with respect to
WSN	Wireless Sensor Network
χ^2	Chi-Squared (Distribution)

Chapter 1 Introduction

1.1 Machine Condition Monitoring

Machine Condition Monitoring (MCM) is a practice in which machines are monitored to assess their health. Conclusions drawn from monitoring machinery can inform the need for maintenance, the goal of which is to ensure a manufacturing chain does not suffer breakdowns.

Machine monitoring can be performed in many ways. Some common examples include using temperature, acoustic (including ultrasonic), and vibration measurements. Traditionally, such measurements were taken by MCM specialists who would walk a factory line equipped with a range of portable sensing equipment. However, it is impossible for a few MCM specialists to monitor hundreds of machines with portable sensors, and therefore, it becomes incumbent on floor workers to report on symptomatic machinery (increased noise, vibrations, etc.). Unfortunately, as plant automation becomes more prevalent, fewer floor workers are available to flag symptomatic machinery [1], [2]. Similar challenges are present in the case of monitoring systems installed in remote locations, such as radar systems, wind turbines, solar tracking systems, and hydroelectric dams, to name a few. With the advent of intelligent Internet of Things (IoT) sensors, such measurements can be recorded online at regular intervals, without the presence of a human.

Once data is acquired, machine data is compared with baseline (healthy) readings, and inferences are made as to the health of the machine. This process can be complex. To make inferences, MCM specialists analyze the data and, using knowledge of the machinery being analyzed (contextual data), they may make specific conclusions about the faults that exist in the machinery. They consider the machine's rotating speed, bearing ball counts, bearing sizes, bearing types, motor type, motor pole pairs, couplings attached to the machinery, axial and radial loads on the shafts of the machinery, and many more factors. Using all this contextual data, one can build a narrative as to what machinery components need servicing, and maintenance scheduled accordingly.

1.2 Motivation

Operational downtime prevention through MCM is a very valuable and growing industry, with a market size of approximately 3 to 4 billion USD worldwide in 2024 [3], [4], [5], though implementation of MCM systems can bear a significant cost burden. To actively monitor a plant, MCM specialists must equip all critical machinery in a facility with dedicated sensors. This can quickly become very expensive using commercially available sensing products. These costs remain prohibitive for small and medium-sized enterprises (SMEs). Concerns over privacy and the proprietary nature of data currently being acquired by enterprises that can afford MCM systems further complicates this problem. This limits data sharing, which prevents researchers attempting to modernise the field of MCM from creating intelligent models that allow for the automated analysis of sensor data. Ultimately, the development of automated intelligent models would significantly lower implementation cost of MCM programs.

In this age of big data and as Industry 4.0, data collection, storage, and processing have become more accessible than ever. The wide adoption of Artificial Intelligence (AI) models to solve challenges in industry presents an option for automating inferences in the field of MCM. However, advancements in the field are limited by the scarcity of open-source data. This prevents MCM systems benefiting from the insights offered by AI in trends and patterns of measurements affecting equipment failure. The availability of open-source data through the implementation of low-cost sensors in plants would reduce equipment failure and would offer a reduction in costly operational shutdowns.

1.3 Objectives and Contributions

The work of this thesis outlines the design, construction, and testing of a low-cost reliable MCM sensor. The availability of such a sensor is key to the creation of large, multivariate datasets that will be useful to the training of AI algorithms. This sensor features an enclosure that is designed to be simple to manufacture using machinery that can be found in most machine shops. As part of the design, a 3D-printed cover, which includes captive hardware, and two O-ring interfaces, will be tested for ingress protection against fluids, demonstrating the validity of additive manufacturing methods in the low-volume production of enclosures. Additionally, the work outlines various testing methods used to validate a microelectromechanical system (MEMS) accelerometer and microphone against a capacitive-type analog accelerometer and microphone. It also describes the design and testing of a resistance temperature detector circuit. Finally, the work describes the design of a 2-parallel, 2-series (2p2s) Nickel Metal Hydride (NiMH) battery pack on a printed circuit board (PCB), and outlines considerations for such a board, including safety, charging, and expected battery life. In this thesis, we have named the novel device the machine monitoring system (MachMoS) node, as it acts as a node in a larger IoT ecosystem.

1.4 Thesis Structure

This thesis documents the design, construction, and testing of the MachMoS node, and is divided in the following chapters:

2. **Literature review**, which contains the necessary fundamentals and information to inform the decisions made during the design process.
3. **Methodology**, which outlines the process in which the device will be designed and tested.
4. **Device design and construction**, which discusses the design of the device, including benchmarking, individual component selection, creation of the enclosure and printed circuit boards, as well as their assembly and integration.
5. **Results**, which describes the testing conducted to validate the design, as described in the methodology section, as well as any deviations from testing procedure.
6. **Conclusion**, which highlights the successful design and construction of the MachMoS node for temperature and vibro-acoustic measurements.

A detailed appendix is provided at the end of this thesis to support the text with various information, analyses, method validations, build documentation, and design resources specific to the device. The code that was written by the author over the course of this project is also included in the appendix.

Chapter 2 Literature Review

2.1 Machine Condition Monitoring

MCM consists of using sensors to measure various signals emitted by machinery. These signals can be used to monitor the health of machinery. For many years, vibration, sound, and temperature signals have been used for condition monitoring [6], [7]. Current draw and stray magnetic flux has also been used to detect faults in induction motors [8], [9]. Sensors used to measure these signals are discussed in later sections of this literature review.

2.1.1 Condition Monitoring Standards

The following standards from the International Organisation for Standards (ISO) exist for condition monitoring of machinery. Standards can provide valuable insight on the characteristics of the machinery in industry. The list of standards that follow is not exhaustive.

2.1.1.1 ISO 17359

The ISO 17359 standard: Condition monitoring and diagnostics of machines — General Guidelines is a guide listing the procedures that must be considered when a condition monitoring program is to be set up for machines. It lists some of the condition monitoring standards and is applicable to all machines [10].

2.1.1.2 ISO 10817

The ISO 10817 standard: Rotating shaft vibration measuring systems. Part 1: Relative and absolute sensing of radial vibration provides guidance on the sensing device, signal conditioning, attachment methods, and calibration procedures for absolute and relative radial measurement methods on shafts [11]. This standard is an important reference for proper vibration sensing but was not consulted for this work.

2.1.1.3 ISO 22096

The ISO 22096 standard: Condition monitoring and diagnostics of machines — Acoustic emission provides general principles for the use of acoustic emission in condition monitoring of machinery operating under a range of conditions and environments [12]. The ISO 22096 standard lists multiple faults that may be monitored using acoustic emissions, such as bearing deterioration, mechanical seal rubbing, generalised wear, lubrication contamination and loss, severe misalignment, and mounting faults.

2.1.1.4 ISO 13379 and ISO 13381

The ISO 13379 standard: Condition monitoring and diagnostics of machine systems — Data interpretation and diagnostics techniques is split into two parts: Part 1: General Guidelines [13] and Part 2: Data-driven applications [14]. The standards provide guidelines for the interpretation of data and the diagnosis of machines. Training of a statistical model during normal machine operation is recommended by the standard, followed by the regular comparison of the machine data to that of the statistical model for deviations. The method from the standard is much like what is found in academic literature, where artificial intelligence (based on statistics) is used to determine the health state of machinery (subsection 2.1.2). The ISO 13381 standard: Condition monitoring and diagnostics of machine systems — Prognostics is meant to act as a guide for the development and implementation of prognosis processes [15]. It expands on ISO 13379 to obtain a measure of machinery remaining useful life (RUL).

2.1.1.5 ISO 13373

The ISO 13373 standard: Condition monitoring and diagnostics of machines — Vibration condition monitoring contains 10 parts, 2 of which remain under development. This series of standards aims to provide a more generalised perspective for the use of vibration data, where conclusions (diagnoses) are drawn from the change in the vibration behaviour of machinery, and not the specific behaviour of the machinery. The 10 parts to the standard are the following:

- Part 1: General Procedures [16]
- Part 2: Processing, analysis and presentation of vibration data [17]
- Part 3: Guidelines for vibration diagnosis [18]
- Part 4: Diagnostic techniques for gas and steam turbines with fluid-film bearings [19]
- Part 5: Diagnostic techniques for fans and blowers [20]
- Part 7: Diagnostic techniques for machine sets in hydraulic power generating and pump-storage plants [21]
- Part 8: Diagnostic techniques for industrial pumps (under development) [22]
- Part 9: Diagnostic techniques for electric motors [23]
- Part 10: Diagnostic techniques for electrical generators with fluid-film bearings [24]
- Part 11: Diagnostic techniques for gearboxes (under development) [25]

The first part of the series presents specifics of establishing machine condition monitoring systems, such as sensor (transducer) types, transducer placement, frequency range, continuous vs. periodic monitoring, trending, and alarms. The second part of the series presents fundamentals of vibration signals, such as signal frequency content (spectra), signal envelopes, transients, etc. The third part of the series presents general guidelines for the monitoring of a large range of machinery. The other parts of the series are guidelines aimed at specific types of machinery.

2.1.1.6 ISO 20816

The ISO 20816 series of standards: Mechanical vibration — Measurement and evaluation of machine vibration supersede the older standards: ISO 7919 (measurements performed on rotating parts of machinery) and ISO 10816 (measurements performed on non-rotating parts of machinery). The first part of this series repeats some of the content provided by ISO 13373 on the selection and placement of sensors but also covers the setting of alarms/trips when trending machine vibrations. The other parts of the standard provide the vibration limits for various types of machinery. The parts to the standard are the following:

- Part 1: General Guidelines [26]
- Part 2: Land-based gas turbines, steam turbines and generators in excess of 40 MW, with fluid-film bearings and rated speeds of 1 500 r/min, 1 800 r/min, 3 000 r/min and 3 600 r/min [27]
- Part 3: Industrial machinery with a power rating above 15 kW and operating speeds between 120 r/min and 30 000 r/min [28]
- Part 4: Gas turbines in excess of 3 MW, with fluid-film bearings [29]
- Part 5: Machine sets in hydraulic power generating and pump-storage plants [30]
- Part 8: Reciprocating compressor systems [31]
- Part 9: Gear Units [32]
- Part 21: Horizontal axis wind turbines [33]

The vibration limits provided by ISO 20816 are often summarized by a table of machine power, mounting type, and machine health as a function of root-mean squared (RMS) vibration. A summary table based on Standard ISO 10816-1:1995 and ISO 20816-1:2016 can be found in reference [34]. The acceptable vibration limits for pumps > 15 kW on a rigid foundation are less than 1.4 mm/s RMS for new and reconditioned machines, 2.3 to 2.8 mm/s RMS for machines that can operate indefinitely, and 3.5 to 4.5 mm/s RMS for machines that cannot operate for a long time. In addition to this, if the vibration is above 7.1 mm/s RMS, the vibrations are causing damage.

The ISO 10816-1:1995 standard lists the vibration severity for electric motors with a power rating < 15 kW and are as follows [35]: less than 0.71 mm/s RMS for good, 1.12 to 1.8 mm/s RMS for satisfactory, 2.8 to 4.5 mm/s RMS for unsatisfactory, and over 7.1 mm/s RMS for unacceptable operation.

2.1.2 Artificial Intelligence in MCM

Within the field of artificial intelligence exists machine learning and deep learning methods. Conventional machine learning methods (including artificial neural networks, principal component analysis, support vector machines) have been used to detect and categorize equipment faults for decades [36]. Machine learning is also used in the field to build early warning systems, which aim to predict the failure of a system. These systems make use of sensor data moments (mean, variance, skewness and most notably kurtosis) to infer that equipment has become unstable and will eventually fail [37], [38], [39].

In the past 10 years, the increase in available computation power has led to a rise in the study of deep learning methods. These methods have been shown to be superior to machine learning methods in diagnosis and classification tasks. It has also recently been shown that deep learning methods internalize moments of a dataset [40], [41] such that (among many other reasons) they are particularly well-suited for early warning systems and fault detection.

When testing an AI model, a dataset is separated into two subsets: a *train* subset, which is used to train the AI model, and a *test* subset, which is used to test the model. Much of the AI models tested in the literature are validated using datasets that vary motor loads during tests, which has no impact on bearing loads. This is specifically the case for the Case Western Reserve University (CWRU) dataset [42]. In the past, researchers have been criticized for unknowingly cross contaminating their *test* subsets with *train* data when testing AI models. A benchmarking method for AI models using the CWRU dataset, in which no cross-contamination occurs has been proposed by [43]. This method proposes to test model performance by testing the model on bearings with various fault sizes, and not on different motor loads. When this method is used, the average accuracy of AI models used successfully in the literature (ACDIN, WDCNN, AlexNet, ResNet) is significantly reduced from 80-90 % to at most 53 %.

Sehri, M. and Dumond, P.'s work on Optimizing Rolling Element Bearing Data Collection and Algorithm Hyperparameters for Machine Learning [44] makes recommendations as to the minimum sampling frequencies and capture lengths required to ensure good deep learning model performance. These recommendations include 4 s capture lengths for accelerometer data, and 5 s for microphones provided that the sampling rate used can capture at least 40 fault harmonics (see subsection 2.4.2). Four and five seconds per capture provides a spectral resolution of 0.25 Hz and 0.2 Hz per capture, respectively (see subsection 2.1.2).

These results show that much work remains to be done in developing AI models for MCM. Real-world data is also lacking in the literature. The models that are currently being developed are trained on noise-free laboratory-generated datasets, and the data that is available in these datasets is often insufficient to train robust and generalized AI models that can be successfully transferred to industry [36].

2.2 Existing Datasets for Preventative Maintenance

Large quantities of vibration, temperature, and acoustic data exist in industry, but this data remains proprietary and is not accessible to researchers. As such, the datasets that are open access are typically generated by universities, colleges, and governmental organisations.

The CWRU in the United States hosts a dataset of seeded fault tests [45]. Seeded fault tests are conducted on bearings that have been artificially damaged in a controlled manner. This dataset is widely used in the literature to study bearing faults. It is also used as a means to benchmark the accuracy of various fault classification AI models [46]. However, this dataset has been criticized for how the test rig is set up. In the tests, the same bearings are tested at various motor loads, which does not increase the radial or axial loads on the bearings [47]. However, increased load increases the fields generated by induction motors, causing measurable interference in the vibrations signals [8], [48]. The dataset contains tests conducted on healthy bearings and bearings with inner race faults (5 different fault sizes). These tests were repeated for various motor loads from 0 to 3 hp and motor speeds from 1720 to 1797 RPM. The dataset contains data from three accelerometers, one on either end of the drivetrain in the setup and one on the base of the setup [45].

The University of Paderborn in Germany hosts a similar dataset to that of the CWRU dataset. However, this dataset contains more bearings (6 healthy, 12 with seeded faults) and does not only have test data for bearings with seeded faults but also contains data for 14 different bearings with faults caused by accelerated life tests. The dataset contains data from a single accelerometer, an RPM sensor, a temperature sensor for the oil temperature in the bearing module, and load sensors to measure the radial forces on the bearing under test and the shaft torque [49]. Zhang et al. consider this dataset better suited to the training of AI algorithms for practical applications due to the inclusion of bearings damaged by accelerated life tests [36].

NASA as well as some universities have provided datasets for the purpose of Prognostics and Health Management (PHM) challenges [50], [51], [52], [53], [54]. These challenges present a good way to benchmark algorithms developed by researchers, and is not only limited to machine condition monitoring, but also to the prognostic of machinery. The PRONOSTIA dataset [51] for the study of bearing RUL was used in the Institute of Electrical and Electronics Engineers (IEEE) PHM 2012 Prognostic Challenge, and remains a popular dataset for RUL prediction algorithms [36]. The PRONOSTIA dataset used accelerated life tests and captured the full life of bearings. Measurements from two accelerometers (two axes), a load module to apply and measure loads to the bearing under test, a temperature detector, a torquemeter, and an RPM sensor are included in the data [51].

Finally, the University of Ottawa recently released the University of Ottawa Rolling-element Dataset – Vibration and Acoustic Faults under Constant Load and Speed conditions (UORED-VAFCLS) dataset, which also captures the full life of bearings through accelerated life tests [55]. The data available in this dataset is like that of the PRONOSTIA dataset in that it consists of accelerated life tests, but notably also contains microphone data. It also has longer capture cycles (10 s per cycle as opposed to 0.1 s each 10 s in the PRONOSTIA dataset) and more bearings tested (20 as opposed to 11). The dataset provides the types of faults, and conveniently contains the machine health labels: *healthy*, *developing fault*, and *faulty* for all captures.

The aforementioned datasets are all useful for creating predictive models. However, these datasets are all generated in laboratories, under controlled conditions. Therefore, models developed with these datasets, understandably, have difficulty transferring to the real world [56], [36].

2.3 Periodic Signals, the Fourier Theorem and Transform

Periodic signals are signals that are repeated indefinitely. The Fourier Theorem states that all periodic continuous signals can be represented using a sum of sinusoidal signals (sines and cosines) of various frequency and amplitude [57]. The Fourier transform (FT) provides a method of finding which frequency components (and their amplitudes) are present in a non-periodic signal (spectrum). The inverse Fourier transform (IFT) can be used to reconstruct a signal in the time domain using the frequency components of the signal. In discretely sampled signals, the discrete Fourier transform (DFT) is used to obtain the spectrum of the data. The resulting spectrum provides the magnitudes for discrete frequency bins. The size of the bins is referred to as the spectral resolution of the FT and is a function of the sampling frequency and the sampling length [58]. The DFT and its inverse have been optimised for computation through the fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT) developed by Cooley and Tukey [59].

2.4 Using Vibro-Acoustic Data Spectra for Diagnosis

The analysis of vibration measurement spectrum is a powerful tool in MCM that is used regularly in industry [60], [61] and in academic papers [36]. When analysing the frequency content of a vibration signal, the magnitude of specific frequency components is used to determine if a machine requires servicing. This spectrum analysis also provides valuable insight into the fault that is causing the elevated vibrations (shaft misalignment, imbalance, defective bearings, etc.). Such faults are identified by observing specific frequency components considered fundamental to a specific system. These fundamental frequencies are determined as described in the following subsections.

2.4.1 Frequencies Emitted by Shaft Misalignment and Imbalance

Newly commissioned machines might have installation issues such as shaft angular or parallel misalignment. It is also possible for this machinery to be imbalanced. Shaft misalignments will generate vibrations at frequencies double that of the rotation frequency of the machinery. Imbalances will generate vibrations at the rotation frequency [62].

2.4.2 Frequencies Emitted by Defective Bearings

Bearings are composed of an inner and outer race, between which rolling elements are contained. These rolling elements are spaced using a cage. Degradation of bearings through cage, ball, and race faults cause the amplitude of the vibrations generated by a system to grow as the faults become more pronounced. This increase in vibration amplitude can be used to identify faults in the spectrum of vibration signals at the bearing’s fault frequencies [47], and many investigators have used this technique successfully [6].

When f_r is considered to be the shaft speed (or the speed of the inner race relative to the outer race), n the number of rolling elements, d the rolling element diameter, D the pitch circle diameter of the bearing, ϕ the contact angle, the bearing fault frequencies can be calculated using equations (1) to (4) in Table 2.1.

However, these fault frequencies are specific to bearings with single point defects, and while literature exists to relate the frequencies to accelerometer and acoustic data, bearing degradation in industry is caused by general roughness, where multiple surface defects form a general roughness of the races and/or rolling elements. General roughness might explain some of the disconnect between the methods used in industry to diagnose machinery and those used by researchers [63]. Fault harmonics are often referenced in the field of MCM. Fault harmonics are peaks that are expected in the spectrum of a bearing's vibration signal if faults become more pronounced. These peaks are found at multiples of the fault frequency for ball pass frequency outer (BPFO) and ball pass frequency inner (BPFI) race faults, and multiples of twice the ball spin frequency (BSF) for faults on rolling elements [64].

Table 2.1 - Bearing Fault Frequencies [64]

Frequency	Equation	Nr.
Ball pass frequency, outer race (BPFO)	$\mathrm{BPFO} = \frac{nf_r}{2}\left(1 - \frac{d}{D}\cos\phi\right)$	(1)
Ball pass frequency, inner race (BPFI)	$\mathrm{BPFI} = \frac{nf_r}{2}\left(1 + \frac{d}{D}\cos\phi\right)$	(2)
Fundamental train frequency (cage speed) (FTF)	$\mathrm{FTF} = \frac{f_r}{2}\left(1 - \frac{d}{D}\cos\phi\right)$	(3)
Rolling element spin frequency (BSF)	$\mathrm{BSF} = \frac{Df_r}{2d}\left[1 - \left(\frac{d}{D}\cos\phi\right)^2\right]$	(4)

As there are no limits to the rotational speed of machinery in industry, nor limits on the number of rolling elements in a singular bearing, there is no way to determine the maximum frequency that may be generated by machinery. However, a good coverage of machinery in industry can be obtained with accelerometers operating at sampling frequencies high enough to capture signals oscillating at up to 2.5 kHz, according to [65], though this recommendation is most likely for traditional fault diagnosis which only requires 10 harmonics to be captured [66], and an approach using machine learning would require a sampling rate that is four times larger than this rate (40 harmonics per [55]). This corresponds to a sampling rate of at least 20 kHz considering the Nyquist-Shannon sampling theorem discussed in subsection 2.1.2.

2.5 Filters

Filters consist of devices or programs that attenuate noise in a signal to increase the Signal to Noise Ratio (SNR). Filters can be used to attenuate the energy of frequencies above, below, or between specific frequencies. These filters are known as low pass (LP), high pass (HP), and band pass (BP) filters, respectively [67]. LP filters attenuate frequencies beyond a specific cut-off frequency, whereas HP filters attenuate frequencies from 0 Hz (DC) to a cut-off frequency, and BP filters will attenuate all frequencies outside of a range of frequencies. A BP filter can be created from a combination of a LP and a HP filter. Finite impulse response (FIR) filters also exist, in which specific frequencies are rejected from the band. The cut-off frequency of a filter is typically considered the point at which the filter has a gain of -3 dB. That is, when the filter output magnitude is roughly 71 % of the input magnitude. The filter will then progressively attenuate signals until its stopband, at -40 dB (1 % gain).

Filters can be very useful in attenuating high frequency content from signals going into a sensor, as the captured signal could otherwise suffer from aliasing. It is important to note that when discussing the Nyquist frequency of a system, the theorem assumes an ideal LP filter which can attenuate all frequency content of a signal that is above the Nyquist frequency. Since an ideal filter cannot be implemented (although digital filtering can sometimes get close), calculations for the highest frequency content allowable for a sensor is typically accompanied with a safety factor. In machine condition monitoring, it is common to calculate the Nyquist rate (the sampling rate required to capture a signal without aliasing) as being 2.56 times the highest frequency of the phenomena being observed (see section 2.6.3) [68], [69], [70].

2.6 Discrete Signal Processing

Discrete signal processing (DSP) is a very important subject in the world of monitoring. All signals are continuous in nature. That is, for any time t, there exists a value y that describes the signal (e.g. magnitude, rate of change, etc.). In digital systems, such as those used to capture signals, it is impossible to capture signals in their entirety, as digital systems are discrete, that is, they may only sample systems a finite number of times (in a non-continuous fashion), leaving an infinite number of un-captured signal states [71]. Practically speaking, digital systems cannot sample continuous signals continuously as they do not have an infinite amount of storage (that is, they simply cannot store a dataset such that the state y of a signal can be resolved for any time t). Digital systems must also perform various tasks to transform an analog signal into a digital value it can store, including analog to digital conversion (A/D conversion), filtering (in discrete time), and storage. When sampling signals, discrete systems must be able to capture a bare minimum amount of information about a continuous signal to ensure that useful conclusions about the state of the system may be inferred. Such information (data) about a signal may be considered lossless, as opposed to a lossy signal, form which conclusions inferred may be erroneous, or where inference is simply impossible. Acquisition systems must be designed and used appropriately to make them useful. In DSP, noise can infiltrate a signal during the analog to digital conversion (quantization noise), or if aliasing occurs.

2.6.1 Analog to Digital Converters

Analog to Digital Converters (ADCs) are devices that convert an analog (continuous) signal into a digital signal (discrete). The number of bits in the representation of the discrete signal generated by an ADC is typically one of the main characteristics of an ADC, however, some ADCs may output 20 to 24-bits of resolution, but only 12 of those bits are stable (noise-free). The sampling rate of an ADC is also of the utmost importance, as aliasing can occur in a capture, a phenomenon that is near-impossible to attenuate once the signal has been converted (see 2.6.3). ADCs can convert analog signals in many different ways, but the most common ones are Pipeline ADCs, Successive Approximation Register ADCs (SARs), and Delta Sigma Modulators ($\Delta\Sigma$) ADCs [72]. Flash ADCs (parallel ADCs) are also used, but only for very specific applications where extremely high sampling frequencies are desired, such as in oscilloscopes (GHz range).

Generally, the resolution of an ADC is higher when the sampling rate is lower, both due to implementation challenges as well as the architecture of each type of ADC. For instance, flash ADCs can make a conversion at each digital clock pulse, making it very fast. However, the architecture of flash ADCs is such that the complexity of the ADC goes up exponentially for each bit of resolution. $\Delta\Sigma$ ADCs, on the other hand, use a switched capacitor design to capture and hold a signal, but then need multiple clock cycles to perform a single conversion. ΔΣ comparators also use a very high frequency clock to oversample the input signal, and outputs a single-bit (low resolution) bit stream that is filtered through a digital decimator to obtain a low frequency, high resolution signal. Since ΔΣ ADCs use a simple single-bit stream to encode the input signal, its complexity does not increase significantly with a higher bit-resolution requirement. Its resolution is related to its oversampling rate and the performance of its individual components, such that it can be implemented with high bit resolution at the cost of lower sampling rates.

A quick search through the *Texas Instruments* offerings for ADCs illustrates this relation, with folding interpolating ADCs (which builds on the flash architecture) boasting up to (at most) 12 bits of resolution and GHz range sampling rates. These, of course, come at great cost given the complexity in their implementation, with the cheapest 12-bit ADC costing $90 USD, for a maximum sampling rate of 0.8 GHz. The integrated circuit (IC) features 144 pins to interface. On the other end of the spectrum, a space-rated 10.4 GHz ADC costs $51.5k USD. ΔΣ ADCs in the *Texas Instruments* offerings can sample up to 32-bits of resolution, but do not exceed 38 kHz sampling rates, for which the cost can be as low as $10 USD [73]. These architectures are the extremes of the sampling rate vs. resolution spectrum, with the SAR ADC second best in resolution (with higher sampling rates than ΔΣ ADCs), and pipeline ADCs between the SAR and the flash ADCs [72].

It should be mentioned that the presence of oversampling and decimation in the architecture of ΔΣ modulators makes these ADCs resistant to high frequency noise (the decimation stage in the architecture is a low-pass filter). Other digital components use the principle of oversampling and decimation for the analog to digital conversion and consequently benefit from integrated low pass filtering (either by implementing a ΔΣ ADC or outputting an oversampled bitstream). Examples include MEMS accelerometers, gyroscopes, pressure sensors, and microphones [74], [75].

2.6.2 Quantization Noise

Quantization noise refers to noise that enters the signal as it goes through the process of quantization, such as that which happens in an ADC device [71]. One source of quantization noise is called round-off error and stems from the error introduced by the lack of bits available to represent a sample adequately. For instance, a 3-bit ADC that may only output positive numbers would have only the integers 0 to 7 as possible outputs. If the full scale of the device is from 0 V to 7 V, this means the ADC may only resolve readings of 0 V, 1 V, 2 V..6 V, 7 V. In this case, if a signal has a non-integer voltage, it will be rounded in the quantization process. The difference between the ADC's rounded output and the true signal is quantization noise.

2.6.3 Nyquist-Shannon Theorem

In the simplest of terms, the Nyquist-Shannon theorem states that the sampling frequency required to capture a signal must be at least double the frequency of the signal of interest to obtain perfect reconstruction of the signal [76]. The theorem presents two concepts: the Nyquist rate and the Nyquist frequency. The Nyquist rate is defined as the minimum (exclusive) rate at which a periodic signal must be sampled to ensure it can be completely reproduced. The Nyquist rate f_N can be calculated simply from the maximum frequency of interest in a signal of frequency B: $f_N = 2B$. Conversely, the Nyquist frequency $f_{\max}$ is the maximum frequency that can be captured for a specific sampling rate. That is, $f_{\max} = f_s/2$, where f_s is the sampling rate. It is useful to make the distinction between f_N and $f_{\max}$, as f_N is a requirement for the capture of a known signal, whereas $f_{\max}$ is a limitation of an existing sampling system.

2.6.4 Aliasing

Aliasing is a phenomenon which occurs when the sampling frequency of an acquisition system is less than the required Nyquist rate for a specific signal. When this occurs, high frequency information about the original signal is lost, and appears in the measured signal as an alias of the fast signal, in the form of a slower signal [71]. An imprint of the higher frequency signal is observable in the frequency domain representation of the captured signal. In the frequency domain, the aliased frequency appears to have been folded (reflected) across the $f_{\max}$ of the sampling system. Aliasing is a concerning problem in signal processing, as aliasing happens without leaving any clues behind. It is impossible to filter out an aliased signal once it is captured, unless the frequency of the aliased signal and the sampling rate are known, though filtering at this stage induces obfuscation of part of the original signal. The challenges of aliasing highlight the importance of proper signal filtering before sampling occurs. Aliasing's folding effect is very useful in evaluating a filter's effectiveness. To test a system of known sampling frequency, one can introduce a signal above $f_{\max}$. A system with sufficient filtering will show no aliased peaks in its frequency content. An example of aliasing in the case of a video of an accelerating vehicle is presented in Appendix B.1.

2.7 Current MCM and PHM Systems

There exist many commercially available sensors to gather data for the purpose of MCM. Some of these sensors are described in detail and benchmarked in section 3.1. These sensors are typically marketed as a system which includes MCM platforms or software, where the data can be streamed by users. Some of these systems are more sophisticated than others and include analysis features that are specific to PHM. The software that is used with the sensors presented in section 3.1 are listed below with a short description.

Table 2.2 - Some Existing MCM and PHM Platforms

Software/Platform (*Company*)	**Type (MCM/PHM)**	**Description**
RotationLF™ (*Nanoprecise*)	PHM	"(…) scalable and sensor agnostic platform that monitors the condition of the machines in real-time, and predicts their Remaining Useful Life." [77]
System-1 (*Bently Nevada*)	PHM	"(…) gathers and stores important machine health data across [an] entire enterprise in one central location, for real-time analysis, diagnosis, and preventative condition-based maintenance." [78]
Echo® Monitoring Software (*PCB Piezotronics*)	PHM	A platform that allows for the storage and viewing of data. This includes setting alarms and generating reports [79].
Treon Connect/Apex (*Treon*)	MCM or PHM	"A suite of ready-made operational applications, including condition monitoring, fleet management, and events for workflow automation" [80]
NCD Base Station (*NCD.io*)	MCM	An open-access application for configuring, managing, and gathering data [81]
Node-RED (*OpenJS Foundation*)	MCM or PHM	"Node-RED's goal is to enable anyone to build applications that collect, transform and visualize their data (…)" [82]. Platforms such as Node-RED can be used by platform-agnostic sensors to host data and create dashboards with data analysis that includes PHM functionality.

2.7.1 OpenPHM

OpenPHM is a PHM system proposed by the Dumond Design Laboratory at the University of Ottawa (DDL) [83]. The openPHM system comprises a community of researchers, a database, a webapp (portal), a data acquisition system, and industrial users. OpenPHM attempts to respect the structured data ontology and schema for PHM systems used for AI proposed by [84]. OpenPHM is the primary software to be used with the MachMoS system.

2.8 Sensors

2.8.1 Temperature

There exist many types of temperature measurement devices. Among the popular ones are thermistors, infrared measurement devices, thermocouples, and resistance temperature detectors (RTD).

2.8.1.1 Thermistors

Silicon-based diodes can be used as temperature transducers, as they have a linear temperature coefficient. The p-n junction in bipolar transistors is often used as a diode for temperature measurements due to the transistor's consistent temperature coefficient. A constant current source large enough to excite the diode can then be used to generate a potential difference (forward voltage) across the device. The forward voltage is compared to the temperature coefficient to obtain the temperature of the transistor. Some ICs exist which integrate the transistor, necessary amplifier, ADC and digital engine to generate a digital output (typically I^2C or SPI) [85], [86]. While thermistors are good for making measurements, their readings are known to drift over time due to degradation [87]. As such, they are not suitable for use over long periods of time.

2.8.1.2 Infrared Temperature Detector

Infrared (IR) temperature detectors are widely used for temperature measurements as they offer the possibility of safely taking contact-less temperature measurements. These devices use the concept of radiative heat transfer to make temperature readings. As a reminder, every part of a surface in which energy is bound emits energy through radiation. The ability of a grey body (non-ideal radiator) to emit energy through radiation, compared to a blackbody (an ideal radiator) is given by its emissivity ε, a ratio from 0 to 1. The rate of energy radiated by a surface of emissivity ε_s at surface temperature T_s $[K]$ due to its own internal energy is given by equation (5) [88].

$$q''_{\text{rad}}(T) = \sigma \varepsilon_s T_s^4 \tag{5}$$

where σ is the Stefan-Boltzmann constant: 5.67E^{-8} W/m^2·K^4. This equation considers the energy radiated across the whole electromagnetic spectrum. The net energy flux for the surface, however, must also consider the energy reflected and transmitted. A surface's capacity to reflect and to transmit are given by ratios φ_s (reflectivity) and τ_s (transmissivity), respectively, and the sum of these three ratios must be 1 to satisfy conservation of energy, as all radiation must be reflected, absorbed, or transmitted [88]. Formally,

$$\varepsilon_s + \varphi_s + \tau_s = 1 \tag{6}$$

In an infrared temperature device, the energy transferred to the sensor in the device (pyroelectric material) from the surface being measured generates an electric signal V_{pyr}, and is only sensitive to IR radiation. This restricted band means that the Stefan-Boltzmann constant and exponent are substituted for a constant C, and exponent n, both functions of the detector's specific sensitive band. That is, for a specific band, the surface emits at a rate of $C\varepsilon_s T_s^n$. It should be reiterated that the surface being measured does not only radiate its own internal energy, but also background radiation, which it reflects. Most importantly, background radiation comprises radiation that is emitted ($C\varepsilon_{\text{amb}}T_{\text{amb}}^n$), as well as radiation that is reflected ($C\varphi_{\text{amb}}T_{\text{amb}}^n$), by all ambient surfaces. If an assumption is made for no transmissivity of the incident radiation for any of the surfaces, then their reflectivity is given by $1 - \varepsilon$ (this is an appropriate assumption for irradiation of most surfaces in the IR band [89]), and $\varepsilon_{\text{amb}} + \varphi_{\text{amb}} = 1$. Finally, the detector itself is designed as a black body, such that it emits all incident radiation ($\varepsilon_{\text{pyr}} = 1$). As such, the voltage at the detector, V_{pyr}, can be given by equation (7).

$$\begin{gathered} V_{\text{pyr}} = C\varepsilon_s T_s^n + \varphi C(\varepsilon_{\text{amb}} + \varphi_{\text{amb}})T_{\text{amb}}^n - C\varepsilon_{\text{pyr}}T_{\text{pyr}}^n \\ V_{\text{pyr}} = C\left[\varepsilon_s T_s^n + (1-\varepsilon)T_{\text{amb}}^n - T_{\text{pyr}}^n\right] \end{gathered} \tag{7}$$

from which the surface temperature can be isolated [89], giving equation (8).

$$T_s = \sqrt[n]{\frac{V_{\text{pyr}} - CT_{\text{amb}}^n + C\varepsilon_s T_{\text{amb}}^n + CT_{\text{pyr}}^n}{C\varepsilon_s}} \tag{8}$$

As the detector is contained in the measurement device, its temperature is measurable with a thermocouple. The constant C is calibrated by the manufacturer using measurements made on surfaces of known temperature and emissivity, which is not without challenges.

The main issue with measurements by IR temperature detectors is that their accuracy depends on proper compensation for the measured surface's emissivity. Not all devices include options to compensate for specific types of surfaces [90]. Lack of compensation can lead to significant divergences in result accuracy [89].

2.8.1.3 Resistance Temperature Detector

RTDs are resistive elements with a known relation between resistance and temperature. This relation function is called the Callendar-Van Dusen equation (CVD) and is given by equation (9), defined by parts.

$$R_{\mathrm{RTD}}(T)\begin{cases} R_{\mathrm{RTD}}(0)\cdot(1+AT+BT^2) & \text{if } T \geq 0 \\ [1+AT+BT^2+C\cdot(T-100)T^3] & \text{if } T<0 \end{cases} \tag{9}$$

where $R_{\mathrm{RTD}}(T)$ is the resistance of the device, in Ohms, at temperature, T, in °C. A, B, C are coefficients with values $A = 3.908 \times 10^{-3}$, $B = -5.775 \times 10^{-7}$, $C = -4.183 \times 10^{-12}$ for platinum RTDs. $R_{\mathrm{RTD}}(0) = 1000\ \Omega$ for a pt1000 RTD, and $100\ \ \Omega$ for a pt100. Typical application circuits use ratio-metric measurements of voltage across the RTD referenced to a voltage generated by a high resistance that matches the expected maximum temperature of the system. The reference resistor is placed in series with the RTD to ensure either component receives the same amount of current [91], [92].

2.8.1.4 Thermocouples

Thermocouples are devices created by bonding two dissimilar metals at two points, one "hot" junction, with which measurements are made, and the other, "cold". Due to the Seebeck effect, when a temperature difference occurs between these metals, an electromotive force is generated, leading to a measurable change in potential. It is important to note that this potential difference is often very small, which leads to the need for proper signal filtering and an amplifier circuit (i.e. the highest Seebeck Coefficient at 25°C is around 62 μV/K for type E thermocouples). Additionally, these often require compensation of the cold junction to measure temperatures close to ambient, as the cold junction is typically at ambient temperature. These also have non-linear responses, such that conversion tables are required to obtain a temperature from readings [93]. Thermocouples are also prone to oxidation and contamination which affects their accuracy, such that they often need a protective enclosure. Due to all these challenges, thermocouples are best suited in the measurement of extreme temperatures, in controlled environments [94]. Thermocouples come in many different types, each with their own recommended measurement range. These are listed in Table 2.3.

Table 2.3 - Thermocouple Types and Recommended Temperature Ranges [94]

Type	Range (°C)	Notes
B	1370 to 1700	Easily contaminated.
C	1650 to 2315	No oxidation resistance.
E	95 to 900	Resistant to oxidation at cryogenic temperatures.
J	95 to 760	Iron leg is subject to oxidation.
K	95 to 1260	Resistant to oxidation.
N	650 to 1260	General use, better oxidation and sulfur resistance than Type K.
R	870 to 1450	Easily contaminated. Recommended for oxidizing atmosphere.
S	980 to 1450	Easily contaminated. Highly reproducible, laboratory standard.
T	-200 to 350	Stable at cryogenic temperatures. Excellent in oxidizing and reducing atmospheres within temperature range.

2.8.2 Position, Velocity, and Acceleration Sensors

Position, velocity and acceleration sensors can all be used to measure machine vibrations. Position sensors are specifically useful for measuring shaft parallel misalignment or eccentricity. These measurements are conducted at low rotational speeds [95]. Position sensors are useful for monitoring relatively low vibration frequencies, as displacement are larger at low frequencies, leading to a strong signal (large SNR). At high frequencies, accelerometers are preferred, as displacements are very small, but accelerations are large. Velocity sensors are typically implemented by combining an accelerometer and an integrator. Their performance is adequate regardless of frequency [96].

2.8.3 Microphones

Different microphones are available for machine condition monitoring, though recent commercial systems have used digital MEMS microphones as they are less expensive and much simpler to implement [97]. Microphones work in similar ways, despite the output type. The deformation of a membrane excitable by sound pressure is used to generate a change in capacitance in a system. This change in capacitance translates to a change in voltage at the membrane. The voltage output is then amplified within the microphone, and output to an external component, or to other hardware within the microphone, depending on the output type.

The microphone outputs can be of analog or digital types. In analog type microphones, a suitable external audio ADC must be included in the system to convert the signal. In digital microphones, the voltage is oversampled within the microphone by a large factor but only at a single bit of resolution, and the output of the microphone is either the raw oversampled signal (Pulse Density Modulation) or a decimated output (Inter-Integrated Sound) [98], [99], [100]. Separate hardware is required to interpret the output of digital microphones, but this hardware is common to many microcontrollers.

2.8.4 Fluxgate

Fluxgate (flux) sensors are used to measure a magnetic field's strength. They work by using an electromagnet through which an oscillating voltage is applied, generating an oscillating magnetic field. An external magnetic field can then disrupt this field, generating a voltage in the electromagnet that affects the drive voltage. The discrepancy between the drive and measured electromagnet voltages can then be translated to an external field strength reading [101].

Flux sensors have recently been used to measure stray flux from induction motors used in industry. The measured signal can be used to determine the true motor rotational speed of a motor [102]. Other works have used flux sensors to diagnose stator winding faults and rotor faults (referred to as bar faults in squirrel-cage motors). Such faults can cause excessive vibration and temperatures in equipment [9].

Similarly to what was done in subsection 2.1.1, it is important to know how fast the signal changes for induction motor fields to inform the requirements for flux sensors. In the case of motors connected to the mains, where a motor is driven without the use of a variable frequency drive (VFD), the frequency of the magnetic poles changing is that of the mains, 50 or 60 Hz. In the case where a motor is connected to a VFD, the output field frequency can be anything up to 600 Hz. VFDs are not designed to output a power switching frequency above 600 Hz (typically 590 Hz) due to export regulations that are part of the nuclear non-proliferation treaties [103].

2.9 Enclosure Design

There exist 7 levels of enclosures, ranging from semiconductors (i.e. an unpackaged semiconductor) to environment enclosures (i.e. server racks, transformer boxes) [104]. In this work, the term "enclosure" is used in the general sense and refers to the housing used for the MachMoS device. This enclosure is a level 4 enclosure, and its design should consider how the various PCBs in the enclosure will be connected and located within the enclosure. Many considerations for thermal management and interconnectivity are made for level 4 enclosures. It is recommended that natural convection only be used in applications where the electronic device dissipates less than 3 W. Otherwise, forced air cooling might be required [105].

It is very important to consider Design for Manufacturing and Assembly (DFMA) when designing an enclosure, not only as a cost-limiting measure, but as a means to force the justification of each part in a design, to ensure assembly is possible using simple tools, and to ensure the design can be manufactured using processing technologies which are readily available to the manufacturer [106]. Selecting the materials and processes to be used in the fabrication of a product is an important first step in DFMA, followed by part and assembly modeling.

Design considerations for enclosures to be used in vibration monitoring applications include:

- Ensuring that a Helmholtz resonator is not created when an orifice in an enclosure is created for a microphone, as suggested by TDK InvenSense [107], and
- Ensuring that the lowest resonant frequency (mode) of the enclosure is outside of the accelerometer band, as outlined by Analog Devices [108].

To prevent Helmholtz resonators that might introduce noise in the microphone signal, the highest resonating frequency of the enclosure must be outside of the microphone band. To calculate the resonating frequency of a resonator, geometry of the orifice of the cavity must be considered [107]. The resonant modes of an enclosure can be determined by performing a modal analysis (calculations or numeric simulations) [108].

Additive manufacturing has increasingly been used in educational and research institutions worldwide [109]. As such, it is reasonable to assume that Fused Filament Deposition machines are available for use at research institutions to produce various components for the MachMoS device. Design for Additive Manufacturing guidelines have been published in literature [110], and the use of additive manufacturing for electronics enclosure manufacturing through the careful consideration of the physical processes that occur during printing (shrinking, warping, etc.) has been studied by Selivanov, K. et al. [111]. Waterproofing printed enclosures has also been shown to be successful in literature [112], [113], [114], but the level of protection (ingress protection rating) is not provided by those works.

2.9.1 Ingress Protection Ratings

Ingress Protection (IP) ratings are a set of ratings attributed to devices that inform users on their ability to protect against ingress of dust or water. These ratings and the testing procedures that devices must pass are found in the International Electrotechnical Commission (IEC) 60529 standard. This standard is well recognised and has seen wide industry adoption. In their most well-known configuration, IP ratings are given as a set of two numbers, though a third number, as well as supplementary letters can be used to convey more information about a device's protection against ingress.

The first numeral in their rating system represents a device's level of protection against ingress of solid objects (and particles), and the second represents the level of protection against ingress of fluids. If a device is not tested for solid or liquid ingress, an 'X' is used instead of a number. Table 2.4 provides a list of ratings as well as their respective meanings. It is important to note that the table provided is specific to protection against ingress of objects and fluids. Other tables may describe protection against exposure specifically to hazards within the device.

Table 2.4 - IP Rating Numbering Scheme [115]

	First Numeral (IP from Solids)		Second Numeral (IP from Fluids)	
Num.	**Protection**	**Description**	**Protection**	**Description**
X	Not tested	Denotes that the enclosure is not tested for IP from solids.	Not tested	Denotes that the enclosure is not tested for IP from fluids.
0	-	No protection against ingress of objects.	-	No protection against ingress of water.
1	> 50 mm	An access probe (⌀50 mm sphere) shall not fully penetrate.	Vertical water drops	Vertical falling drops shall have no harmful effect.
2	> 12.5 mm	An access probe (⌀12.5 mm sphere) shall not fully penetrate.	Drops at 15°	Vertically falling drops shall have no effect when enclosure is tilted at any angle up to 15° from the vertical.
3	> 2.5 mm	An access probe (⌀2.5 mm sphere) shall not penetrate at all.	Spraying water	Water sprayed at an angle up to 60° on either side of the vertical shall have no harmful effects.
4	> 1 mm	An access probe (⌀1.0 mm sphere) shall not penetrate at all.	Splashing water	Water splashed against the enclosure from any direction shall have no harmful effects.
5	Dust protected	Ingress of dust not fully prevented but shall not penetrate in a quantity to interfere with satisfactory operation of the apparatus or to impair safety.	Water jets	Water projected in jets against the enclosure from any direction shall have no harmful effects.
6	Dust tight	No ingress of dust.	Powerful water jets	Water projected in powerful jets against the enclosure from any direction shall have no harmful effects.
7	*Not applicable to IP from Solids*		Temporary immersion	Ingress of water in quantities causing harmful effects shall not be possible when the enclosure is temporarily immersed in water under standardized conditions of pressure and time.
8			Continuous immersion	Ingress of water in quantities causing harmful effects shall not be possible when the enclosure is continuously immersed in water under conditions which shall be agreed between manufacturer and user, but which are more severe than for numeral 7.
9			Powerful high-temperature water jets	Protected against close-range high-pressure, high-temperature spray downs. Specimens are mounted in the intended position when used and are tested freehand for at least 3 minutes at a distance of 0.15–0.2 meters (5.9 in – 7.9 in).

2.10 IoT Networks

An internet of things (IoT) network is a network of distributed information and communication technology. This technology can include sensors, computing devices, algorithms, and physical objects, which can collect and transfer data without human intervention. These make up the *things* that are individually identifiable in the network [116]. Many IoT systems comprise sensors connected to a gateway that transmits data to the cloud. Various wireless communication protocols are used in IoT networks, such as cellular for high transmit rate, long distance communication, Bluetooth® and Bluetooth Low Energy® (BLE) for low-range moderately high transfer rates, IEEE 802.11 (Wi-Fi™) for low range and high transfer rates, Low-Power Wide Area Network (LPWAN) and Low Range Wide Area Network (LoRaWAN) for very long range but low transfer rates [117].

Most IoT networks will use a mix of communication protocols in their implementation, with battery-powered *things* using low-power protocols and wired *things* using high bandwidth protocols which require more power. IoT wireless sensor networks (WSN) are of particular interest to this work. These networks consist of sensor nodes capturing data and transmitting it to a gateway that then uploads that data to the internet [118]. In WSNs the gateway is not battery powered, as it manages multiple data streams and must always remain on, ready to receive data from the downstream components of the network.

Several IoT networking protocols support meshing of devices. Meshing allows data to move from one node to another until the data reaches the gateway of the network. In an sparse IoT network arrangement where a single gateway might be present, meshing can extend the effective range of nodes in the network [119], [120].

2.11 Electro Magnetic Interference (EMI)

The MachMoS node is a Class A digital device under United States 47 CFR part 15 [121] and under Canadian regulations set by the Interference-Causing Equipment Standard (ICES). It also falls under the Canadian Radio Standards Specification (RSS), which means that it is subject to emission limits. While this device is an intentional radiator (which falls under 47 CFR part 15 and RSS), all electronic devices are also unintentional emitters (47 CFR part 15 and ICES). For instance, a SMPS or a communications protocol in the device might be switching (square wave) at high frequency, in which case, the device may radiate electro magnetic energy at this frequency and above (as square waves are formed from multiple higher frequency harmonics), above legal limits. The power of the unintentional emissions can be mitigated in the design stages, to avoid being above the limits set in the regulations. ICES-003 [122] is the specific Canadian regulation that should be used for the MachMoS device. The emission limits for class A devices listed in the standard are listed per frequency band and are given in Table 2.5. It should be noted that the 47 CFR part 15 are converted from µV/m to dB·µV/m (the units provided by ICES-003) and show that the limits are unsurprisingly the same in both standards.

Table 2.5 - Radiated Emissions Limits for Unintentional Radiators at 10 meters.

	47 CFR part 15 [121]	ICES-003 [122]
Frequency Band (MHz)	**Field Strength (dB·µV/m)**	**Field Strength (dB·µV/m)**
30 to 88	39.08	39
88 to 216	43.52	43.5
216 to 960	46.44	46.4
> 960	49.54	49.5, 69.6 pk.*

* Peak applies beyond 1 GHz. Testing shall be conducted with a linear range average detector as defined in IEC CISPR 16-1-1

As outlined in either document, if radiated emissions testing is performed at a distance other than 10 m, the value shall be extrapolated to 10 m using a factor of 20 dB/decade.

In the case of intentional radiators, the tests are conducted at 3m as opposed to 10m. The limits are provided in Table 2.6. Limits are also provided for frequencies below 30 MHz, but the MachMoS device will likely not intentionally emit at such low frequencies (30 MHz is the top end of the spectrum for shortwave radio). Finally, multiple bands are restricted in the listed ranges, in which intentional radiators may not operate. These are provided in Appendix C.1.

Table 2.6 - Radiated Emissions Limits for Intentional Radiators at 3 meters.

	47 CFR part 15 [123]	RSS [124]
Frequency Band (MHz)	**Field Strength (µV/m)**	**Field Strength (µV/m)**
30 to 88	100	100
88 to 216	150	150
216 to 960	200	200
> 960	500	500

2.12 Microcontrollers

Microcontroller Units (MCUs) are very basic computers which are implemented on a single IC [125]. At a minimum, they contain a central processing unit (CPU), some memory, and input/output devices. A system on a chip (SoC) MCU is an MCU that also includes other functionalities, such as Bluetooth® wireless communications. SoC are well-suited for data acquisition as they can interface with multiple sensors and transfer that information to larger systems (i.e. a personal computer or the cloud).

A CPU is used for computations, facilitates the logical flow of information, and processes input/output requests. The operations that may be performed by a CPU are described by instruction codes. The instruction codes make up an instruction set. The smaller the instruction set, the cheaper the microcontroller. The reduced instruction set computer (RISC) can perform a highly optimized set of instructions, leading to reduced cost and high speeds (working off the principle that the smaller instruction set may not do as much, but can run at very high speeds) [126].

The memory in microcontrollers is electrically erasable read-only memory (typically flash memory) and random-access memory (RAM) [125]. Flash memory is not erased when a circuit is power cycled, but RAM is. As such, the instructions that define a microcontroller's program are stored in flash (and is loaded to RAM on power-up). The MCU uses RAM to store all program variables.

2.13 Voltage Regulators

Sensors and microcontrollers are often rated for operation using a specific input voltage. Most recently, 3.3 V (3v3) is used for microcontrollers and their logic. However, 5 V remains a common voltage and is the bus voltage for common protocols, such as USB 2.0 and USB 3.0 [127], [128]. It is important to regulate source voltages for devices before they are routed to other devices. This is achieved using voltage regulators. Among the common types are the linear regulator (specifically low dropout regulators, LDO) and the switch-mode power supply (SMPS) [129].

LDOs can only be used to lower the voltage for a circuit and do so by releasing the excess energy at the input as heat. As such, the larger the reduction in voltage from input to output, the more heat is generated and the less efficient the regulator is. For instance, a 5 V to 3.3 V LDO can 66 % efficient at most, as 1.7 V (34 %) is dropped from the source to obtain the output voltage.

Additionally, LDOs require the source to be at a voltage that is slightly higher than the output. This difference in voltage is called the drop-out voltage [129].

A SMPS uses a switch node, an inductor, and a diode to increase (boost) or reduce (buck) voltage from the input to the output. When an inductor goes from a charged state, in which it contains energy in the form of a magnetic field, to an off state, its magnetic field collapses and is converted to a large electric field of inverse polarity (voltage spike). The voltage spikes are filtered by a capacitor parallel to the load. Depending on the placement of the diode, inductor, and switch node in the circuit, the regulator circuit can easily change from a boost to a buck circuit [129]. Some SMPS ICs are designed in such a way that they can be used as buck-boost regulators, which allow for the increase or decrease of the input voltage without any changes to the circuitry [130], [131].

2.14 Power Sources

The energy sources used for condition monitoring devices can be placed into three categories: wired, self-powered, and harvested.

2.14.1 Wired Devices

Wired devices consist of wiring a device to an Alternating Current (AC) or a Direct Current (DC) source near-by, using a power cable. A wired device is the simplest to implement in terms of device design, as the device is assumed to have an unlimited supply of energy. In practice, however, one finds that it can be very difficult to find an available power socket or device which can provide power to the sensor. Power sources feeding into large machinery are often multi-phase, and devices that exist close to machinery (variable frequency drives (VFDs), current sensors, etc.), are not equipped with output jacks that could be leveraged for wired power to a sensor.

2.14.2 Self-Powered Devices

Self-powered devices consist of devices which do not require outside power sources to operate. These devices use on-board batteries for power. Batteries are categorized into two categories: primary and secondary. Primary cells are non-rechargeable and must be replaced once discharged. Secondary cells, on the other hand, are rechargeable, making them more sustainable, but require a special charging circuit. There exist many chemistries for batteries, each with their advantages and drawbacks. The following sections will cover some of the chemistries relevant to industrial use.

2.14.2.1 Primary Cell: Lithium-Thionyl Chloride (Li-$SOCl_2$)

Li-$SOCl_2$ batteries are common in self-powered industrial applications. This battery chemistry features high energy density (650 Wh/kg [132]) and outputs a very stable voltage (3.6 V) over its useful life, making it well-suited for compact, low-cost applications where a regulator is not desired. The chemistry is used to produce cells that exhibit very low self-discharge (1 % per year for the *Saft* LS 26500 cells). This chemistry is well-suited for low discharge rates (under 500 mA per cell) [133]. The recommended temperature range for this chemistry is -55°C to 80°C, but some cells can be designed to withstand temperatures of up to 165°C [134], [135].

2.14.2.2 Primary Cell: Lithium Metal (Li-FeS_2)

Li-FeS_2 cells have moderately high energy densities (300 Wh/kg [136]) that are well-suited for applications with low discharge rates. The recommended temperature range for a cell of this chemistry is -40°C to 60°C. The cells do not exhibit a consistent voltage drop across their discharge cycle, and have a relatively stable output voltage throughout their cycle [137].

2.14.2.3 Secondary Cell: Lithium-Ion (Li-Ion)

Secondary Li-Ion cells have become very common in recent years, with a few common cathode chemistries: $LiCoO_2$ (LCO), $LiNiCoAlO_2$ (NCA), $LiNiMnCoO_2$ (NMC), $LiMn_2O_4$ (LMO), $LiFePO_4$ (LFP). Li-Ion cells are popular due to their high energy density in a rechargeable package (170 Wh/kg to 200 Wh/kg), with continuous research into cell construction to increase density to 500 Wh/kg [138]. Li-Ion cell chemistry is well-suited for very high discharge rates, reaching up to 30 A per cell. Secondary lithium cells exhibit a consistent voltage drop over their discharge cycle [139]. Li-Ion batteries are recommended for use in very narrow temperature ranges: 0°C to 45°C, with degraded performance when operated outside of the 15°C to 35°C range [140].

2.14.2.4 Secondary Cell: Nickel Metal Hydride

Nickel Metal Hydride (NiMH) batteries are very common and are well-suited for applications where a moderately large current draw is required. They have been criticized for their high self-discharge, but quality cells rated for “low self-discharge” such as the *Panasonic* Eneloop cells have been available on the market for many years and have similar self-discharge ratings to Li-$SOCl_2$ cells. A detailed reference discharge curve for an Eneloop low self-discharge cell is provided in Figure 2.1. The curve was generated by discharging two distinct cells (A and B) at various rates. The discharge and monitoring setup is provided by reference [141]. As can be seen in the figure, the voltage of the cells does not drop consistently over the life of the cells. A *Panasonic* Eneloop battery has an operational temperature range between -20°C and 50°C [142], [143], with some wide-temperature batteries ranging -40°C to 85°C [144].

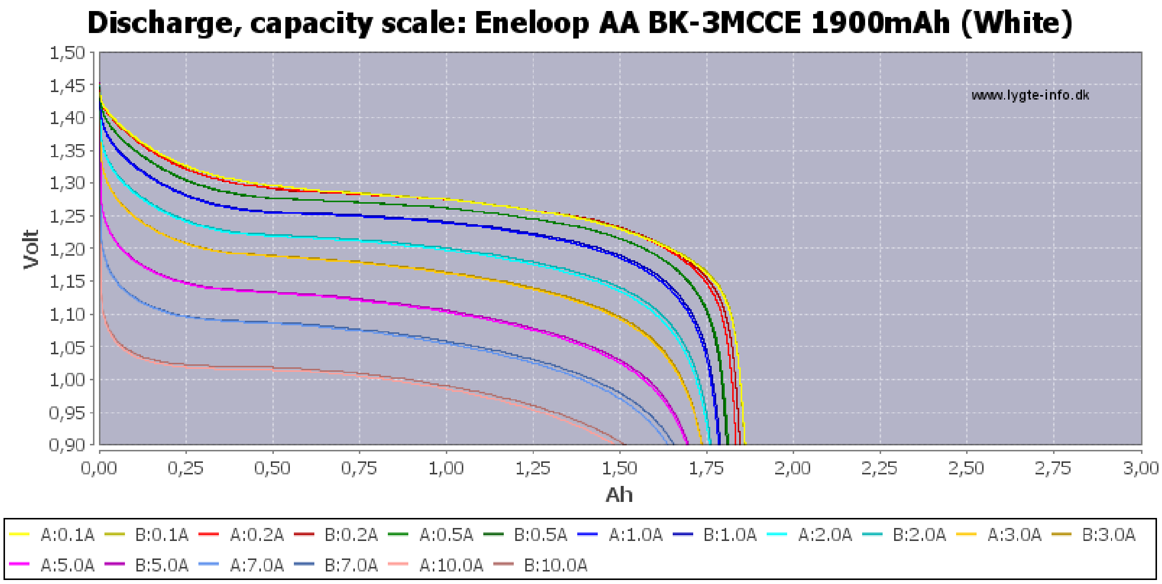


Figure 2.1 - Reference Discharge Curve for a NiMH Cell [141]

2.14.2.5 Estimating Remaining Battery Charge

The remaining battery charge can be difficult to obtain from directly observable metrics such as battery voltage (using an ADC) [145]. This is due to the inconsistent remaining charge to voltage relation for different chemistries and implementations. A more holistic approach which tracks voltage, temperature, and current draw must be used to accurately obtain battery remaining charge, factors that affect the charge to voltage relationship of a battery [146]. Devices that track all this data are known as Battery Fuel Gauges and are tuned to different cell chemistries. Fuel gauges are extremely useful when measuring the remaining capacity of cells with very stable voltages, such as Lithium-Thionyl Chloride cells. In the case of other chemistries with relatively variable voltages, an ADC can be used to estimate the remaining battery voltage depending on the chemistry of the battery, the average current draw of the application, and the average temperature of the battery. Other solutions include mathematical modeling and the measurement of device on time to provide an estimate of remaining battery life.

2.14.3 Energy Harvesting

The use of energy harvesting in IoT applications, specifically WSNs, has been widely researched [118], [147]. As sensors only use a bit of energy at specific times and are on standby the rest of the time, it makes a lot of sense to use energy harvesting to provide a small amount of charge to the device over time. Energy harvesting can take many forms, but only some are applicable to industrial settings. Table 2.7 lists energy harvesting techniques which are applicable to industrial settings and provides advantages and disadvantages for each. As can be seen in the table, the highest energy densities are attainable using solar energy, but only if used outdoors. RF energy sourcing, such as using energy on the 2.4 GHz and 5 GHz bands from Wi-Fi communications or even that of cellular networks (most recently, in the GHz range), can provide up to 0.1 μW/cm^2, which translates to 360 joules generated per hour using a 1 cm^2 antenna. As for mechanical methods (from vibrations and pressure), they provide more energy density, but the methods suffer from unpredictability. In the field of MCM, the worse the machine health, the more energy is available for harvesting. A good PHM system would trigger alarms and maintenance action would ensue. Once maintained, the quantity of energy available for harvesting would once again be low.

Table 2.7 - Energy Harvesting Techniques. Reproduced from [148]

Source	Technology	Power Density	Advantages	Disadvantages
Solar	Photovoltaic Cell (PV)	**Outdoors:** 10-100 mW/cm^2 **Indoors:** <100 μW/cm^2	- Relatively high output voltage - Low fabrication cost - Predictable	- Unavailable at night when outdoors or if panels are covered. - Non-controllable
RF	Antenna	0.01-0.1 μW/cm^2	- Widely available - Predictable - Controllable	- Distance-dependent - Low power density
Mechanical Vibrations and Pressure	Piezoelectric	4-250 μW/cm^3	- High power density - No external voltage source - Simplicity in design and fabrication - Controllable	- Highly variable output - Unpredictable
	Electromagnetic	300-800 μW/cm^3	- High output currents - Robust - Low-cost - Controllable	- Relatively large size - Unpredictable
	Electrostatic	50-100 μW/cm^3	- High output voltage - Relatively larger output power density - Possibility to design low-cost devices. - Controllable	- Requires bias voltage. - Unpredictable

Chapter 3 Methodology

3.1 Requirements, Benchmarking, and Specifications

Before designing a product, it is important to understand its requirements. Without clear requirements, the development of a product can veer off-track or become too complex. From these requirements, one can then develop specifications that the product must meet. Reasonable values for the specifications can then be obtained by benchmarking existing products.

3.1.1 Requirements

An industry supplier of bearings and industrial machinery was approached and interviewed to determine the general requirements for an ideal vibration monitoring device. These requirements were further refined over the course of the project through a collaboration between the industrial supplier and the DDL. It was determined that the device should meet the following requirements:

- be affordable (material cost in the range of \$20 to \$100 CAD).
- be simple to set up, use, and reconfigure.
- be highly sensitive to equipment vibrations, sounds, and changes in temperature.
- allow for mounting with a screw, adhesive, or magnetic base.
- allow for the greasing of machinery without removing the device, even if screwed onto a grease fitting port. This feature is not available on most commercial devices.
- provide suitable ingress protection for outdoor or harsh environment use.
- function in both hot and cold environments.
- work without Wi-Fi™ coverage.
- be battery powered (self powered).
- have long operational life.
- be rugged.

The following requirements were then added after further discussions within the DDL:

- be manufacturable by most machine shops.
- be rechargeable or have a replaceable energy source.
- allow for the addition of other (external) sensors.
- allow for over-the-air updates to firmware.
- perform simultaneous sensor measurements to facilitate sensor fusion.

From the list of requirements, the following metrics were identified: device purchase cost, device battery life, IP rating, temperature range, full scale measurement range and sampling rate. Given that the device is a wireless device, the line of sight (LoS) range of the wireless connection was also listed as a metric.

3.1.2 Benchmarking

Benchmarking was performed on a set of commercial sensors to determine specifications for the MachMoS device. Battery-powered wireless sensors from *PCB Piezotronics*, *Nanoprecise*, *Treon*, *Bently Nevada*, *Erbessd Instruments*, and *NCD.io* were used in the benchmarking process. These were compared against the requirement set in subsection 3.1.1 and shown in Table 3.1, with shading applied to the most desirable specification (dark shading with white text is best, grey shading is marginal, unshaded is not desirable). Rows that do not have any shading are provided for informational purposes only.

It should be noted that some of the metrics in the benchmarking process were difficult to compare, such as the battery life of the device and the maximum range and resolution. The battery life of a device, as specified by many companies, highly dependent on user settings. A metric based on the number of samples captured could have led to a proper comparison, but this information was only provided for the Phantom Expert Gen 3.

Battery life also depends on the implementation of wireless communications by the manufacturer, as well as the data being transmitted (i.e. raw time series data vs. key performance indicators (KPI), such as root-mean squared acceleration data (RMS) or FFTs). The Ranger Pro and Vibration Sensor Gen 4 were attributed the best score on this metric given that their large battery capacity could indeed lead to the listed life considering similar configurations to those of the better-documented Machine Doctor or Phantom Expert devices. The *Treon* Industrial Node 6 has neither sufficient information on configuration nor a battery large enough to yield a 5-year life using a similar battery life as that of the Machine Doctor, especially if sending raw data as opposed to features.

Table 3.1 - Benchmarking

	Echo Wireless (TO670A01) [149]	**Machine Doctor WiFi (NS002)** [150], [151], [152], [153], [154]	**Industrial Node 6 (2111)** [155], [156]	**Ranger Pro (70M303)** [157], [158]	**Phantom Expert Gen 3 (EPH-V11E)** [159], [160], [161]	**Smart Vib. Sens. Gen 4 (PR63-3A)** [162], [163]
Company	*PCB Piezotronics*	*Nano Precise Sci*	*Treon*	*Bently Nevada*	*Erbessd Instruments*	*NCD.io*
Price [CAD]	3500	966	937	1518	595	388
Outputs	Accel. (Z), Temp.	Accel. (X, Y, Z), Temp., Amb. Sound, Humidity, Mag. Flux, RPM	Accel. (X, Y, Z), Temp.	Accel. (X, Y, Z), Temp.	Accel. (X, Y, Z)	Accel. (X, Y, Z), Temp., Mag. Flux
Output Type	RMS	Raw or Features	Raw or Features	RMS	RMS	Raw or Features
Max Accel. Sampling [kHz]	61.4	24.0	26.7	20.0	25.6	25.6
Rated Accel. Rate (-3 dB)	15.0 kHz	8.0 kHz (X, Y) 5.1 kHz (Z)	6.3 kHz (X, Y, Z)	10.0 kHz (Z) 4.0 kHz (X, Y)	10.0 kHz (X, Y) 5.1 kHz (Z)	**
Battery Life	1 yr. at 3 meas./day	1.5 yrs. At 4 meas./day	5 yrs.	5 yrs.	4 yrs. at 24 meas./day	5 yrs.
Battery Type, Size, Capacity	Primary (Li-$SOCl_2$) 2x AA, 3.6 Ah	Primary (Li-FeS_2) 1x CR123A, 1.55 Ah	Primary (Li-$SOCl_2$) est.: 1x AA, 1.8 Ah	Primary (Li-$SOCl_2$) 1x D, 19.0 Ah	Primary (Li-FeS_2) 1x CR2477, 1.1 Ah	Primary (Li-$SOCl_2$) 1x D, 13.0 Ah
Mounting	Stud, Magnet, Adhesive (Epoxy)					
IP Rating	IP66	IP68	IP67	IP67	IP69	IP67
Temp. Range [°C]	-20 to 70	-40 to 120	-40 to 85	-40 to 85	-40 to 80	-40 to 70
Max. Range [g], (Resolution [mg])	±20 (7) 16-bit	±40 (0.031) 21-bit	±16 (1) 16-bit	±20 (*) *	±32 (2) 16-bit	±16 (1) 16-bit
Range (LoS), Radio Freq.	1600 m, 916 MHz	50 m, 2.4 GHz	60 m, 2.4 GHz	150 m, 2.45 GHz	100 m, 2.4 GHz	3200 m < 900 MHz 90 m at 2.4 GHz
Protocol	LoRaWAN®	WiFi™	Wirepas Mesh®	Wireless-HART®	BLE 5.0	Digi XBee/Mesh®

* Data transmitted to the gateway is 32-bit, but the accelerometer resolution is unspecified [158].

** Unspecified.

The full-scale measurement range of each device's sensors is also highly related to resolution, and these are not typically specified. Estimated bit-counts (obtained from measurement range and resolution) are therefore included in the benchmarking table to assist in comparing the devices. Finally, the cost of the *PCB Piezotronics* device should be addressed. This cost is over twice that of the *Bently Nevada* device. This is most likely attributable to the company's past of manufacturing wired devices with piezo-ceramic elements for vibration sensing [164], [165]. Without changing the sensing element, devices using piezo-ceramic elements are very difficult to convert to wireless, as the source voltage for exciting these elements is high (typically 20 V), and the return signal is rather weak (100 mV/g). This requires expensive signal conditioning hardware as well as highly precise ADCs. Of the benchmarked devices, the *PCB Piezotronics* sensor is the only one that uses a piezo-ceramic accelerometer rather than a MEMS accelerometer. This demonstrates the cost savings related to the use of MEMS accelerometers.

3.1.3 Target Specifications

Target specifications were generated from the benchmarking exercise outlined in subsection 3.1.2 as well as the requirements and literature review. The specifications are as follows:

Based on Metrics

S.1 The accelerometer -3 dB rate ($f_{\max}$) must be at least 6.3 kHz. Ideally, 10 kHz.
S.2 The battery life must be at least 1 year, ideally 5 years.
S.3 The IP rating shall be at least IP66, ideally IP68 or IP69.
S.4 The operational temperature range must be at least -40°C to 85°C, ideally 120°C.
S.5 The temperature sensor in the device shall be accurate to ±0.5°C.
S.6 The full-scale range must be at least ±16g.
S.7 The data resolution must be at least 16-bit.
S.8 The LoS wireless range of the device must be at least 100 m (ideally 150 m). If its radio operates over 1 GHz, 1.6 km (ideally 3.2 km).

Based on Qualitative Properties

S.9 Maintain reasonably low material costs to build an open-access device that can lead to the widespread distribution of open-source data. A purchase cost of $ 350 CAD is considered reasonable and competitive.
S.10 The node must feature the following sensors:
    S.10.a Tri-axial accelerometer,
    S.10.b Ambient sound (microphone), and
    S.10.c Temperature sensor.

It would ideally also feature the following:

S.10.d Magnetic flux sensor,
S.10.e Hygrometer,
S.10.f RPM sensor.

S.11 The node device must output raw time series data. Ideally, it would also be able to transmit features or KPIs such as RMS, FFTs.
S.12 The battery in the device must be rechargeable (secondary type). The charge circuitry shall be contained in the device.
S.13 The node shall be easy to install:

S.13.a The mounting method for the node device must be either by stud or magnet. It can be epoxy based if stud or magnet mounting is not possible.
S.13.b The node shall be designed such that the node must not be removed for routine maintenance (i.e. greasing of the bearing).

S.14 The node must be able to communicate wirelessly.

3.1.4 Manufacturability

It is important that the device enclosure be manufacturable in most research institutions. This consideration was kept in mind during the design process. The following DFMA rules were applied in the initial design stages of the sensor:

For milled components, the parts shall have:

- internal geometries (pockets, slots) larger than .125 in.
- only standard-sized tapped (UNC or UNF threads) or drilled holes (letter and American Wire Gauge numbered drills or metric-sized drills, 0.5 mm increments).
- blind-tapped holes that have threads no deeper than 2 diameters to the bottom (bottoming).
- holes that are no deeper than 10x their diameter.
- surface finishes that are coarser than Roughness average (R_a) 63 µin.
- **no** undercut geometries that would require more than 3 axes of movement.
- **no** features that require fits tighter than 5 tenths of a thousandth of an inch. If any tighter fits are required, it shall be a standard fractional size that can be reamed.
- **no** more than 2 setups per part (preferred).

For turned components, if there are any external threads, the thread shall have a gutter large enough for manual machining. Additionally, the parts shall have:

- surface finishes that are coarser than R_a 32 μin.
- outer diameters that can be held in standard-sized collets.
- **no** features that require fits tighter than 2 tenths of a thousandth of an inch.
- **no** diameters larger than 1.5 inches to minimize parting tool stick-out for production.

Other processes such as sanding, low-pressure forging or pressing (using a hand-operated arbor press), hand tapping, were all considered acceptable. The printed circuit boards in the device did not fall under the DFMA considerations above, since prototype printed circuit boards from companies such as *JLCPCB* (China), *PCB Way* (China), and even *OSH Park* (United States) have become very cost-effective and accessible. The DFMA rules obtained from those manufacturers were applied during the printed circuit board layout stage.

Assembly rules were also considered. Manual assembly and disassembly using a minimal number of tools was prioritized. To reduce assembly complexity and potential errors, screws with a uniform head type and drive size were selected for mounting both the PCB and enclosure components. The use of adhesives to combine enclosure components was deliberately avoided to facilitate disassembly and maintenance.

3.2 Experimental Setup

The following subsections discuss the testing methodology used to evaluate the performance of the MachMoS node against the target specifications. In this section, the node is referred to as the device under test (DUT).

3.2.1 Temperature Sensor

The temperature sensor was initially standardized by submerging the MachMoS node in an ice bath at 0°C. This approach was unsuccessful as the bath did not have the capacity to cool the device to 0°C. To enable sufficient cooling, the DUT was partially submerged in a bowl of ice water, and the bowl was placed in a freezer at -16°C. Using a wired connection, a computer was used to capture the temperature measurements from the DUT at a rate of 10 Hz. The test was considered successful if the temperature measured by the DUT plateaued at 0.0 ± 0.5°C as the water froze. As water cools, its temperature decreases until it reaches its transition temperature of 0°C, at which solidification occurs. At this temperature, it releases its latent heat, and the phase change is isothermal. Once the phase change is complete, the temperature will decrease as the ice cools. It was assumed that the cooling was slow enough that no supercooling occurred during the phase change.

3.2.2 Accelerometer and Microphone

Per specification S.10.a, the MachMoS node developed in this work was equipped with a microphone and a tri-axis accelerometer. The MachMoS node was therefore tested for its achievable data rate, frequency range (bandwidth), accuracy, and immunity to aliasing.

The achievable data rate of the node was first evaluated by configuring the node's accelerometer and microphone to their maximum output data rate (ODR) and ensuring the measurement data wasn't lost or corrupted during a capture cycle. If data loss or corruption occurred, the ODR was reduced, and the capture cycle repeated until the measurement data for a capture cycle was deemed complete. This procedure was conducted by an embedded software developer employed by the DDL, and not the author. This procedure was not documented, and the maximum achievable data rate remains highly dependent on software implementation.

The bandwidth, accuracy, and aliasing tests were performed using a vibrator/exciter driven by a waveform generator. As the response of the vibrator/exciter was unknown, the accuracy of the node was evaluated by comparing its measurements to baseline measurements. These were obtained by performing measurements, using an off-the-shelf accelerometer and microphone pair. To ensure the MachMoS node and off-the-shelf sensors measured the same sounds and vibrations, the measurements were performed simultaneously on a test jig. This is further explained in subsection 3.2.2.3.

To perform the bandwidth and accuracy tests, the waveform generator was set to frequencies within the node's expected bandwidth (below f_{max}). The voltage and oscillation frequency was recorded at the input of the vibrator/exciter to ensure the test conditions could be repeated. A Python script (Appendix G.2) was then used to simultaneously trigger capture cycles from the MachMoS node and benchmark sensors. The capture cycles lasted 1 s, and the measurement data was saved to a computer. Plots of the data were made to perform comparisons. A test was considered successful if the frequency content of the benchmark measurements and that of the node's showed peaks at the same frequency, and if the amplitude of either device's time series captures were similar. The test setup, shown in Figure 3.1, is further detailed in subsections 3.2.2.1 to 3.2.2.3.

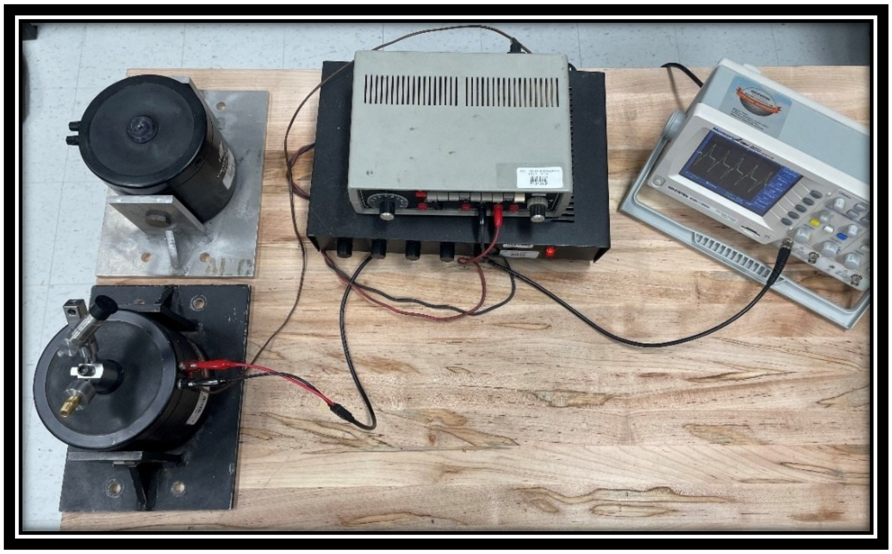


Figure 3.1 - Accelerometer and Microphone Test Setup

Aliasing tests were performed as described above but were conducted at frequencies slightly above $f_{\max}$. These tests were considered successful if the resulting frequency content did not show any folded peaks in the measured frequency range.

3.2.2.1 Sensors for the Baseline Measurements

The benchmarks used in this work are driven by a *PCB Piezotronics* 482C signal conditioner. This signal conditioner is used to excite sensors with the required voltage, and filters the signal returned by the sensors. The signal conditioner outputs an analog signal. As such, it cannot be read directly by a computer. The 16-bit ADC and USB interfaces of a *National Instruments* USB-6212 BNC data acquisition device was used to convert and transmit the signal [166].

The accelerometer used to create the benchmark for these tests was the *PCB Piezotronics* M623C01. When driven by the 482C signal conditioner, the M623C01 has a rated resolution of 100μg, spectral noise of 7μg/√Hz at 10 Hz, and a measurement bandwidth of 0.8Hz to 15kHz [167]. This accelerometer was chosen as these ratings far exceed those of typical MEMS accelerometers. The *Erbessd Instruments* EPH-V11E and *NDC.io* Gen 4 accelerometers have a much higher rated spectral noise of 630μg/√Hz and 70 μg/√Hz, respectively. This is much higher than the 7μg/√Hz of the *PCB Piezotronics* M623C01.

The *PCB Piezotronics* 130F20 microphone was also used in these tests. The microphone was mounted to a tripod which stood off the ground to isolate it from the test jig's vibrations. The 130F20 has a rated bandwidth of 10 Hz to 20 kHz when driven by the 482C signal conditioner [168].

3.2.2.2 Vibration Source

A *Brüel & Kjaer* vibration exciter type 4809, which is rated for oscillating at frequencies ranging from 10 Hz to 20 kHz, was used as a vibration source. To drive the exciter, a *Bogen* Classic C60 Mixer Amplifier was used to amplify signals from a *Keysight* 33500B Waveform Generator. To monitor the signal going to the vibration exciter, a *GW Instek* GDS-1152A oscilloscope was used. A schematic of this configuration is shown in Figure 3.2.

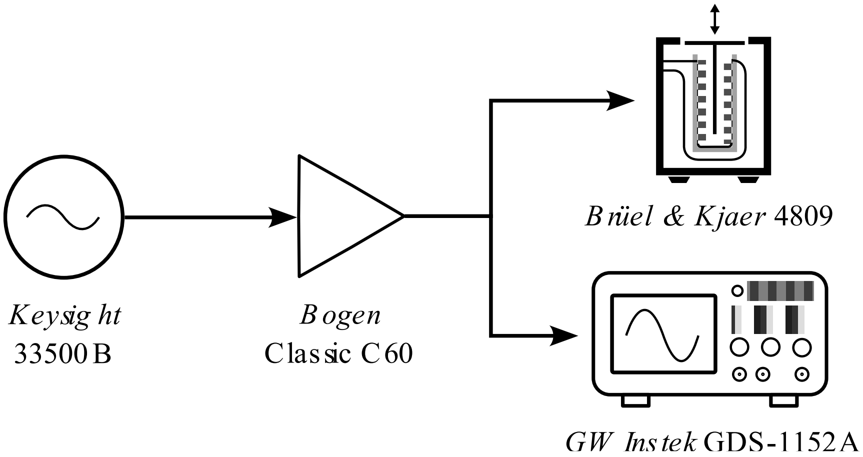


Figure 3.2 - Test Setup Schematic

3.2.2.3 Vibro-Acoustic Test Setup and Jig

As the vibrator/exciter can only vibrate in one direction, the MachMoS node was oriented at a compound angle of 45° around its X-axis and 45° around its Z-axis during testing. This ensured that a measurable vibration signal was obtained on all three axes of the node accelerometer. As the benchmark is a single-axis accelerometer, it was also necessary to align it to any of the three axes of the node's accelerometer. A steel jig was designed and manufactured for this purpose. The full vibro-acoustic test setup is shown in Figure 3.3. In this figure, the benchmark can be seen in its three possible configurations, aligned with the node accelerometer's axes (shown in green).

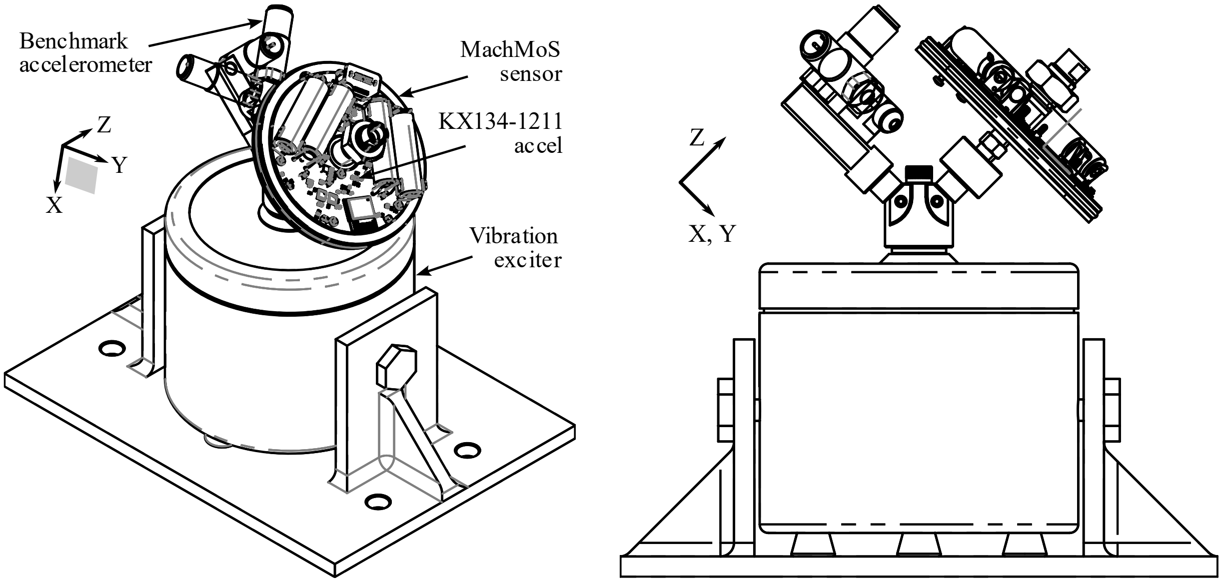


Figure 3.3 - Vibro-Acoustic Test Setup with Accel. Coordinate System Orientation Shown

3.2.3 Ingress Protection

The MachMoS sensor node must achieve an IP rating of IP66 at a minimum, though an IP68 or IP69 rating is preferred, per specification S.3. As previously mentioned, the first numerical in the IP standard represents ingress protection against solids, and the second, ingress protection against liquids. The IPX6 test consists of building a test rig with water jets that can spray the enclosure at various angles. The test for the IPX7 level of protection consists of placing the DUT at a depth of 1 meter under water, for 30 minutes. IPX8 can be obtained if testing is successful at any depth greater than 1 meter.

After testing, the device is opened and inspected for any water ingress. Per the standard, if water is present in the device, the quantity of water that enters the device shall not cause failure of the device. The standard additionally specifies that the test is to be performed with the device in the orientation in which it is to be installed during normal operation [115]. As it is often impractical to install vibration sensors in an upright position, such sensors should be tested in all orientations. If the device meets IPX7 or IPX8 protection, it is assumed to exceed the less stringent requirements of IPX6. The MachMoS node was tested against the IPX8 requirements, at a depth of 1.3 m. This test was conducted in upright and upside-down positions.

The IP6X test for solids infiltration was not performed, as the sensor must be subjected to recirculating talcum powder and a slight negative pressure applied to the enclosure. These test facilities were not available for this project and solids ingress tests are beyond the scope of this work.

3.2.4 Charge Circuitry

After a period of 1 to 5 years of operation (S.2), the battery contained in the MachMoS node must be charged. The battery is to be charged through an external connector. A charge circuit was included in the node. To test the circuit, the device was charged while ensuring the temperature of the battery did not rise above its rated operating temperature. This temperature was obtained from literature on cell chemistry, found in section 2.14.2. Once the charging cycle was complete, the resulting voltage of the battery was also measured and compared to the literature in section 2.14.2. The test was considered successful when the charger reached the end of a charge cycle without triggering faults. It was considered failed if any other scenario occurred (battery isolation from the rest of the device is unconfirmed, temperature too high, battery voltage incorrect, error triggered on the charger side).

3.2.5 Energy Usage

The energy usage of the MachMoS node was tested during three device states: *Acquire, Transmit,* and *Sleep*, as defined in section 4.2.3.1. The test was conducted using two multimeters. The first multimeter was used to measure the battery voltage (across the positive terminal and the DUT ground). The second multimeter, in current measurement mode, was placed in series with the device battery and the load. The current draw and battery voltage were measured for each device state, then multiplied to obtain the power drawn in each state. The expected battery life between charges was obtained by multiplying the power drawn with the time spent in each state. The test was considered failed if the energy usage would lead to the MachMoS device not meeting the device life target specification (S.2). The test was otherwise considered successful.

3.2.6 Data Transfer and Wireless Range

Since the BLE transmitter in the MachMoS node is contained in the enclosure, the node was tested at 100 m (S.8) and 150 m (the advertised range of the Ranger Pro device from section 3.1.2). The test was considered successful if data could be transmitted without fault over the required distance. The time required to complete the transfer was recorded for accurate battery life calculations. The test locations to obtain these distances are shown in Figure 3.4, between the West side of the 161 Louis Pasteur 7th floor conference room, and two locations along the Rideau Canal.

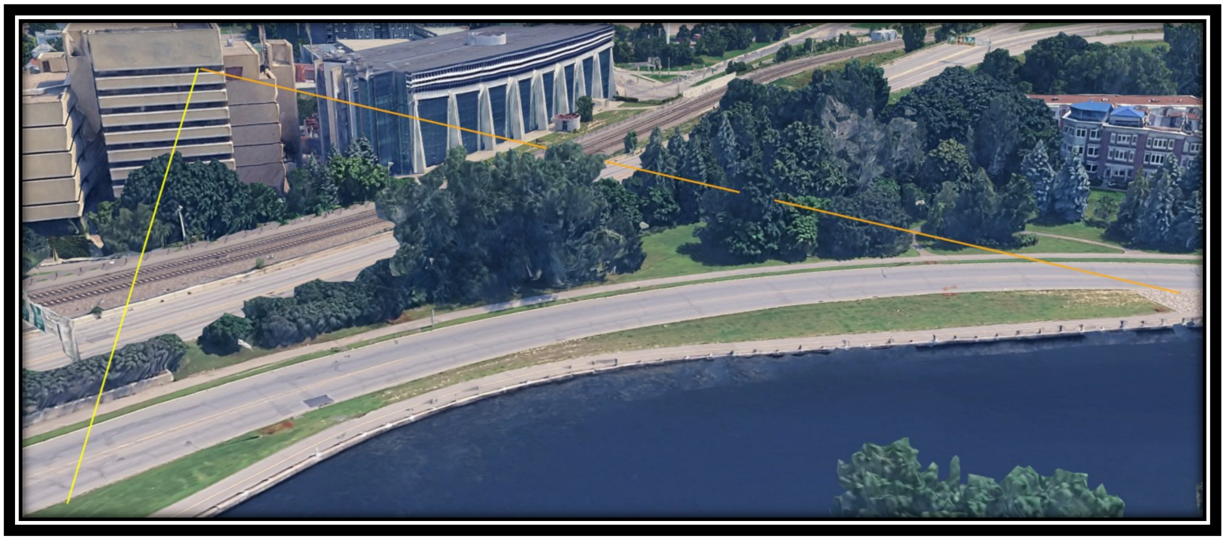


Figure 3.4 - 100 m (Yellow) and 150 m (Orange) LoS Distances

3.2.7 Electro Magnetic Interference and Compatibility

Per the information provided in subsection 2.11, the MachMoS node must meet emissions regulations listed in 47 CFR part 15 in the United States, and the ICES-006 as well as the RSS standards, in Canada (Appendix C.1). This is verified through electromagnetic interference (EMI) tests. The MachMoS node must also be tested for immunity against external emissions (electromagnetic compatibility, EMC). The criteria for successful tests are as outlined in 47 CFR part 15 regulations as well as the ICES-006 and RSS standards. These tests are to be conducted on the final version of the MachMoS node and remain beyond the scope of this work.

Chapter 4 Device Design and Construction

The following chapter discusses the steps undertaken to design and construct the MachMoS device. It should be noted that the design of a device such as the MachMoS node has multiple packaging challenges and is an iterative process. Due to this, the main design steps have been listed to explain the design process.

In this work, the microcontroller, sensors, battery, connector, enclosure materials and hardware were selected and purchased to construct the MachMoS node. Schematics of the system were prepared, ensuring no vital components were missing. This was performed in Electrical Computer-Assisted Design (ECAD) software and included necessary calculations for the expected life-between-charges of the node, the battery charger implementation, as well as voltage regulation. The materials were then chosen for the enclosure of the MachMoS node, with consideration for manufacturability, vibration response, and heat transfer characteristics.

Once these preliminary steps were complete, the PCB outline and enclosure could be modeled in Mechanical Computer-Aided Design software (MCAD). The PCB could then be laid out in ECAD using the schematics and MCAD data. The node was then manufactured and assembled for testing. The embedded software necessary to operate the MachMoS node was not developed by the author. As such, it is not discussed in this thesis.

An overview of the main subsystems of the MachMoS node are provided in Figure 4.1 and Figure 4.2.

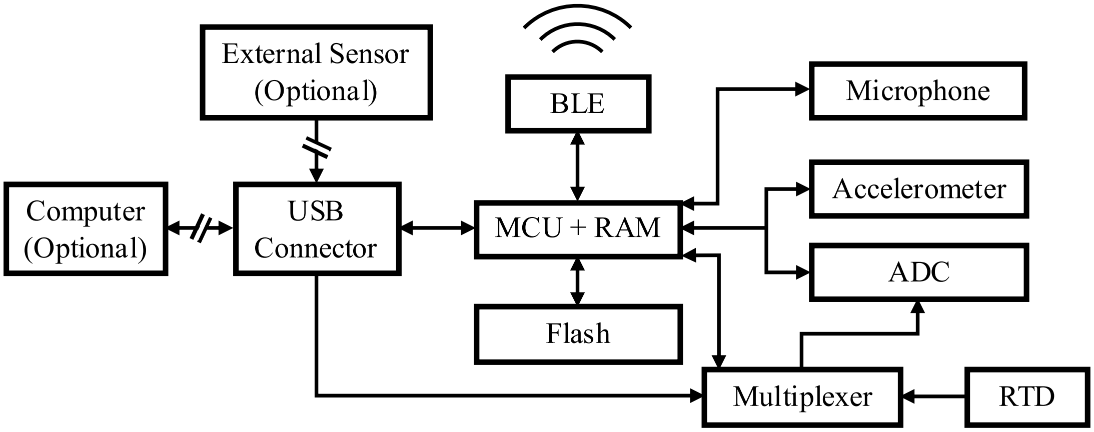


Figure 4.1 - MachMoS Signal Path

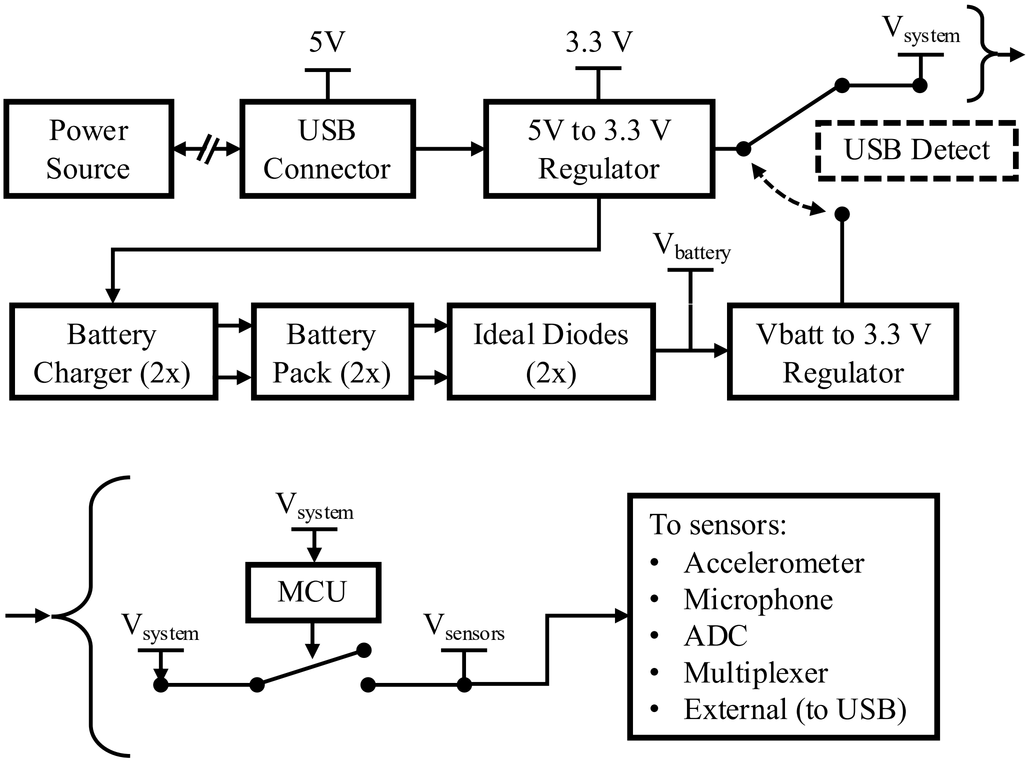


Figure 4.2 - MachMoS Power Path

An annotated picture of the fully-assembled MachMoS node, with lid removed, is provided in Figure 4.3.

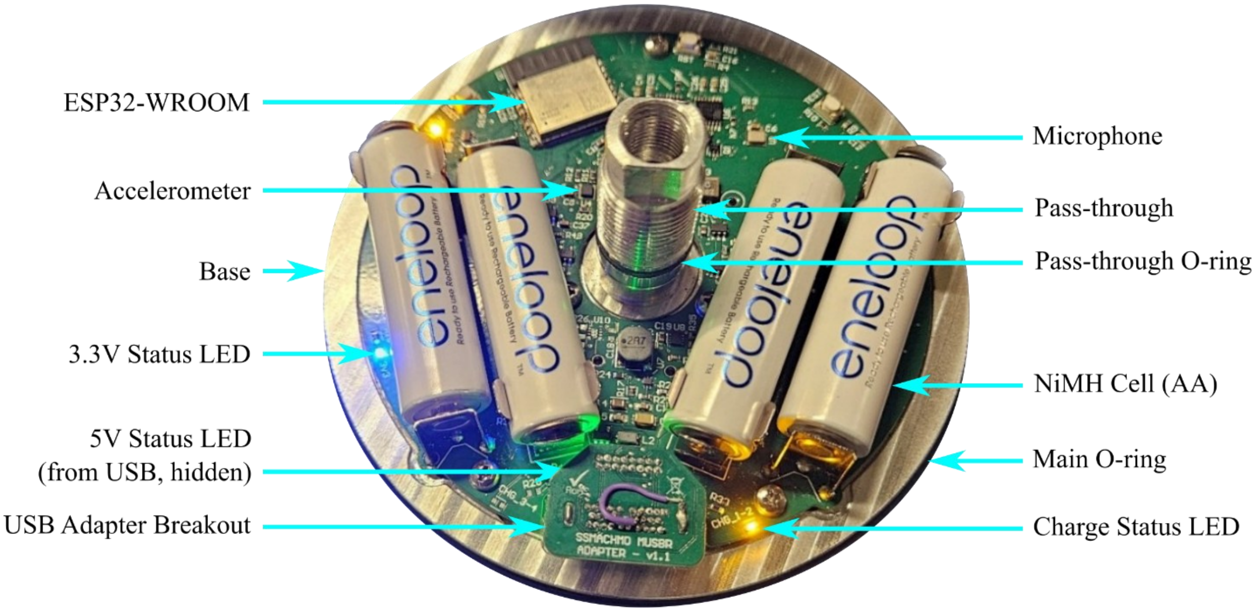


Figure 4.3 - The MachMoS Node (Lid Removed)

4.1 Microcontroller and Sensors

The following sensors and their choice will be outlined in this section:

- Tri-axial accelerometer (S.10.a),
- Microphone (S.10.b), and
- Temperature sensor (S.10.c)

The inclusion of an external magnetic flux sensor (optional, S.10.d) will also be discussed. Of course, a microcontroller capable of accommodating the data rates of the device (S.1) and a wireless communication system that has appropriate range (S.8) must also be included in the device.

4.1.1 Microcontroller

To organise, synchronise, and transmit data, the MachMoS node requires a microcontroller unit (MCU). As the MCU is a central part of the device, the right MCU will cover most of the requirements set in section 3.1. To meet these requirements, the selected MCU must be low cost (affordable), simple to write code for (simple to set up, use, and reconfigure), consume little energy (have long operational life), and be able to communicate wirelessly with other devices (work without Wi-Fi coverage, allow for over-the-air updates). Taking this into consideration, the ESP32-C3 microcontroller was selected. This MCU is a System on Chip (SoC) device, meaning it integrates many useful subsystems. Among these subsystems, the ESP32-C3 features a radio frequency (RF) subsystem which allows for direct input/output (IO) over Wi-Fi™ or BLE using an antenna. The ESP32 can also leverage Bluetooth Mesh to communicate with other nodes in a network to increase its effective range. Other features include many General-Purpose Input-Output (GPIO) pins, two integrated 12-bit analog to digital converters (ADC), and its ability to communicate with peripherals using common protocols such as the Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I^2C), Inter-IC Sound (I^2S), and Universal Serial Bus (USB). Finally, a low deep-sleep current consumption of 5 µA will ensure a long battery life of the sensor nodes.

To avoid the extra complexity of impedance matching required to accommodate a custom PCB antenna on every node main board, the ESP32-C3 MCU is not used directly on the board. Rather, a ESP32-C3-WROOM sister board is used. This sister board, developed by *Espressif Systems*, is a low-cost printed circuit board featuring a PCB antenna, the ESP32-C3 and other essentials such as 4 MB to 16 MB of external flash memory. The WROOM features castellated holes to allow soldering directly to a main board. Using the ESP32-C3 on a separate PCB also keeps the design modular.

If a different MCU were to replace the ESP32-C3 (due to deprecation, difficulty in sourcing the device, etc.), a sister board featuring a different MCU could be designed. As the expected functionalities that would need replacing with a custom sister board are very common among other MCUs (SPI, I^2C, I^2S, etc.), finding an alternative would not be difficult. Of course, transporting the device to another MCU could require significant code changes, unless the code is kept sufficiently abstracted and modular to ease portability to other MCUs.

4.1.2 Accelerometer

The accelerometer in the MachMoS node is perhaps its most important sensor. If a user is uneasy about sound captures from microphones in their plant, the microphone must be disabled in the device. From the specifications in section 3.1.3, the minimum capture rate for the accelerometer should be approximately 6.3 kHz (S.1). It should also be able to capture up to a ±16 g range (S.6) and be noise-free enough in this range to maintain an adequate SNR. The accelerometer must also feature a hardware-implemented low-pass filter before its digitization stage to ensure no aliasing occurs. To keep cost down, a MEMS accelerometer is to be used, as opposed to an analog capacitive-type accelerometer.

Digi-Key is a leading global distributor of electronic components [169] (Digi-Key). Their website was used to select key components in this work. By using filters in the *Accelerometers* [170] and the *Inertial Measurement Units (IMUs)* [171] sections of the Digi-Key website, the list of accelerometers was reduced from 2183 to 3, and the list of IMUs from 351 to 20. IMUs were included as these sensors typically have accelerometers. The filtered list of IMUs remains large given there are no filters for sensor bandwidth in this section of the Digi-Key website. The filters used to obtain these results were, in either section:

- Output Type: *I^2C, SPI* (and combinations of these)
- Product Status: *Active*
- Stocking Options: *In Stock*
- Exclude: *Marketplace Products*
- Environmental Options: *RoHS Compliant*

in the accelerometer section:

- Axis: *X, Y, Z*
- Voltage-Supply: *3.3 V* (and ranges containing this voltage)
- Bandwidth: *5.3 kHz* (minimum)

in the IMU section:

- Mounting Type: *Surface Mount, Wettable Flank* (and combinations, excluding chassis mount)
- Operating Temperature: *-40C to 85C* (or better)
- Package/Case: Any package excluding ball grid arrays (BGA) (to facilitate hand soldering and inspection)
- Price: Not exceeding 25$ CAD per unit

Of the 20 IMUs in the list, most are part of a sensor series. For instance, the BMI series of IMUs from *Bosch Sensortec* contains 3 IMUs: BMI088, BMI270, and BMI323, and those have similar features and ratings. A review of the datasheets for the IMUs shows that the most common IMU accelerometer has a maximum output data rate of 6667 Hz, with a low-pass filter at ODR/2 (though some have filters at ODR/9 at their *high-performance mode*). This restricts the bandwidth of these accelerometers at no more than 3333 Hz, lower than the minimum desired bandwidth. Unfortunately, none of the IMUs found feature bandwidths above 4 kHz. Of the 3 accelerometers that passed the filtering stage of the search, one was removed, the IIS2DHTR, because the maximum ODR for the device (5.3 kHz) was entered in the bandwidth field by mistake in Digi-Key. The bandwidth for a 5.3 kHz ODR is at most 2.65 kHz. The 2 accelerometers that meet the minimum specifications for the MachMoS node are the KX132-1211 and KX134-1211. These two accelerometers are from the same series, with the only big difference being their full-scale range (g-range). The KX132-1211 has selectable g-ranges of ±2 g, ±4 g, ±8 g, and ±16 g, and the KX134-1211, ±8 g, ±16 g, ±32 g, and ±64 g. Of course, the noise density of the KX134 is higher than that of the KX132, but that is to be expected given the larger measurement range. At the time of writing, the KX132 was approximately 13$ CAD, and the KX134 was 18$ CAD. The extended g-range attainable between the two devices (from ±2 g to ±64 g), along with these sharing the same software library and the same pinout makes this combination of accelerometers a good choice for the device, as the MachMoS node could be configured to use either device, depending on the size of the machinery being monitored. For the testing and validation of the MachMoS node, the

KX134-1211 will be used, as it features a better rated mechanical bandwidth, as shown in Table 4.1. It also meets the ideal larger g-range.

Table 4.1 - KX132-1211 vs. KX134-1211 Mechanical Bandwidth (-3dB) Comparison

Device	X-axis BW (Hz)	Y-axis BW (Hz)	Z-Axis BW (Hz)
KX134-1211 [172]	8200	8500	5600
KX132-1211 [173]	4200		2900

It should be noted that mechanical bandwidth denotes the linear region of the device, such that the device can capture frequencies above these, but the signal will be attenuated. Of course, this is problematic for comparing the device's output against a standard threshold for amplitude, but it remains useful when trending the RMS amplitude of the device's vibrations as a measure of the machine's health. In fact, the KX134 features a maximum ODR of 25.6 kHz (unfiltered), which was verified in this work using a similar setup as the one shown in section 3.2.2. The tests show that it was possible to capture all 3 axes near the Nyquist frequency (12.8 kHz) with an adequate SNR, and without artefacts being introduced in the frequency content for the X and Y axes. The Z axis, however, was attenuated and showed some artefacts slightly above 9 kHz. This is well beyond the rated bandwidth for the Z axis (5.4 kHz).

As per section 2.6, it was extremely important to choose a device that featured a hardware-implemented LP filter before digitization. The KX134 and KX132 both implement a LP filter with a configurable cut-off at ODR/2 or ODR/9, but it is unclear whether the filter was implemented before or after digitisation. As an additional interesting feature, the KX134 can be set up to "wake up" and send an interrupt to an external device if a pre-configured acceleration threshold is met. As for the device's accuracy, the lowest g-range of the device, ±8 g, is adequate for capturing some classes of machines throughout their life, from healthy to unhealthy, per ISO 20816. Both devices are 16-bit devices and meet specification S.7. With a g-range of ±8 g, a 16-bit resolution translates to 0.2 mg per code of resolution. The rated spectral noise is 300 µg/√Hz at ±8 g range, 50 Hz ODR, LP filter at ODR/2.

4.1.3 Microphone

The microphone in the MachMoS node is an important part of the device. While it does not directly sense mechanical vibrations, it allows for the sensing of vibration through the air pressure resulting from such vibration. It should be noted that this signal is also only one-dimensional, such that it is not as rich as that of a tri-axis accelerometer. However, it provides a low-cost solution to increasing the MachMoS node's measurable signal bandwidth.

When browsing for microphone solutions, MEMS microphones with an I^2S interface were the most desired. Higher bandwidth microphones were also favoured (as these are for recording sounds that can be heard by the human ear, their maximum bandwidth is typically in the range of 52 kHz, giving an $f_{\max}$ of 26 kHz. By using the following Digi-Key filters in the *Microphones* section of the website, it was possible to narrow down the list of viable options to 2 different I^2S microphones:

- Output type: *Digital, I^2S*
- Voltage: *3.3 V*
- Product Status: *Active*
- Stocking Options: *In Stock*
- Exclude: Marketplace *Products*
- Environmental Options: *RoHS Compliant*

The I^2S communication protocol is highly desired in this case as a device using I^2S contains all the required hardware to convert the analog sound signal inside the microphone to a digital signal, alleviating the cost burden of designing and implementing an audio ADC system. Additionally, I^2S is the standard used by the ESP32-C3 for communicating with microphones, and the I^2S hardware within the ESP32-C3 can push data to its RAM using Direct Memory Access (DMA). This means the audio data can be written to the ESP32-C3 memory without direct intervention from the MCU's processor, allowing for simultaneous microphone and accelerometer acquisition. Finally, the use of I^2S allows complete control of the sampling rate and data capture start/end times to the MCU (see subsection 2.8.3 of the literature).

Two microphone options were found, including the ICS-43432 and ICS-43434 from *TDK InvenSense*. Both microphones are similar in performance and price. The ICS-43434 was selected, as it is a more recent model and draws less current (0.55 mA as opposed to 1.5 mA).

The ICS-43434 is a low-cost MEMS microphone with a 24-bit resolution (meets S.7) and a linear bandwidth between 60 Hz and 20 kHz. The microphone features a built-in low pass filter with a pass band that extends to 41.7 % of the microphone's sampling frequency (at its maximum sampling frequency, 51.2 kHz, the -3 dB cut-of is 21.4 kHz). like with the accelerometer, it is possible to increase the sampling rate of this microphone to obtain a band beyond 20 kHz, with the caveat of reduced sensitivity.

Given the higher signal frequencies involved (at its maximum sampling frequency, 51.6 kHz, the clock signal from the MCU must be set at 3.3 MHz), care was taken when laying out the PCB to maintain a short distance between the MCU and the microphone. The circuit board was designed such that the I^2S signals were routed directly above an uninterrupted ground reference plane to reduce the impedance of the signal carrier traces and to ensure that its electromagnetic field is contained.

4.1.4 Temperature Sensing with the ADC

An important metric in MCM is the temperature of the equipment being monitored. Many options for temperature measurements exist such as thermistors, IR temperature detectors, thermocouples, and RTDs. These four options were fully described in the literature review (section 2.8.1). As mentioned in the literature review, thermistors are known to drift, making them unsuitable in this application. As for IR temperature detectors, their accuracy is highly dependent on estimating the correct emissivity for the surface being measured. At room temperature, thermocouples generate very small voltages. As such, they require signal filtering and an additional amplifier circuit. Measurements made by platinum RTDs are based on an intrinsic material property (i.e. platinum's electrical resistivity). Since platinum is a very stable metal, it is the best option in this application as it will provide stable, accurate and repeatable measurements.

The RTD was selected using the Digi-Key website based on the following parameters:

- Excitation current: 0.05 mA to 1.5 mA, corresponding to the excitation current range of the ADC used.
- Operating Temperature: All ranges containing -40°C to 120°C.
- Mounting Type: *Free-Hanging, Through-Hole*
- Product Status: *Active*
- Stocking Options: *In Stock*
- Exclude: Marketplace *Products*
- Environmental Options: *RoHS Compliant*

This search reduced the count of RTD models to 109 results. Sorting by rated current draw shows one pt10000 at 100 μA, one pt2000 at 200 μA, followed by many pt1000 RTDs rated for 300 μA. The prices vary greatly, from 2.75$ CAD to 175$ CAD, with an average price of 22$ CAD (median of 7$ CAD). To obtain a more reasonable estimate for average (reasonable) platinum RTD prices, outliers were removed from the list using a Chi-squared (χ^2) distribution fitted to log-price data, and a 90th percentile cut-off gave a new average to 8.5$ CAD. This analysis is provided in more detail in Appendix A.1. To further narrow down the selection of RTDs and to ensure more confidence in the values obtained when testing the ADC circuitry, the Digi-Key list was narrowed down to A-rated RTDs only (rating per IEC-60751). Using this filter, the list was further reduced to nine results, three of which were 300 µA (pt1000) RTDs. All three pt1000 devices had large quantities of stock available. These were from the L and M (low and medium temperature) series of the *Yageo Nexensos* company. Of these RTDs, the low-range detector (part number: 32207582) was chosen. The price of this RTD is lower than average (5.50$ CAD per unit) which is lower than 53% of platinum RTDs in the original filter (with outliers removed).

The two-wire ratio-metric measurement method was used in the node device to determine the resistance of the RTD. This method ensures accurate resistance readings despite fluctuations in the supply to the circuit and is easily implemented using a differential ADC that accepts an external voltage reference. Unfortunately, the ESP32-C3's ADC accepts neither differential inputs nor external reference voltages. Due to this, a *Texas Instruments* ADS1120 was added to the circuitry. While its sampling rate is rather high for this application, 2 kilo-Samples per second (kSPS), it must be kept in mind that this ADC can be used for other analog sensor inputs that may be added to the node in the future (see section 4.1.6). In the two-wire ratio-metric measurement method, the resistance of the RTD may be computed using a simple ratio of the digital binary number output from the ADC (ADC Code), as shown in equation (10):

$$R_{\mathrm{RTD}} = R_{\mathrm{REF}} \cdot \frac{\mathrm{Code}_{\mathrm{ADC,\,RTD}}}{\mathrm{Code}_{\mathrm{ADC,\,FS}}} \tag{10}$$

where $\mathrm{Code}_{\mathrm{ADC,\,RTD}}$ is the output code (ADC reading between -2^{15} and 2^{15}-1) from the ADC, and $\mathrm{Code}_{\mathrm{ADC,\,FS}}$ is the ADC full-scale code, 2^{15}-1. R_{RTD} is the resistance of the RTD, and R_{REF} is the resistance of the reference resistor, in Ohms. With an advertised 16-bit noise-free resolution [91], [174] and an expected device range of -40°C to 130°C (corresponding to an RTD resistance of 842.7 Ω to 1500 Ω), the range of valid RTD codes is 18408 to 32767, giving a final RTD resolution of 0.012°C per binary digit.

The ADS1120 was chosen because it has the necessary features to perform temperature measurements using an RTD. Of note, this ADC can output a current to excite sensors, reducing the need for extra circuitry. Preference was given to RTDs that have a rated current draw close to that of the ADC's current source, between 0.05 mA to 1.5 mA. It should be noted that these current draw ratings are recommended current draws that consider RTD self-heating. Of course, smaller current draw values are more desirable to increase the device battery life but negatively affect the SNR.

An error propagation analysis was performed under the assumption that the R_{REF} chosen is a ±0.1 % tolerance resistor, knowing that an A-rated RTD has a 0.06 % tolerance, and knowing that the ADC has a 0.015 % gain tolerance. The full analysis is shown in Appendix A.2. The propagated uncertainty can skew the temperature up to ±0.40°C. This uncertainty can be alleviated from calibrating the RTD and reference resistor, but calibration is an extra step that would add cost to the sensor. The ±0.40°C accuracy is acceptable for this application, where variations in temperature are more important for analysis than the actual temperature value.

In this design, once the resistor value for the RTD is determined from the ADC reading, a solved version of the CVD equation (9), is used to convert the measurement to a temperature reading. For readings above 0°C (1000 Ω), the equation is a 2nd order equation, and solving for temperature is as simple as using the quadratic equation. Below 0°C, however, temperature would be obtained by solving a fourth order polynomial, and while closed form solutions exist for such problems, it is much more computationally efficient to use a best-fit approximation of the CVD equation over the full temperature range. For a first order model, the conversion would be $T = (R - 999.34)/3.861$, with at most 0.8°C error on temperature. For a second order model, the conversion is given in equation (11).

$$T = \frac{-3.9084 + \sqrt{15.272 + 2.3237\mathrm{E}^{-3}(1000.0 - R)}}{-1.1619\mathrm{E}^{-3}} \tag{11}$$

or, slightly more computationally efficient, equation (12).

$$T = 3364.0 - \sqrt{1.3038\mathrm{E}^{7} - 1.7214\mathrm{E}^{3}R} \tag{12}$$

with less than 0.01°C error over the whole range, as shown in Figure 4.4. It should be noted that the ESP32-C3 is not equipped with a Floating Point Unit (FPU) and emulates floating-point math through software [175]. Due to this, the computational cost of a square root operation is high. However, this operation is performed once every capture. As such, the 2nd order approximation is used in the MachMoS node. The plots for the error on temperature when using 1st or 2nd order approximations are shown in Figure 4.4.

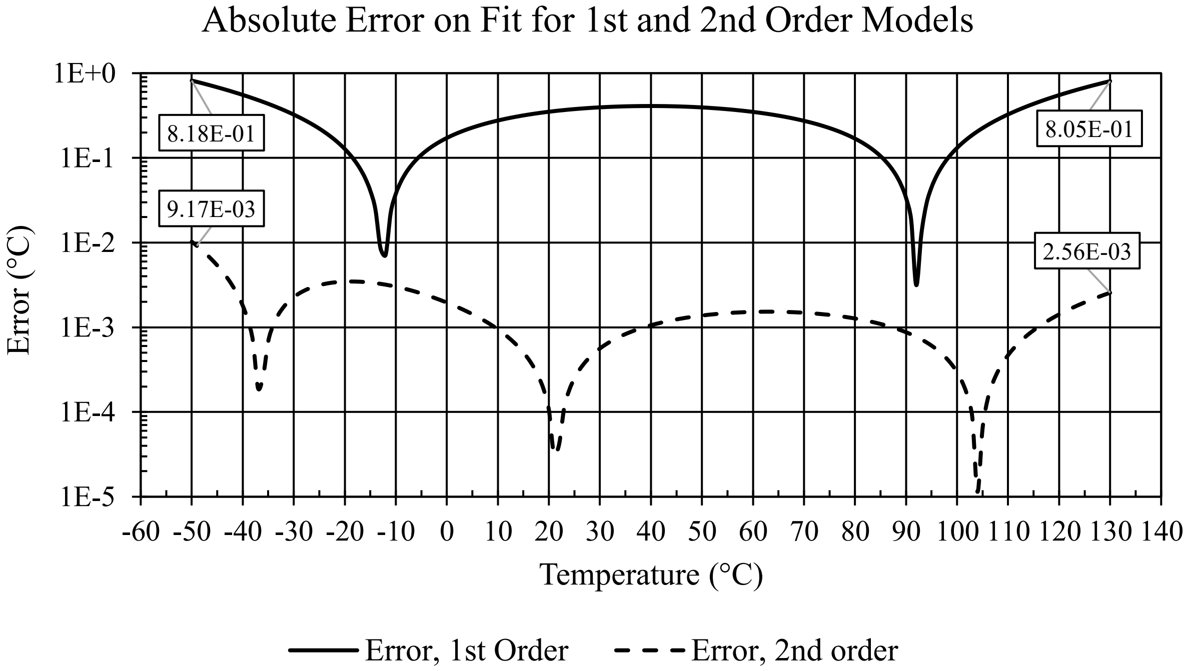
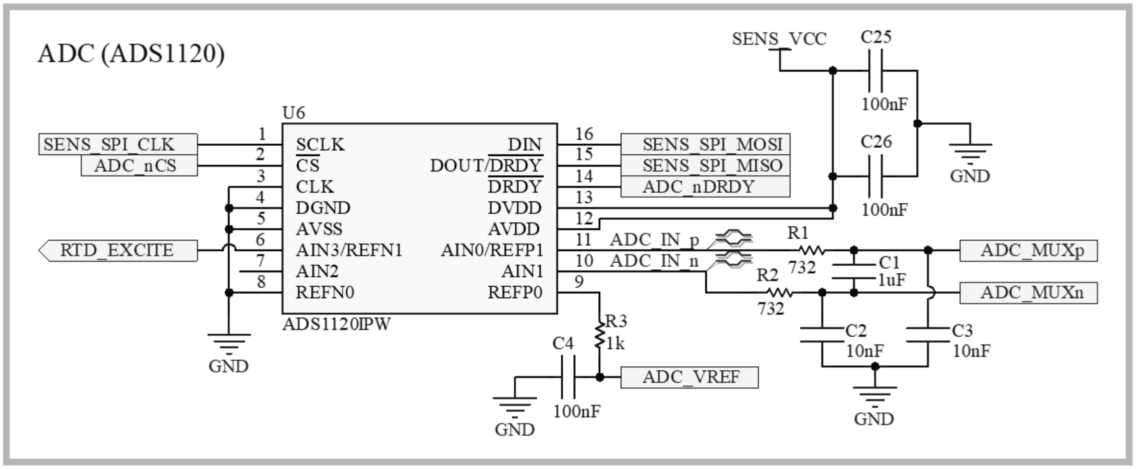
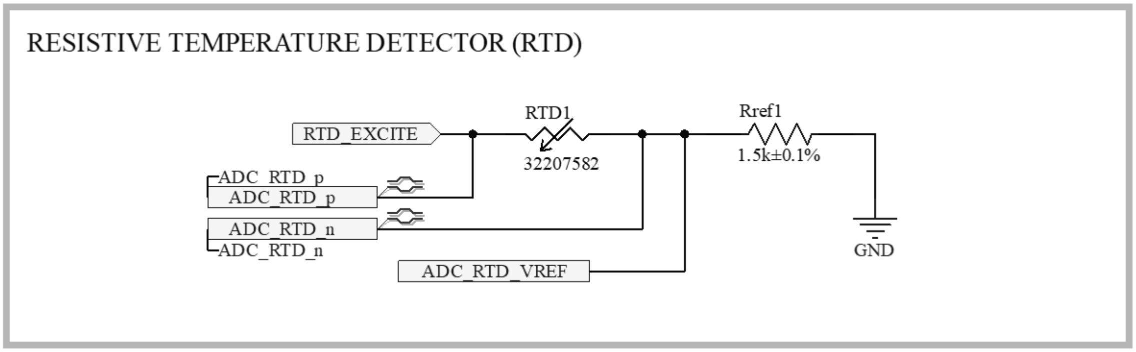


Figure 4.4 - Error on Fit for 1st and 2nd Order Models

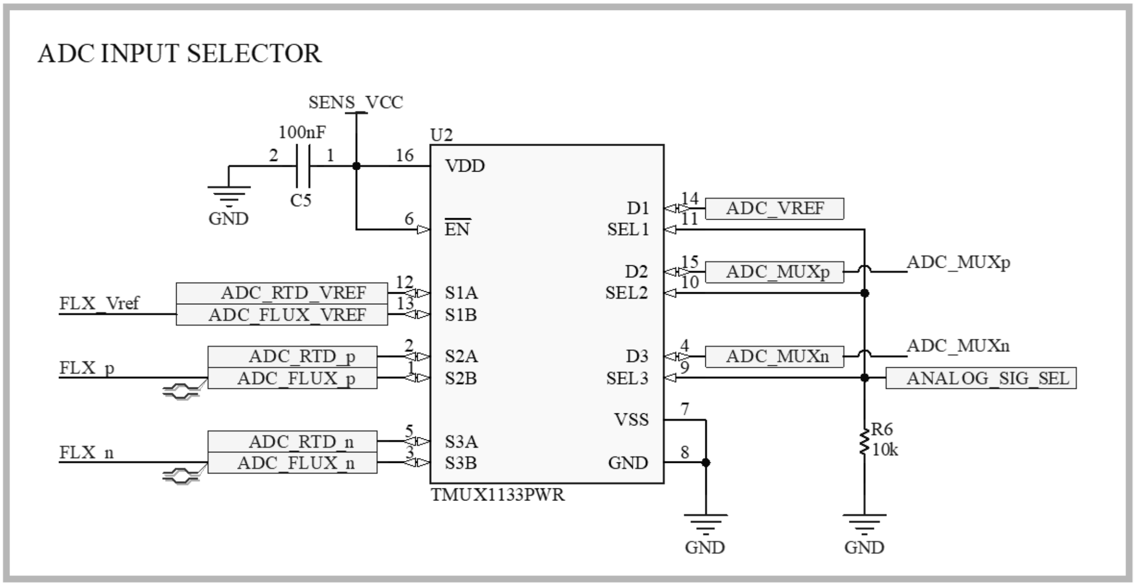
Figure 4.5 shows the circuit that was implemented for the ratio-metric temperature measurements using the RTD. Since the ADC is not expected to be used at the same time as external sensors, the channel used for the RTD is shared with an external sensor though the use of an analog multiplexer circuit (TMUX1133), which allows choosing the input going to the first differential channel of the ADC. In the actual implementation of the circuit, a connector leads to the RTD1 and Rref1 components (to allow for the RTD to be mounted on a breakout board, against the aluminum enclosure). The connector was removed for clarity in Figure 4.5. As can be seen in the figure, the RTD circuitry implements passive filtering for common-mode and differential mode noise rejection [91].



(a) ADC Connections



(b) RTD Connections



(c) ADC Input Selector (RTD vs. External Sensor)

Figure 4.5 - Ratio-metric RTD Circuit Using the ADS1120

4.1.5 Flux Sensing with the ADC

The rotational speed of motors can be extracted by measuring their stray magnetic flux. The fundamental vibration frequencies of machinery can be calculated from motor speed and other parameters. Stray flux can also be used to monitor the health of a motor’s windings.

Flux measurements are enhanced when a sensor is in close proximity to a motor’s windings. Thus, these sensors have different recommendations for mounting from those of a vibration sensor. Therefore, it was determined that a flux sensor should not be integrated within the MachMoS node enclosure. The inclusion of such a sensor external to the MachMoS node requires a channel on an ADC to read its response. The ADC selected in this work provides for such a capability and for additional external sensors should they be required to be added to the MachMoS node in the future.

From section 2.8.4, the stray flux from motor enclosures is in the µT to mT range. Considering this range, the DRV425 Fluxgate sensor was used. This sensor boasts a range of ±2 mT and a spectral noise of 1.5 nT/√Hz. It also features over-range and error flags, which is very useful when optimising the placement of the sensor. Of course, the rate at which the poles alternate in an induction motor connected to AC power lines (mains) is related to the mains frequency (60 Hz). If a VFD is used, the pole switching frequency can be higher, but typically will not exceed 590 Hz, per section 2.8.4.

Given these considerations, the Nyquist-Shannon sampling theorem predicts that the flux sensor signal must be sampled at a frequency of at least 1180 Hz (2x 590 Hz). Second and third order harmonics are commonly used to extract additional information from the magnetic fields generated by motors [9]. As such, the desired sampling frequency for a flux sensor is at least 3.54 kHz for 590 Hz drives, 360 Hz for the more common 60 Hz drives. The ADS1120 used in the MachMoS node can sample at a rate of up to 2 kSPS. It may be used for first, second, and third order analysis of machines generating a flux frequency of 590 Hz, 500 Hz, and 330 Hz, respectively.

4.1.6 ADC

As previously mentioned, the MachMoS node must support external sensors. A multi-channel ADC, the *Texas Instruments* ADS1120, was selected as the ADC in the device. The ADS1120 is a 4-channel, 16-bit ΔΣ ADC (meets S.7) with a maximum sampling rate of 2 kSPS and an FIR filter that can be configured to reject 50 Hz and 60 Hz noise from the mains [174]. It is possible to disable this filter when capturing signals from a flux sensor, which will commonly be capturing 50 Hz or 60 Hz signals. The 4 channels of the ADC can be used as single-ended channels or may be combined to measure differential signals. Each channel's full-scale measurement can be set by an external reference voltage (a requirement for the RTD's ratio-metric measurements). The high sampling rate (2 kSPS) was justified in section 4.1.5. Channels 2 and 3 of the ADC were not connected in this work. The inputs to the ADC were equipped with common-mode LP filtering using 732 Ω in-line resistors and 10 nF capacitors to ground (~21.8 kHz cut-off) as well as differential-mode LP filtering using a 1 uF capacitor between the lines for channels 0 and 1 (~1 kHz cut-off). This is general practice for RTD measurements as recommended in the literature [91].

4.1.7 Synchronization, Sampling Rates, Data Management

To allow for the data to be used in sensor fusion applications, the MachMoS node synchronises the measurements from its accelerometer and microphone. As mentioned in section 4.1.3, the microphone used in the node is an I^{2}S microphone. This type of microphone requires a clock input generated by an external source. In the MachMoS node, the MCU is used as the clock source, such that it controls the initiation of the microphone measurements. The accelerometer, however, is more difficult to synchronise as it generates its own internal clock signal and begins sample capture on its own once configured. Due to this, it was necessary to synchronise the capture routine of the microphone to that of the accelerometer. This was possible using the accelerometer's interrupt pins. This feature is common among digital sensors, where an interrupt pin can trigger when a sample is ready to be read. Using this feature, it was possible to synchronise the capturing of audio data to that of the accelerometer. The MCU configures the accelerometer, after which the accelerometer starts sending interrupts. The MCU can discard the accelerometer data until it is ready to start capturing accelerometer and microphone data simultaneously. Once this is the case, the MCU starts storing accelerometer and microphone data.

Ideally, the accelerometer and microphone sampling rates would be multiples of each-other, so that the raw data can be used as-is. However, it can be difficult to obtain an exact multiple of the sampling rate of the accelerometer for the microphone data for some accelerometer sampling rates and I^{2}S decimation ratio combinations. As the MCU clock outputs are scaled from the main clock for the MCU (160 MHz). For instance, a 25.6 kHz ODR on the accelerometer would ideally translate to a 51.2 kHz sampling rate on the microphone. This combination is possible with no error on microphone frequency for a 192x decimation ratio, but not for any of the other valid decimation ratios compatible with a 24-bit output word length (384x: 6.69 ppm, 576x: 1.45 ppm, 768x: 11.03 ppm and 1152x: 11.03 ppm) [176], as shown in Appendix A.3.

The synchronisation of the data can be enhanced by interpolating the data from either sensor. This can be done on the gateway side to avoid costly floating-point operations. A simple solution is to use Fourier, and Inverse Fourier Transforms (FT, IFT) of the existing accelerometer data to interpolate it to the microphone data's sampling frequency, or a factor of that frequency. This method is shown in Appendix B.2. It is:

- stable and fast, thanks to the Fast Fourier Transform (FFT),
- can be implemented in a single line of Python code on the gateway using the native *scipy.signal* library [177], and
- does not introduce any significant degree of error to the capture.

Finally, it is important to consider the quantity of data being gathered by the MachMoS node. This is to ensure that sufficient memory is available to store this data. By design, the MachMoS node does not compress time series data into frequency domain data or KPIs, unlike many other commercial sensors as listed in the "output type" entry of Table 3.1. This is to give MCM researchers freedom over the pre-processing method of their data for use in training predictive models.

At its maximum rate of capture, the MachMoS node samples 16-bit accelerometer data for the X-axis, Y-axis and Z-axis of the accelerometer at 25.6 kHz, and 24-bit microphone data at 51.2 kHz. This amounts to 2.46 Megabits per second (Mbps) (307.2 kilobytes per second, kBps). The temperature data has minimal memory requirements, as it amounts to a single 16-bit sample per capture cycle. With up to 16 MB of flash memory available on the ESP32-WROOM device, the device can capture up to 53.3 s of data (13.3 s for the more common 4 MB versions).

Finally, flash wear-out was considered in the design. As per the datasheet for the ESP-WROOM, the flash IC can sustain 100 000 write cycles. In the regular operation of the MachMoS node (one-second captures every hour, 24 hours per day), this would mean an expected flash useful life of 5.7 years (including erasing the flash and writing to it at each capture). A wear-levelling algorithm can be used to increase that life to approximately 76 years, with the useful life being as low as 5 months without wear-levelling, and 5.7 years with wear-levelling (if the captures are in fact 13.3 seconds long, one every hour).

4.2 Electronics

4.2.1 BLE

A Bluetooth RF system, including a PCB antenna, is included in the ESP32-WROOM. This makes the BLE implementation very simple. Given its BLE capability, the device meets specifications S.14. Care must simply be taken as to not include copper in the PCB layout under the PCB antenna. This ensures the impedance of the antenna isn't affected, and that there is no return path for the signal, as this would significantly attenuate the antenna's power.

4.2.2 External Connector

When initially deciding on a connector, some considerations must be made, such as the achievable quality of the machining of the enclosure (to ensure an adequate surface finish for sealing), the cost of the connector, the size of the connector, and its hand-solderability. Among available IP-rated connectors are older standard connectors, such as D-Sub, Ethernet, etc. as well as shell connectors, which are typically military-specification. These connectors, however, can be rather expensive, would require special connectors for charging the device, and are relatively large, making their packaging within the sensor challenging. A smaller, more common connector, which is much more compact and, due to the proliferation of smartphones, has been mass-produced in a waterproof configuration, is the USB-C connector.

Full-featured USB-C connectors feature 24 pins, 8 of which are for transmit and receive (TX/RX) for 2 distinct channels (differential pairs). If the MachMoS device configures itself properly using two 5.1 kΩ VBUS pull-up resistors on the CC1 and CC2 pins, as stated in the USB-C specification [178], this allows this type of connector to be used for 4 single-ended extra signals or 2 differential signals. While this is convenient for the addition of analog sensors to the device, it is not entirely prudent. Some USB-C cables contain integrated circuits that are used to completely disable the TX/RX lines if they are not configured for use. As such, only some cables may be used for external sensor communications. Of course, regular USB communications can be used with external sensors, and the USB pins on the ESP32-WROOM are reserved for this purpose. Interestingly, Digi-Key and other popular electronics hardware distributors have few IPX7 rated (or better) USB-

C full-featured connectors, and even fewer vertical ones. A vertical connector is the best way to ensure a gasket is effective, as the connector will be interfacing with the metal part of the enclosure and not with the plastic cover. Since this plastic cover will be 3D printable, it would not meet the surface finish requirements to ensure an effective seal. The best choice for this configuration, at the time of prototyping, was the Amphenol MUSBR series, which has a connector for approximately 12$ CAD (at time of prototyping), the most expensive component in the MachMoS device. This component, the MUSBRM5C1M0, is a shielded 24-pin full-featured, flange-mounted USB-C connector. While expensive, this flange mounted connector makes it the perfect prototyping connector due to its very low risk of fluid ingress. The connector simply requires a break-out board designed to accommodate for its size (roughly 13 mm tall). Unfortunately, the component's price has increased to 24$ CAD at the time of writing. It now has much cheaper competitors from CUI Devices (Same Sky), GCT, and Adam tech. If the MachMoS device were redesigned, the Amphenol UC-00PFFT-QF7001, CUI Devices UJ40-C-V-SMT, GCT USB4725-03-C or Adam Tech USB-C40-S-VT-CS51A-TSMT-WP-BK-PP are all options under 4$ CAD that could be leveraged as long as good geometric controls are applied to the enclosure design. These options, which average 5.5 mm in height from PCB to sealing interface, would not require a break-out board like the MUSBR connector does.

As for maximum current ratings and rated voltages, the MachMoS device is configured for legacy USB support. As such, most USB-C connectors will be rated for higher current draws and voltages than they would encounter: 5 V, 2 A. Currently, full-featured USB-C connectors are typically rated for at least 12 V, 5 A (the most recent USB Power-Delivery standards attains 48 V).

4.2.3 Battery

4.2.3.1 Energy Use

Energy delivery is a very important consideration in the design of an IoT sensor. The MachMoS node battery must be rechargeable to meet target specification S.12. The requirements for the MachMoS node do not stipulate a specific battery life, though a 12-month minimum is a good target. This would allow for yearly maintenance to align with charging the device. To determine the expected battery life, the duty of the device is split into 3 tasks: *acquire*, *send*, and *sleep*. The power draw from the battery is calculated from the average expected power usage by each powered component during a capture cycle. Leakage is included in the calculation whenever a component should be shut off during a task. Since the flux sensor is a separate device, it is not included in the analysis. The efficiency of the volage regulator in the device must be considered in the analysis. The regulator chosen for this application is a SMPS. A well-chosen SMPS will feature voltage conversion efficiencies over 90 % and is generally more efficient than linear regulators such as LDOs. However, these regulators are not efficient in light-load applications (in the nA to µA range) such as when the device is in deep sleep between capture cycles.

In a capture cycle, the device first wakes up from its sleep mode to start the *acquire* task. In the *acquire* task, the MCU and sensors are switched on and require some time to start outputting valid data. The MCU must wake up first, before it can activate the proper GPIO pin to allow current to flow to the sensors. The wake-up time for the MCU is not listed in the manufacturer's datasheets, but users have reported MCU wake-up times of approximately 200 ms [179]. Per datasheets, the time required for this is at most 20 ms for the microphone, at most 2 ms for the accelerometer, and 50 µs for the ADS1120 ADC. That is, the temperature measurement (first to be measured) can be taken as soon as 50 µs after device wakeup. Once this measurement is made (50 ms at 20 SPS), the accelerometer and microphone are ready to capture. This part of the cycle takes approximately 1.3 s, and the 50 µs required for the ADC setup is considered negligible. Other devices that are powered during this time are the flash device and a p-channel MOSFET (p-FET). The power usage by the MOSFET is negligeable, as the gates only require a charge in the nano-Coulomb range. The on resistance of the MOSFET is in the mΩ range, which would only lead to losses in the µW range, since the sensor current draw is in the mA range. Reading and storing data are considered concurrent tasks. The average time (duty) for the *acquire* task is therefore 1.55 s, 0.2 s of which the ESP32-WROOM is the only device consuming power. The current draws and power consumption (assuming 3.3 V) for each device are listed in Table 4.2.

Table 4.2 - Power Usage During the Acquire Task

Device	Comment	Current Draw (mA)	Power (mW)
ESP32-WROOM	Assuming all clocks running [176]	28	92.4
ADS1120	Normal mode, IDAC and external reference, no PGA usage [174]	0.5	1.65
KX134-1211	High performance mode, 25.6 kHz ODR [172]	0.162	0.53
ICS-43434	High performance mode [180]	0.490	1.62
	Total:	29.15	96.2

By multiplying the power usage with the expected duty for each device, the power required for the *acquire* task is estimated to be 41.2 µWh.

The energy required by the *send* task depends on the amount of data sent. At 1 Mbps transfer rate, the rated peak current consumption is 345 mA [176]. At the maximum ODR, the amount of data generated is roughly 2.46 Mb. Therefore, the transfer should take approximately 2.46 s. During the send task, the only devices that are powered are the ESP-32-WROOM (28 mA) and the embedded radio system (345 mA). The power consumption for the send task is therefore estimated at 840 µWh.

Finally, the energy required by the *sleep* task was determined. Leakage through the sensors, during the *send* and *sleep* tasks, was reduced by a p-FET. Therefore, only the consumption from the regulator (quiescent current) and ESP32-WROOM were considered. Per the manufacturer datasheets, the ESP32-WROOM consumes 5 µA during its deep sleep cycle, and the regulator consumes approximately 0.75 µA in quiescent current. Assuming a single capture cycle per hour, the energy consumption would be 19.0 µWh.

The energy consumption figures must be adjusted to consider the efficiency of the regulator during these tasks (see Section 4.2.4). Table 4.3 lists the tasks, their current draw, the corresponding regulator efficiency, as well as the original and adjusted energy required for each task.

Table 4.3 - Energy Usage per Capture Cycle, Adjusted for Regulator Efficiency

Task	**Current Draw (mA)**	**Efficiency (%)**	**Energy (µWh)**	**Energy, Adjusted (µWh)**
Capture	29.15	96 %	41.2	42.9
Send	373	97 %	840	866
Sleep	0.00575	12 %	19.0	140*
			Total:	1049

* Regulator quiescent current is not affected by efficiency

To estimate the energy usage on a yearly basis, the total energy per capture cycle was multiplied by 8760 [hrs/year]. This results in 9.21 Wh per year for 1 capture cycle per hour.

4.2.3.2 Cell Choice and Battery Configuration

Based on the data in the literature review, NiMH is the only cell chemistry that is a good choice for the desired temperature range of the MachMoS node (-40°C to 85°C, ideally 120°C, S.4). Among NiMH batteries, a low self-discharge battery is required, to avoid the battery not delivering as much energy as desired. *Panasonic*'s line of low self-discharge Eneloop cells have been shown to lose 10 % to 40 % of their charge after 12 years and 9 months [181]. Assuming the self-discharge rate is a function of the current battery capacity, it can be modeled using a simple compounding model: $c_l = (1 - l)^t$, where c_l is the total percentage capacity loss, for time t. l is the loss per unit of time t. A 10 % to 40 % loss per 12.75 years therefore translates to a loss of 22.7 to 110 μWh/Wh·day, or 8.2 to 39 mWh/Wh·yr.

The Eneloop series features normal and Pro grade AA cells. Normal cells have a rated minimum capacity of 1900 mAh, whereas the Pro series has a capacity of 2500 mAh. At 1.2 V nominal, this translates to 2.3 Wh and 3.0 Wh, respectively (2.21 Wh and 2.55 Wh with a year of self-discharge). It should be noted that the Pro series has a much higher self-discharge rate, with an advertised maximum 15 % capacity loss after 1 year. With an energy requirement of 9.13 Wh/yr and self-discharge factored-in, the MachMoS node requires at least 4 AA cells. Considering the 80 % higher cost of the Pro series cells ($8.50 CAD [182] vs. $4.69 CAD [183]) for an extra 15 % capacity after one year, the normal series cell is recommended, though the Pro cell can always be substituted, depending on testing results.

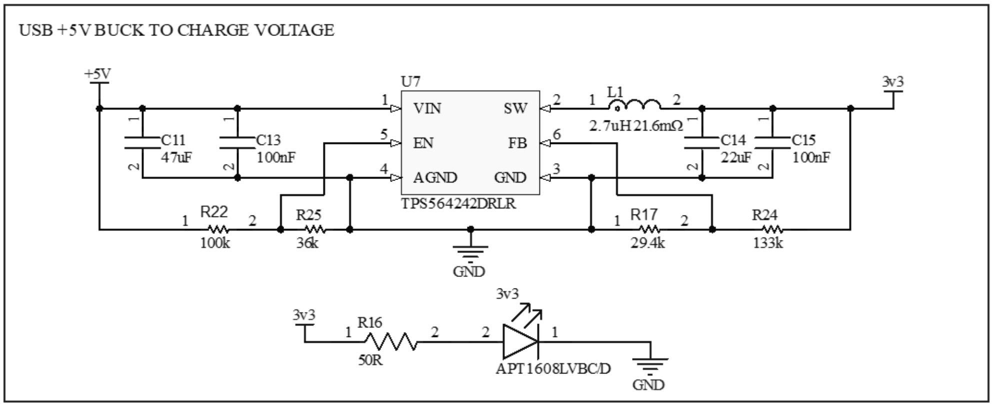
With four cells at 1.2 V nominal each, the possible configurations (battery voltage, regulator type) are 4 cells in series (4s) (4.8 V, buck regulator), 4 cells in parallel (4p) (1.2 V, boost regulator), and 2 parallel batteries of two cells in series (2p2s) redundant configuration (2.4 V, boost regulator). While 3 cells in series (3s) (3.6 V, no regulator) could potentially work, the savings from bypassing the need for a regulator are not significant enough to drop the need for a fourth battery. In addition, the fully charged voltage of an NiMH cell can be closer to 1.5V, which would make a 3s battery voltage 4.5 V, exceeding the working voltage of the ESP32-C3 (3.0 to 3.6 V). An important consideration for the 4p and 4s configurations is potential cell failure. If a cell in a 4p configuration is damaged, it could affect the whole battery, by sinking energy from the other three cells. If a cell in the 4s configuration is damaged, the effect might not be as severe, unless the failure mode creates an open circuit condition, isolating the GND of the battery from the system GND. Finally, if a cell fails in the 2p2s redundant configuration, only one half of the battery pack is affected, and the device can keep functioning, leading to a more reliable battery. This makes the 2p2s redundant configuration, with a boost regulator, a better choice for this application.

4.2.3.3 Battery Charging

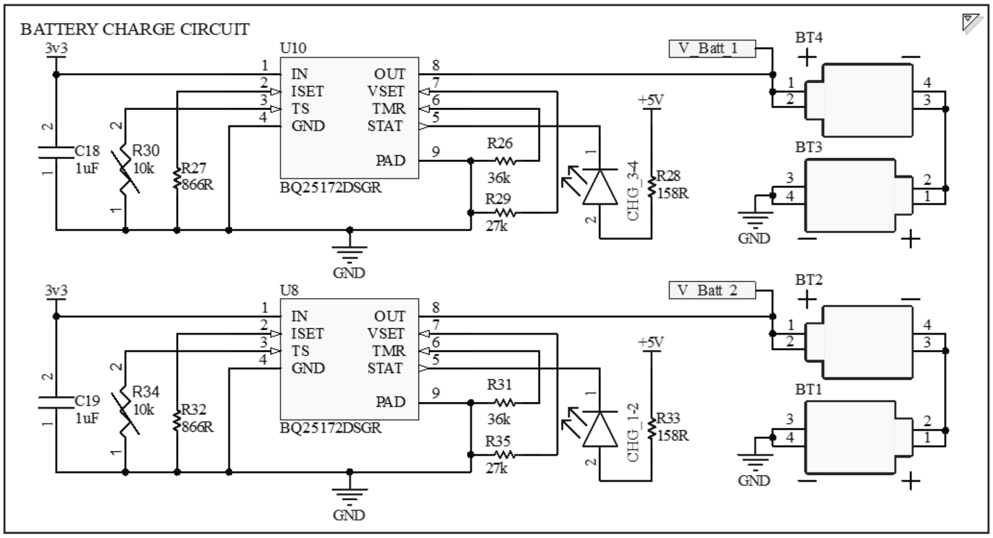
With two separate parallel battery sets, two independent charge circuits are implemented in the MachMoS device. A well-documented charge IC for NiMH chemistries is the *Texas Instruments* BQ25172. This IC features an 800-mA maximum charge current (1.6 A charging current total, approximately 5-hour charge time), with fault protection against overvoltage, overcurrent, and overtemperature conditions. The charger can also detect when the charging cycle is complete using voltage and current sensing, as opposed to using charge time which can damage cells. For proper end-of-charge detection, the pack must be isolated from other systems. Otherwise, a load might be detected by the charger circuit which isn't due to charging, and the charger will continue pushing current into the battery despite it being at full charge.

However, if the batteries are isolated, the device cannot be powered. This is problematic in two ways. In industry, the user may want to wire their device permanently if a power source is readily available. Additionally, maintaining power to the device when an external host is connected via USB is important. To remedy this problem, a bypass circuit can be used. Ideal diodes (*Analog Devices Integrated* MAX30203) are used in series with the battery output and are disabled if voltage from the USB port is detected. The ideal diodes are also useful for toggling which battery pack is outputting to the device, and to avoid one of the packs discharging into the other if the packs are not balanced. A slow RC circuit is used to delay the re-enabling of the diodes (and, therefore, the battery packs) when the USB connector is disconnected.

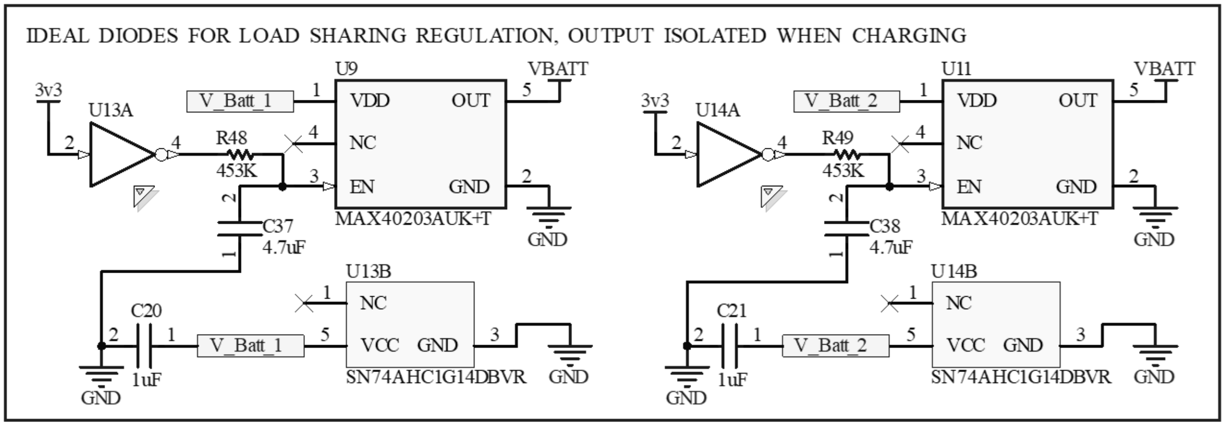
A voltage regulator is required to step down the 5 V USB input to a 3.3 V charge voltage and system voltage when the battery packs are bypassed. The maximum expected current draw for the regulator would be 2x 800 mA at 2.4 V, and up to 400 mA for the MachMoS device. The *Texas Instruments* TPS564242DRLR was selected for this purpose. Figure 4.6 shows the charging, isolation, and regulation circuitry.



(a) USB +5V Regulation to Charge Voltage



(b) Battery Charge Circuit



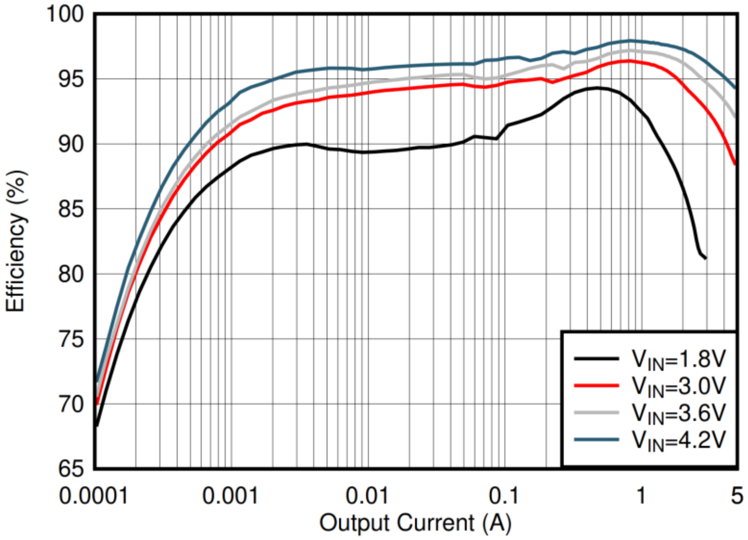
(c) Ideal Diodes for Load Sharing and Battery Isolation while Charging

Figure 4.6 - Battery Charging and Isolation Circuitry

4.2.4 Voltage Regulation

4.2.4.1 Battery to System

With a 2.4 V battery pack voltage and multiple electronics which require a nominal voltage of 3.3 V, a boost regulator is required for the proper functioning of the MachMoS device. Due to the relatively light duty operation of the device, the regulator must be rated for good efficiency at low loads of 10 µA and relatively high loads of 500 mA. The *Texas Instruments* WeBench utility can be used to find a good, low-cost, simple and efficient topology for this application [184]. Using 1.8 V to 3 V (2.4 V nominal) as input and an output of 3.3 V, 500 mA max (30 mA nominal), the design utility recommends a simple boost passthrough topology using the TPS61022 IC, 1 input capacitor, 2 output capacitors, 2 feedback resistors and, of course, an inductor. Since the regulator can operate in auto pulse-frequency modulation (PFM) mode, it is relatively efficient across the load range (as opposed to a forced pulse width modulation mode of operation, where light load efficiency is close to 0 %). The regulator efficiency is not specified for this specific application and cannot be simulated by the WeBench designer as the regulator would be running in auto-PFM, which does not have a model. This mode is, however, tested for a similar voltage gap in the datasheet, 4.2 V to 5 V (0.8 V boost as opposed to 0.9 V for the MachMoS device), and the results are given in Figure 4.7 - TPS61022 Rated Efficiency [185].



V_{IN} = 1.8 V, 3 V, 3.6 V, 4.2 V; V_{OUT} = 5 V

Figure 4.7 - TPS61022 Rated Efficiency [185]

Unfortunately, the efficiency is not given for extremely light loads such as the 5.75 µA load during the *sleep* task (100 µA being the lowest rated load). Therefore, it is important to build a model for extremely light loads. The details of this process are provided in Appendix C.2. The resulting model is shown in Figure 4.8, where the efficiency at 5.75 µA is determined to be approx. 12 %.

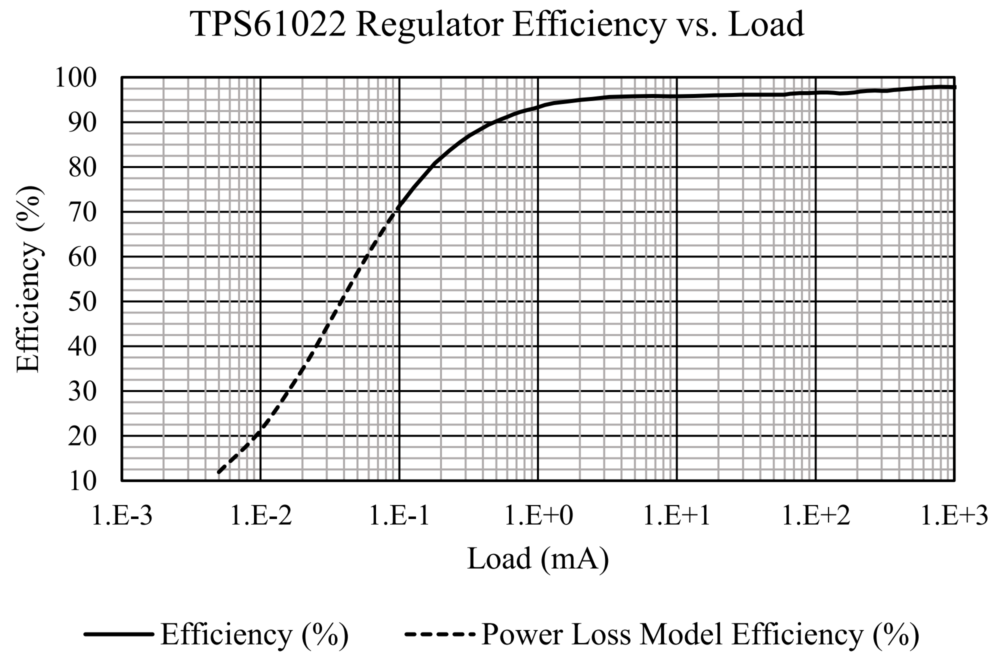


Figure 4.8 - TPS61022 Extended Efficiency

The regulator circuit is shown in Figure 4.9.

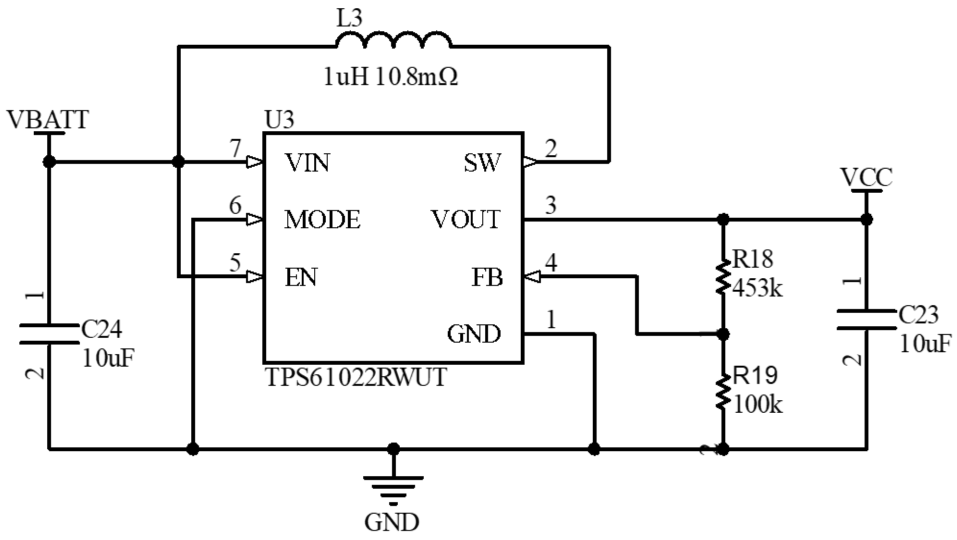


Figure 4.9 - TPS61022 Regulator Circuit

4.2.4.2 USB to Charge Circuitry and System

With an input of 5 V and an output (system) voltage of 3.3 V, the USB to charge circuitry regulator must be a buck regulator. The regulator chosen for this application is the *Texas Instruments* TPS564242DRLR. This regulator is simple to implement and supports input voltages from 3 V to 16 V, leaving lots of margin for immunity to inevitable transients from connecting and disconnecting the USB cabling. The implementation of this regulator is shown in Figure 4.6 (see subsection 4.2.3.3) [186].

4.3 Enclosure

The following subsections will discuss the considerations for the design of the enclosure. Specifically, considerations regarding materials choices, heat transfer characteristics, ingress protection, mounting of the device, as well as other considerations are discussed.

4.3.1 Materials

The material used in the construction of the MachMoS enclosure is of course of great interest. It is important to base the final choice on cost, machinability, heat transfer characteristics, stiffness, and wireless signal attenuation (transmittance). Heat transfer and stiffness are both very important to ensure a good SNR for the accelerometer and the RTD.

If the enclosure material interfacing the machine under test (MUT) does not adequately transfer heat, the temperature differential between the machine and the sensor will be greater. As such, an enclosure material that does not conduct heat properly will require higher temperatures at the MUT to register a significant temperature difference from baseline (healthy) data.

If the chosen material is not stiff, then the signal transmission path from the bearing to the accelerometer will be negatively impacted. In addition, enclosures, like any other object, will resonate when vibrated. It is important to use stiffer materials to ensure the resonant frequencies of the device remains above the frequencies of interest. Otherwise, resonant frequencies will introduce noise in the accelerometer captures [108].

Because of their good heat transfer and stiffness, metals such as aluminum, stainless steel and steel are good candidates as opposed to polymers. However, due to difficulty in machining stainless steel and its relatively high cost, stainless-steel is not considered a good option. Based on heat transfer and machinability, aluminum is a better choice, at the expense of a lower stiffness. It should be noted that the use of steel as an enclosure material requires that some sort of rust-prevention mechanism be used once the device is produced, leading to extra complexity in the manufacturing of the enclosure. The typical post-processing steps required for aluminum enclosures: anodising, coating, or painting, are optional for a functional enclosure, and such, aluminum remains the best material, pending simulations and testing to ensure the lower stiffness does not lead to noise in the captures (see sections 4.7.2 and 5.2.2).

While metals are desired between the MUT and the sensors, the rest of the enclosure can, and should, be made of plastic to avoid the enclosure acting as a Faraday cage and interfering with the BLE signal from the sensor. Considering the current low volume production of this sensor, and the desire to have small research groups manufacturing this device, the plastic component(s) of the enclosure should be 3D printed. 3D printers have made their way into most research institutions in the past few years, with very reliable, low-cost printers now being widely available. In terms of materials, polylactic acid (PLA), acrylonitrile butadiene styrene (ABS), polyethylene terephthalate glycol (PET-G), and aliphatic polyamides (nylon) are most common. However, PLA and PET-G have relatively low maximum service temperatures, at approximately 50°C and 70°C, respectively. ABS and nylon have higher service temperatures (approximately 100°C) and are impact resistant, making them suitable for this application. Polycarbonate (for higher temperature resistance) and polypropylene may also be used, though these materials are considered more difficult to print with, are more expensive, and require hotter nozzle and bed temperatures [187]. In humid environments, nylon might not be a good candidate due to its hygroscopicity. At high relative humidities (RH), thin nylon objects can suffer expansion due to an increase in plasticity. The expansion coefficient can be approximated using equation (13).

$$\Delta l = 1571\varphi^2 - 13.52\varphi \tag{13}$$

where Δl is the expansion coefficient, in ppm, and φ the relative humidity, in %. For instance, at 85 % RH, an expansion of 1100 ppm (1.1 mm/m) is observable. This amount of expansion can negatively affect the seals between the enclosure components [188], [189]. Finally, for outside use, the ultraviolet energy resistant version of ABS may be used. Specifically, acrylonitrile styrene acrylate (ASA) may be used, which shares most of ABS's positive properties. This makes ASA the polymer of choice for the MachMoS node enclosure.

4.3.2 Heat Transfer

Heat transfer properties of the enclosure are very important due to the requirement for a temperature sensor. Since the temperature sensor for the node is placed inside the enclosure, there must be good thermal coupling between the MUT and the enclosure's interior. This ensures that the node's RTD measures a temperature that is approximately that of the MUT. The electronics in the design will consume power in the order of milli-Watts and will not contribute to the overall heat load. Consequently, the material choice for the node's enclosure, as well as its geometry will affect heat transfer to the environment. Heat transfer from the enclosure to the environment should be minimized, as it will lead to errors in temperature measurements.

A polished surface is recommended to reduce the effective surface area for heat transfer and to promote laminar flow over the surface. Heat transfer coefficients are lower in laminar flow than in turbulent flows. A polished surface is also advantageous for mounting directly onto the MUT using thermal epoxy, as heat conduction to the MUT would be enhanced in this configuration. These considerations ensure an optimal SNR for the RTD.

4.3.3 Ingress Protection

Per specifications S.3 and S.4, the MachMoS node should meet an IP68 rating and have an operational temperature range of -40°C to 120°C. Since the node is required to have a long operational life, it is important that the batteries be replaceable. Due to this, the node was split into two parts sealed using O-rings. The McMaster-Carr Supply Company is a large American hardware supplier (McMaster-Carr, McM-C) [190]. Its catalogue is a useful resource to find well-documented standard Commercial Off the Shelf (COTS) hardware. The most common material for oil, fuel and water-resistant O-rings is a co-polymer of butadiene and acrylonitrile commonly named Buna-N [191]. McMaster-Carr features a large selection of standard-size Buna-N O-rings, many of which support the operational temperature of the MachMoS node.

Other considerations for ingress protection stem from the external connector and the orifice that must be included in the enclosure to accommodate the microphone in the design. While IP protection is provided by the external connector as discussed in section 4.2.2, the microphone orifice has not yet been discussed. Due to the orifice that is included in the device enclosure to ensure a good microphone SNR, a specific seal must be included in the design. In this case, a *GORE*® AVP4341.53.5 acoustic vent is used. This type of vent protects the device from water and dust ingress up to IP68 (2 m under water for 30 minutes), while allowing sound to be transmitted with at most 2 dB of insertion loss at 1 kHz [192].

4.3.4 Mounting

An important target specification in the design is ease of installation (S.13). To ensure a good signal transmission path from bearing to enclosure, it is generally recommended to install industrial vibration sensors using grease fitting holes. Grease fittings are used to add lubricants to industrial machinery without having to stop the machinery. A pressurized grease source is pressed against a valve with a captive ball and grease is pushed through to the machinery. Grease fittings are typically installed on machinery using a pipe thread (National Pipe Taper series of standard threads, NPT). These threads are machined into the machine's enclosure, in-line with the bearings in the enclosure, making them excellent for the installation of vibration sensors. If a threaded connection is not available, it is acceptable to use a structural adhesive to mount the sensor to the enclosure, or a strong permanent magnet, though the latter is least favourable, and only available for use on ferrous enclosures.

The MachMoS node must be mounted on many types of machinery. Thus, it is imperative to select an NPT thread size on the node that can be easily adapted to other thread sizes on the machinery. A brief search through the company's catalogue for COTS male fittings shows that 1/8-NPT adapters are very common and most likely to have adapters to other thread sizes. Table 4.4 lists some common adapters found in the McMaster-Carr catalogue that can adapt 1/8-NPT to the NPT, United National Coarse (UNC), and United National Fine (UNF) thread series.

Table 4.4 - 1/8 NPT Male-Male COTS Reducers and Adapters [193]

1/8-NPT Male to Straight Male Threads (1/8-NPT to UNC/UNF)		**1/8-NPT Male to Pipe Male Threads (1/8-NPT to NPT)**	
Adapts to	**McMaster-Carr P/N**	**Adapts to**	**McMaster-Carr P/N**
M5x0.8	1859N178	1/16	5485K37
M6x1.0	1859N179	1/8	5485K21
M8x1.0	1859N181	1/4	5485K37
M10x1.25	1859N182	3/8	5485K35
.1900"-32	1859N163	1/2	50785K559
.2500"-28	1859N164	3/4	50785K561
.4375"-20	1859N166		

4.3.5 Other Considerations

In this subsection, other considerations relating to the weight and size of the node, as well as accommodations for the microphone are discussed.

4.3.5.1 Weight and Size of the Node

While no target specifications are set for the weight and the dimensions of the device, these must be kept reasonable. The device should be as small as possible to ensure that the device can be used on small machines or in situations where there is not much space available around the machinery.

In the case of the MachMoS node, maintaining a small form factor is challenging. NiMH batteries, a battery chemistry that was chosen due to the temperature range of the device, are typically packaged in AAA and AA sizes. Odd sizes, such as pouches, common to Li-Ion or Li-Polymer batteries, are not available for NiMH chemistries. A flat enclosure, in which the NiMH cells are installed horizontally, was deemed preferrable as it restricts the amount by which the node protrudes from the machinery. A larger, flat enclosure additionally allows for leverage when tightening the sensor on an MUT by hand instead of requiring a wrench during installation. From battery life calculations, it was determined that four AA cells would provide adequate device life (see section 4.2.3). Due to the form factor of the cells, the presence of a stem in the middle of the device, as well as the horizontal orientation of the cells and the use of an MCU module (ESP32-WROOM), it is challenging to obtain a small enclosure. The first prototype of the node measures 120 mm in diameter and has a height of 76 mm. Its mass is approximately 750 grams. Smaller cells could be used, but those which have better packing factors, such as 1/2AA are more expensive and difficult to source.

4.3.5.2 Microphone Accommodations

An orifice through the enclosure was included to provide good interfacing of the microphone to the environment. As discussed in section 4.3.3, a *GORE*® vent was added to block the orifice, protecting the microphone from the environment. An O-ring was introduced to seal the microphone from the air inside the enclosure, and to minimize unwanted vibrations from its base. Details of the orifice and O-ring are shown in Figure 4.10. The presence of such an opening between the gore vent and the environment introduces the risk of creating a Helmholtz resonator, which would introduce significant noise to the signal if excited. The resonator cavity and resonator opening are shown in Figure 4.10. The length of the resonator cavity and resonator are L_c and L_o, respectively.

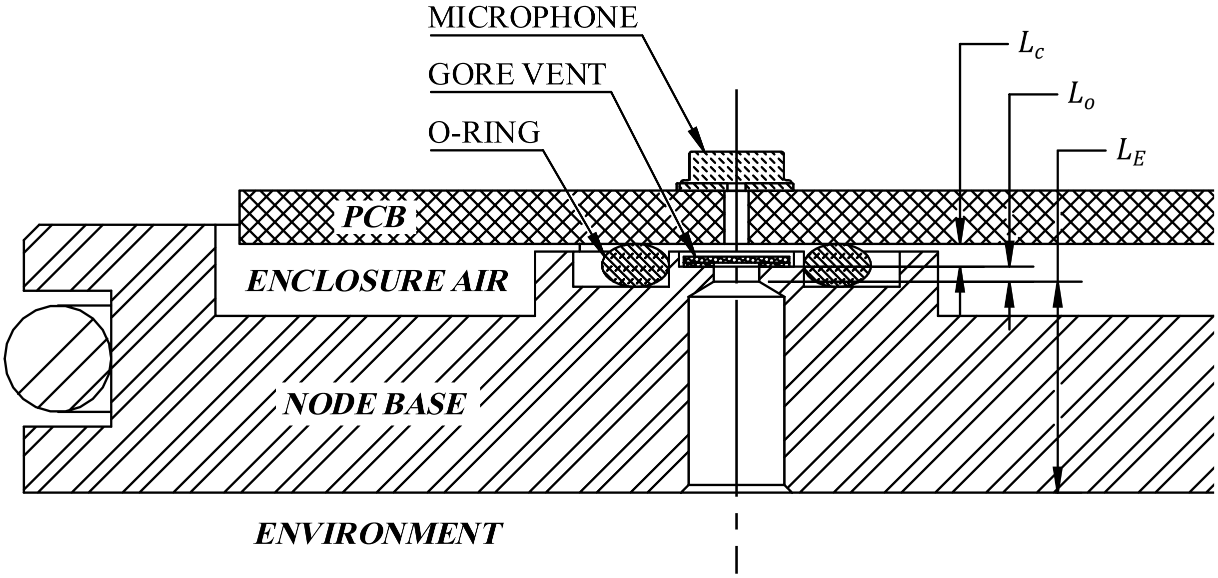


Figure 4.10 - Cross-Section of the Microphone Orifice

It is possible to estimate the resonant frequency of the resonator using equation (14).

$$f_r = \frac{c}{2\pi}\sqrt{\frac{A_o}{V_c \cdot (L_o + L_{cf})}} \tag{14}$$

where c is the speed of sound, A_o is the area of the opening, V_c the volume of the cavity given by multiplying the length of the cavity L_c by its area A_c, L_o the length of the opening, L_{cf} the length correction factor, obtained by multiplying the diameter of the opening by 0.3, and f_r the resonator frequency [194]. In this design, the resonator frequency must be greater than the highest frequency measurable by the microphone (21.4 kHz, as discussed in section 4.1.3). Otherwise, the noise introduced by the resonator will interfere with the measurement. As such, it is important to use the device's lowest operational temperature, which corresponds to the lowest c, as a lower c will reduce the resonator frequency. At the node's lowest operational temperature (-40°C), the speed of sound in air can be calculated to be approximately 306 m/s [88]. Using the values for f_r and c, equation (15) is obtained.

$$\frac{A_o}{V_c \cdot \left(L_o + l_{cf}\right)} > 1.93\mathrm{E}^5\ [\mathrm{m}^{-2}] \tag{15}$$

In the case of the MachMoS node, the resonator cavity area is at least that of the GORE® vent used for ingress protection (3.75mm outer diameter, OD), and at most approximately the inner area of the O-ring used (4.5mm inner diameter, ID). That is, the area of the resonator cross-section (A_c) is between 44 mm^2 and 64 mm^2. The worst-case resonator is the largest one as it leads to a smaller term in the inequality. As for the area of the opening, it should be maximized. However, it is at most 1.5 mm due to the ID of the *GORE®* vent. That is, the length correction factor L_{cf} is 0.45 mm and the area A_o is approximately 7.1 mm^2. Substituting these values into equation (15), a restriction on resonator length and opening can be obtained.

$$L_c \cdot (L_o + 0.45) < 0.575\ [\mathrm{mm}^2] \tag{16}$$

where L_c is the cavity length. Per the inequality, the length of the resonator cavity and opening must be small. For instance, if the length of the cavity is 0.5 mm, then its opening must be shorter than 0.7 mm. Due to this, (and as can be seen in Figure 4.10), the acoustic vent is as close to the printed circuit board as possible, and the opening of the resonator is preceded by an enlargement of length L_E.

4.4 Enclosure CAD Modeling

Using all the considerations discussed in the previous sections of this chapter, the mechanical enclosure of the MachMoS node was designed. The considerations for this part of the design were as follows: The node enclosure shall:

- be made of an aluminum bottom section (interfacing with the MUT).
- feature a pass-through in the middle of the enclosure terminating with a 1/8-NPT female thread.
- feature an IPX7-rated seal to a plastic 3D-printed enclosure.
- contain a printed circuit board with mounting points close to the center of the device to maximise the accelerometer's SNR.
- contain four AA cells (⌀14 mm x 50 mm length), placed horizontally. These shall be secured against vibrations.
- make provisions for the *Amphenol* MUSBRM5C1M0 USB-C connector.
- Be preferably round, with a center of gravity on its axis, to promote an equal frequency response in all directions.

When designing the enclosure, the largest components had to be laid out first. This layout work was performed using sketching tools in SolidWorks and included the footprints of the USB-C connector, the ESP32-WROOM, and the battery cells. It was determined that the base of the enclosure had to be at least 4 inches in diameter to accommodate the components. The AA cells could not be placed over the center of the device as the pass-through was in the way. The diameter of the enclosure was slightly increased to 4.5 inches to allow room for an O-Ring groove.

Marco Rubber is a supplier of sealing solutions for multiple industries [195]. Throughout enclosure modeling, the O-ring grooves were designed around O-ring availability in the McMaster-Carr catalogue and the *Marco Rubber* groove design guide [196]. 0-5% O-ring stretch, and 18-25% cross-section reduction rules were used to determine the diameter and depth of the O-ring groove. The 75% fill rule was used to determine the groove width.

With the batteries modeled, it was noted that the contents of the enclosure protruded considerably from the base. Moreover, the pass-through geometry (stem) also protruded considerably. This posed a challenge in manufacturing. The device could be machined as one component from a billet of material, but the material removal would be too considerable due to the stem. It was decided that the most cost-effective solution was to manufacture the base and stem separately, and to press these components together. A FN2 Medium Drive Fit from the ANSI/ASME B4.1-1967 (R2009) standard on Preferred Limits and Fits for Cylindrical Parts [197] was chosen for use in this enclosure. This fit uses fundamental deviation letters H7/s6 and ensures a permanent, leak-free joint between the base and the stem.

The stem's minimum diameter was determined by the 1/8-NPT thread in either one of its ends. A 1/8-NPT thread has a diameter at the top of a female thread of approximately 0.41 inches. It was decided that the cover of the enclosure should thread onto the stem of the device using a captive nut in the 3D-printed cover. An appropriate thread size for the captive nut was determined to be 0.625-18 from the UNF thread series. This thread has a minor diameter of 0.56 inches, leaving at least 0.075 inches of wall at the very top of the NPT thread. This is approximately 2 mm, and not a lot of material for both threads to share the same part of the stem without risking damage to the stem when tightening the enclosure cover. As such, the threads were placed at different heights along the stem.

Proper sealing had to be provided between the stem and the enclosure cover, as the stem protrudes through the enclosure cover. For this purpose, an O-ring groove was added, following the same rules mentioned above. Flats were added to the top of the stem. These are accessible when the enclosure cover is installed, and their purpose is to allow for the use of a small wrench to properly tighten the node cover onto the base. A section view of the stem of the enclosure base is illustrated in Figure 4.11. In the figure, the threads, their positions along the stem, the flats, the fit to the enclosure base, and the O-ring groove (the groove on the left) are clearly illustrated.

Other important features were included in the base of the enclosure, such as a protrusion for the microphone interface (subsection 4.3.5.2), a pocket to space the USB-C connector to the main board and to obtain a proper seal, and a set of locating protrusions to position the RTD breakout board relative to the main board, ensuring a proper alignment between the mezzanine connectors of each board. Breakout boards to fit the MUSBR connector (which is very tall) and the RTD were also modeled in parallel with the enclosure.

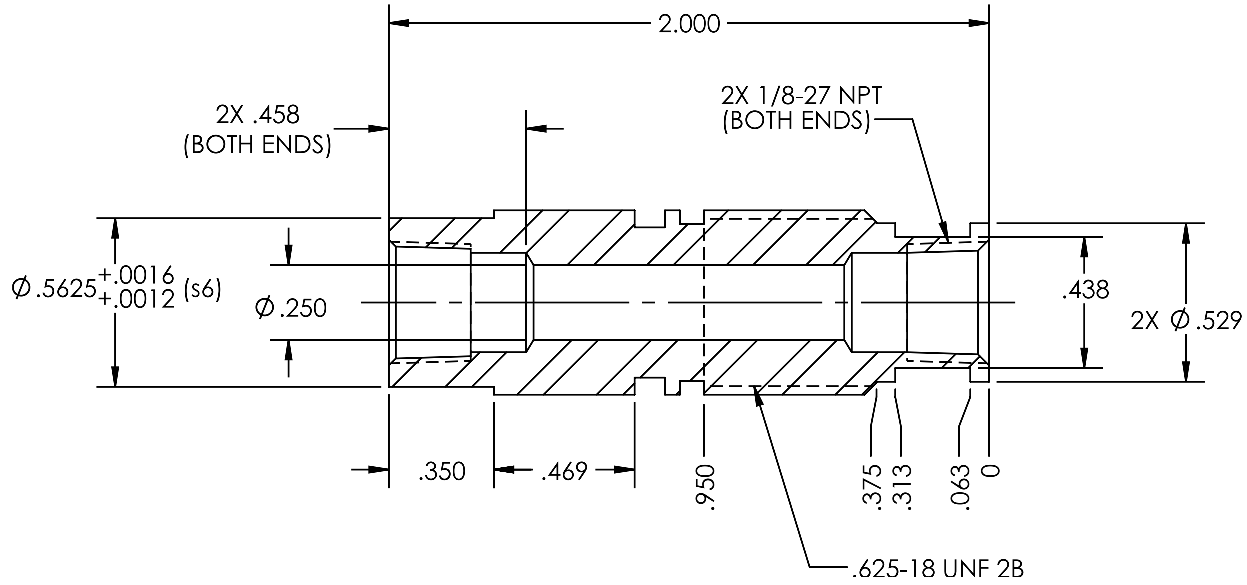


Figure 4.11 - Enclosure Stem (Passthrough). Groove Dimensions Omitted.

The main components of the node are annotated in Figure 4.12. To assemble the node, the PCB is attached to the enclosure base using 8 screws. In this figure, the base and cover of the enclosure are clearly visible. To seal the enclosure, the 3D-printed cover is threaded onto the enclosure base, compressing the radial O-rings against the stem and the base.

4.5 Circuit Board Layout

The layout of the PCB main board and breakout boards were performed once the obstacles and geometry for these were determined in the enclosure modeling step. In the layout stage, the stack-up, component placements, copper pours (regions), traces, and keep-outs are defined. JLCPCB is an accessible Chinese manufacturer of PCBs (JLCPCB). DFM rules from the JLCPCB website [198] were used at this stage to ensure that the board could be manufactured.

It was determined that the MachMoS device could benefit from a 4-layer PCB to ensure signal return planes exist close to the SMPS switch nodes and communication lines (specifically those lines requiring impedance matching, such as those for USB communications). The ideal stack-up was determined to be (from top layer to bottom): signal and ground, ground, power and ground, signal and ground, with the bottom signal layer's traces as far from SMPS regulator switch nodes as possible to prevent noise coupling into those signals.

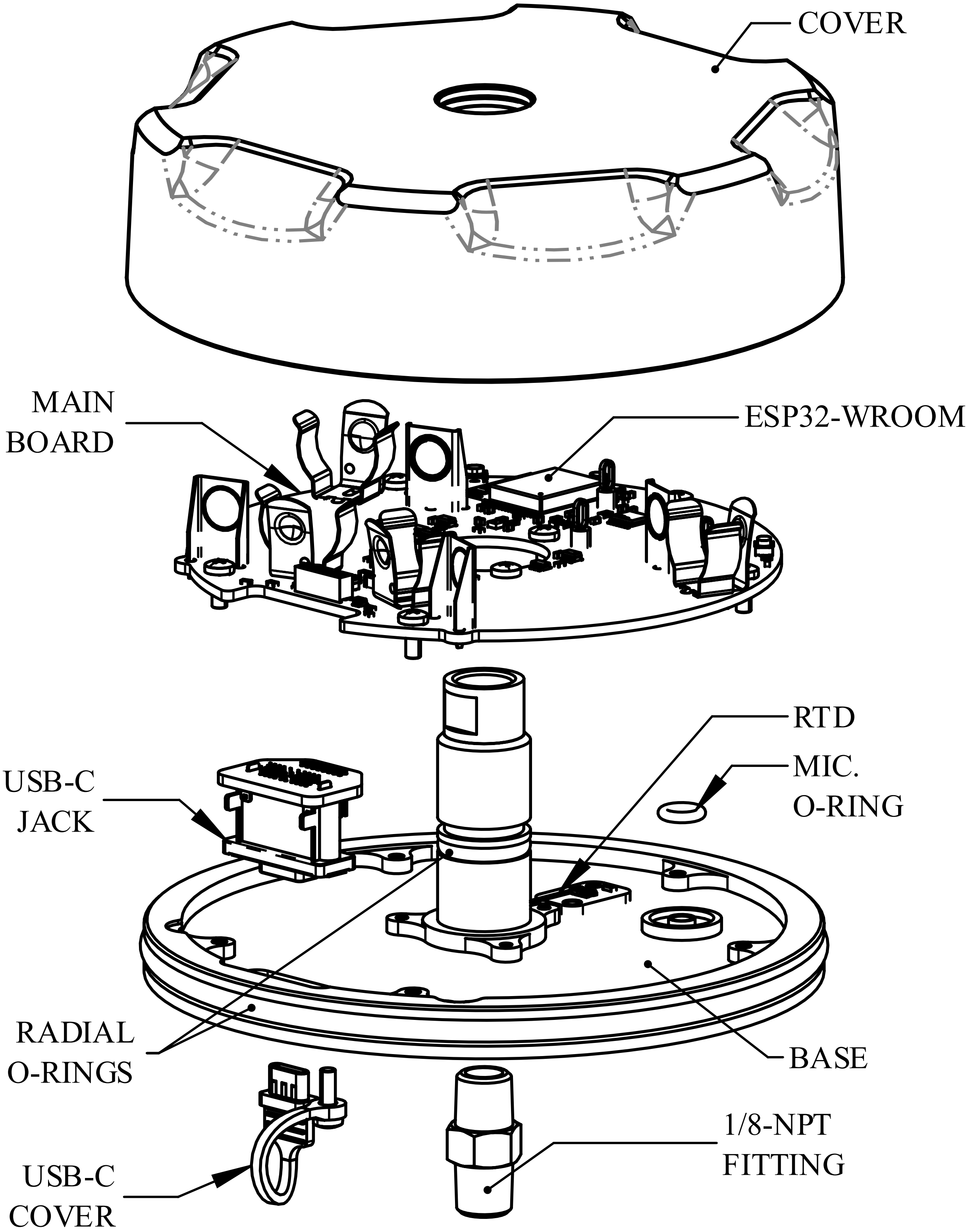


Figure 4.12 - Annotated View of the MachMoS Node. Batteries Hidden for Clarity.

Large copper pours were used for nets which carry power through the PCB. This ensures that a low-inductance path is available for power to flow through, reducing localized voltage drops when power draws are high and reducing the amount of electric noise generated by these nets. Any unused areas in the PCB were poured over and assigned to the ground net. This further ensures that a solid return path is available for the electric fields generated in the PCB. As explained in subsection 4.2.1, a copper keep-out zone was included under the ESP32-WROOM's antenna to ensure that the antenna impedance is not affected by the PCB. Via-stitching was used where possible to promote low-induction power paths.

Where reasonable, thermal reliefs were used between pads and copper pours to facilitate the reflow process used to assemble surface mount components. The mounting points for the main PCB were reinforced using plated vias inside the pads under the screw heads. These pads contact the enclosure base and electrically couple the enclosure to the MachMoS node ground.

The final layouts of the top, middle-top, middle-bottom and bottom layers are illustrated in Figure 4.13, Figure 4.14, Figure 4.15, and Figure 4.16, respectively. In these figures, large copper pours can be seen, with the keep-out zone at the top left of the PCB. The dark-grey areas are ground copper. Five mounting points around the exterior of the board, and three around the cut-out in the middle of the board are shown. The cut-out at the bottom of the board is for the *Amphenol* MUSBR USB-C connector.

The same stack-up and principles were used for the MUSBR connector adapter breakout board. The layout for the board is given in Figure 4.17. In this figure, the top, middle-top, middle-bottom, and bottom layers are shown. The ground pours are clearly visible in the figure, as well as the thermal reliefs and the differential lines for the USB protocol.

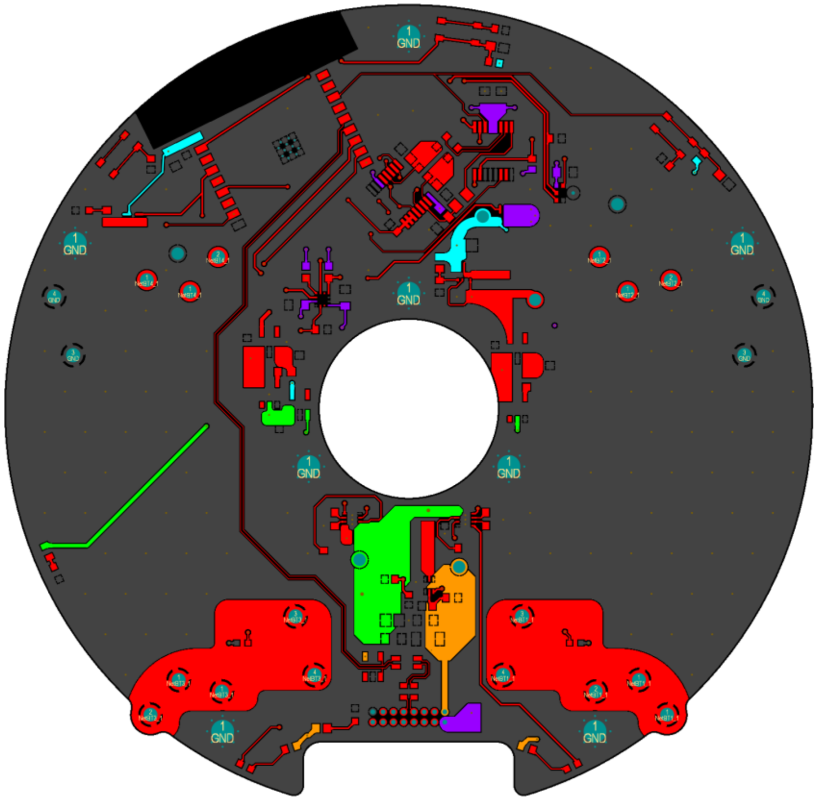


Figure 4.13 - Top Layer (Sig-GND)

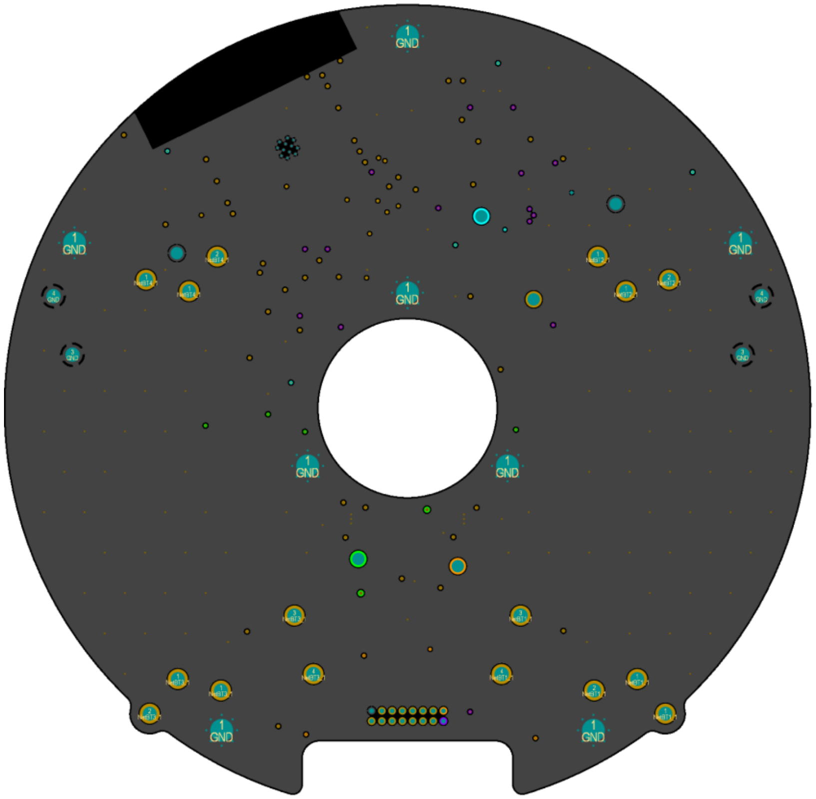


Figure 4.14 - Top-Middle Layer (GND)

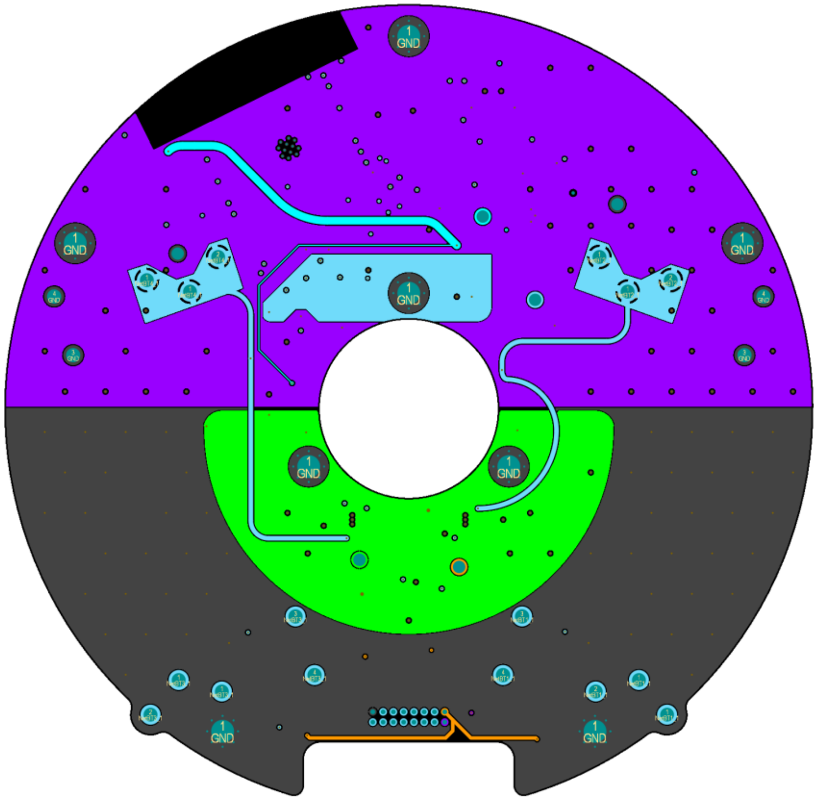


Figure 4.15 - Bottom-Middle Layer (Power-GND)

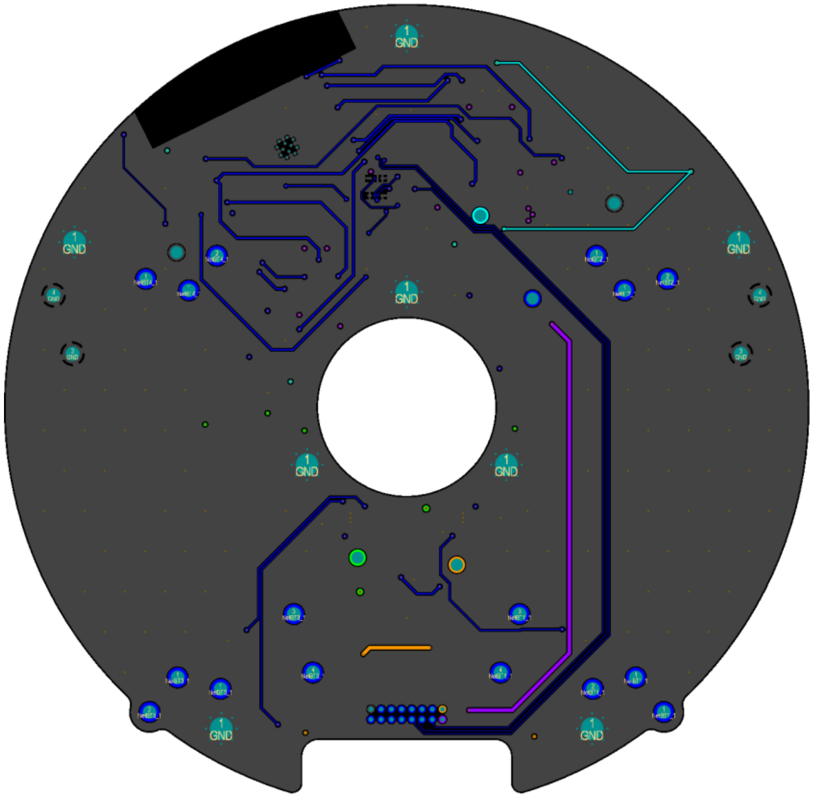
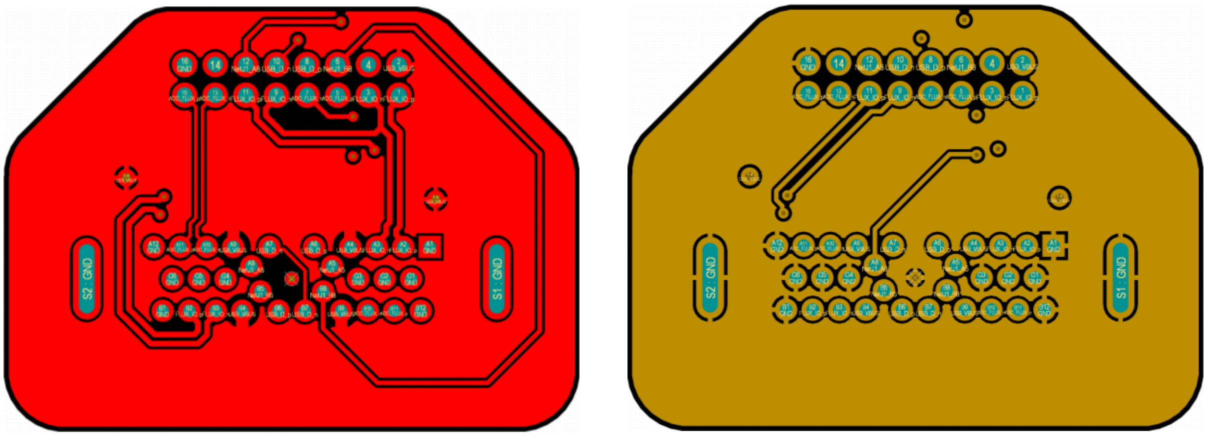
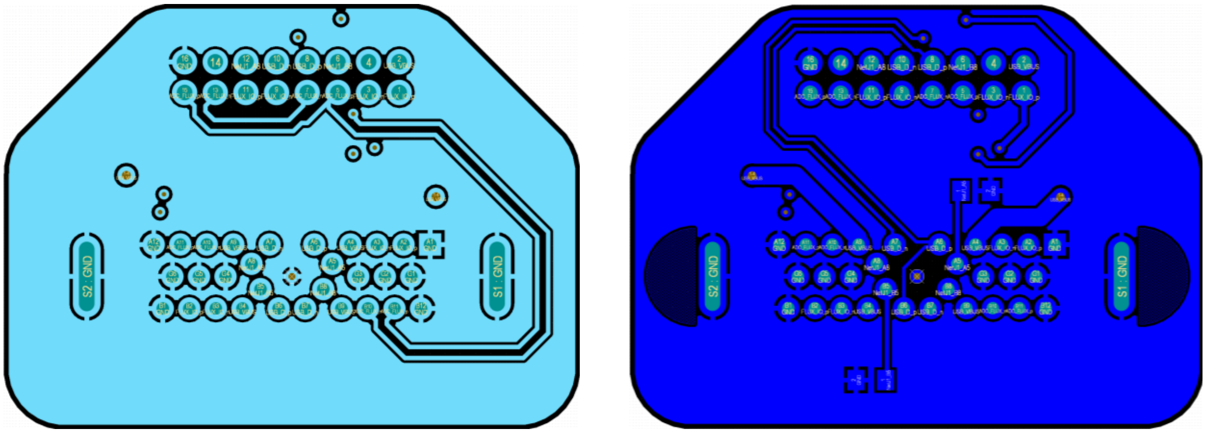


Figure 4.16 - Bottom Layer (Sig-GND)



Top Layer *Middle-Top Layer*



Middle-Bottom Layer *Bottom Layer*

Figure 4.17 - MUSBR Breakout Board Layout

As the RTD breakout board is simple, a 2-layer stack-up was used to save on cost. The layout is shown in Figure 4.18. This figure shows the RTD in grey. The top layer is shown on the left, and the bottom layer on the right of the figure. Since the board is located using a set of pins protruding from the enclosure base, board cutouts were used to define the holes to encourage the milling of these features by the manufacturer (as opposed to drilling).

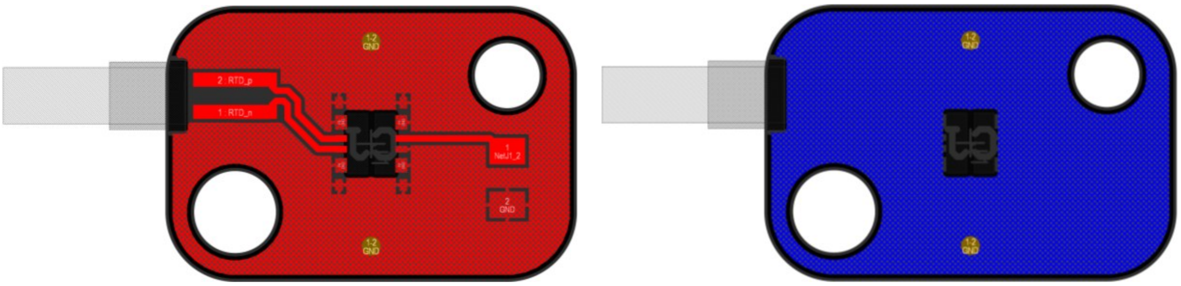


Figure 4.18 - RTD Breakout Board Layout

4.6 Bill of Materials

The bill of materials (BOM) for the electrical, mechanical and software systems of the MachMoS Node are provided in Table 4.5, Table 4.6, and Table 4.7, respectively. The detailed electrical BOM is provided in Appendix E. Taxes are not included. All prices are in CAD.

Table 4.5 - MachMoS Node Electrical Bill of Materials (Categorised)

Category	Count	Cost
Connector	13	35.97
Inductor, Filter	3	5.82
Integrated Circuits	17	71.46
LED, Button, Test Point	14	7.60
Passive	31	12.90
PCB	3	8.20
PCB Assembly (Approx.)	3	30.00
Batteries	4	12.48
	Total:	$ 184.44

Table 4.6 - MachMoS Node Mechanical Bill of Materials

Part Number	Manufacturer/Distributor: Part Descr.	Cost	Qty	Ext.
95911A168	McM-C: M2.5x0.45 x 8 mm Nylok Pan Head	0.69	2	0.69
DDL-PRT-001	DDL: PASSTHROUGH	0.65	1	0.65
	DDL: BASE PLATE	8.72	1	8.72
	DDL: BASE	0.00	1	0.00
8329TFS-25ML	MG Chemical: Thermal Epoxy, 1ml	3.18	1	3.18
AVP**23**41.53.5 or **43**	W.L. Gore & Assoc., Inc.: Acoustic Vent	0.50	1	0.50
9452K194	McM-C: Dash 244 O-Ring, Buna-N	0.59	1	0.59
9452K58	McM-C: Dash 014 O-Ring, Buna-N	0.04	1	0.04
9452K16	McM-C: Dash 008 O-Ring, Buna-N	0.04	1	0.04
95911A164	McM-C: M2.5x0.45 x 6 mm Nylok Pan Head	0.68	8	6.08
DDL-PRT-002	DDL: COVER	2.53	1	2.53
91078A235	McM-C: 5/8"-18 Hex Nut, Thin, Zn Plated	0.79	1	0.79
5485K21, 4757T71	McM-C: 1/8-NPT Brass Fitting and Plug	6.76	1	6.76
MUSBR2M5C004BP	Amphenol: USB-C Jack Dust Cover	4.25	1	4.25
			Total:	$ 34.82

Table 4.7 - MachMoS Node External Software Bill of Materials

Library	Link
Arduino Core API	github.com/arduino/ArduinoCore-API
rwagoner ADS1120 Library	github.com/rwagoner/ADS1120-Library
InvenSense IIM-423XX	github.com/InvenSenseInc/public.mcu.iim42352
SparkFun KX13X	github.com/sparkfun/SparkFun_KX13X_Arduino_Library

As can be seen in the tables, the total cost of the MachMoS node ($ 219.26 CAD) is low compared to other commercial devices. It is therefore reasonable to assume that the device meets specification S.9. The largest costs are the KX134-1211, MUSBR, and ADC ($ 58.79 CAD).

4.7 Calculations and Simulations

Once the mechanical shape and materials of the node were defined, sufficient information was available to perform heat transfer and vibration simulations.

4.7.1 Heat Transfer

Heat transfer simulations were performed on two enclosure base materials and three orientations to determine the temperature discrepancy between the temperature that is read by the MachMoS device, and the actual MUT casing temperature. The simulations assume that the device is stud-mounted with a brass fitting, and that the only significant loss of heat happens at the base of the enclosure, from natural convection (the rest of the device being made of, or being covered with plastic, it is assumed to be well-insulated).

The simulations were performed for ambient temperatures ranging from -40°C to 50°C, and MUT temperatures from 10°C over ambient to 120°C. The increments on ambient and MUT temperatures were 20°C and 10°C, respectively. The temperature differences obtained through simulation for an ambient temperature of -40°C are given in Figure 4.19. In this figure, the results for the horizontal plates, facing down, are not shown, as they are similar to that of the vertical plates (within 0.2°C).

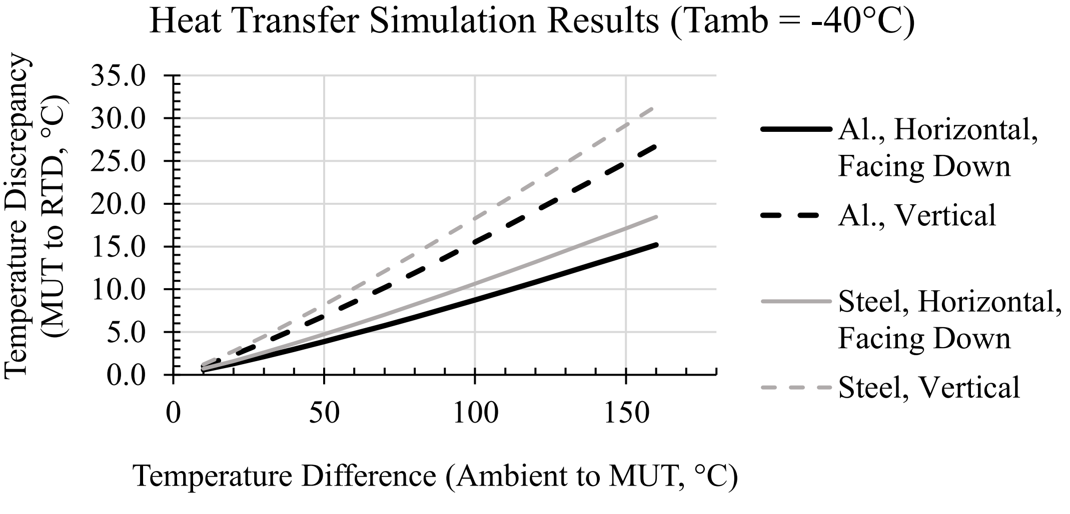


Figure 4.19 - Temperature Discrepancy: MUT Casing vs. Measured

As can be seen from the simulation results, an aluminum base leads to a smaller temperature difference between the MUT and the measured temperature, making it a better option. There can be some significant temperature differences between the machine temperature and the measured temperature, though these figures are conservative. The simulation does not consider the warmer air surrounding the MUT (the effective ambient temperature), which would make this temperature difference less severe.

The exact parameters and results for these simulations are given in Appendix A.4. For reference, the appendix also includes equations derived from the results which provide an estimate of the temperature discrepancy from the measured and ambient temperature. These equations can serve as correction factors, depending on availability of ambient temperature data.

4.7.2 Vibration

When designing accelerometer enclosures, the recommendation is that they be small, light, and stiff along each measured axis [108]. This ensures that the lowest resonating frequency (mode) of the enclosure is outside the bandwidth of the accelerometer. Unfortunately, this was not possible for the MachMoS node. The node is large compared to other commercially available accelerometers, and the accelerometer used in the device is tri-axial, such that stiffness would need to be optimized in all directions. Due to this, it is important to use vibration simulations to determine the resonating modes of the node. Using the results of the simulations, the noise in vibration measurements obtained during testing could be more accurately identified. Finally, these simulations can tell whether the benefits of a stiffer but heavier steel base for the enclosure outweigh the additional complexity in manufacturing.

To obtain an estimate of the resonant modes of the enclosure, a SolidWorks vibration simulation was performed. The bottom threads of the NPT adapter under the node is the interface to the MUT. As such, they were fixed in the simulation. The model was also simplified to reduce the number of degrees of freedom in the analysis. This was done by excluding overly complex geometry that wouldn't contribute any significant change in the device's resonant frequency, including:

- Screw head drive geometry
- PCB holes, except for screw holes
- PCB components (except for the NiMH cells and mounting tabs as these are of considerable mass)
- Enclosure USB-C port, microphone orifice and geometry, RTD geometry.

O-rings were not included in the simulation due to the added complexity accurately modeling the rubber material for vibration analyses. The assumption is that O-rings would simply dampen response to vibrations. The simulation was conducted for frequencies from 0 Hz to 25.6 kHz. A mesh refinement analysis was used to ensure that the simulation results were independent of the mesh density. In the end, simulations were conducted using an adaptive tetrahedral mesh with element sizes ranging from 0.01 inches to 0.125 inches and a growth ratio of 1.4. An example of the vibration study results is provided in Figure 4.20, where a deformation plot with the resultant amplitude (AMPRES) is overlayed.

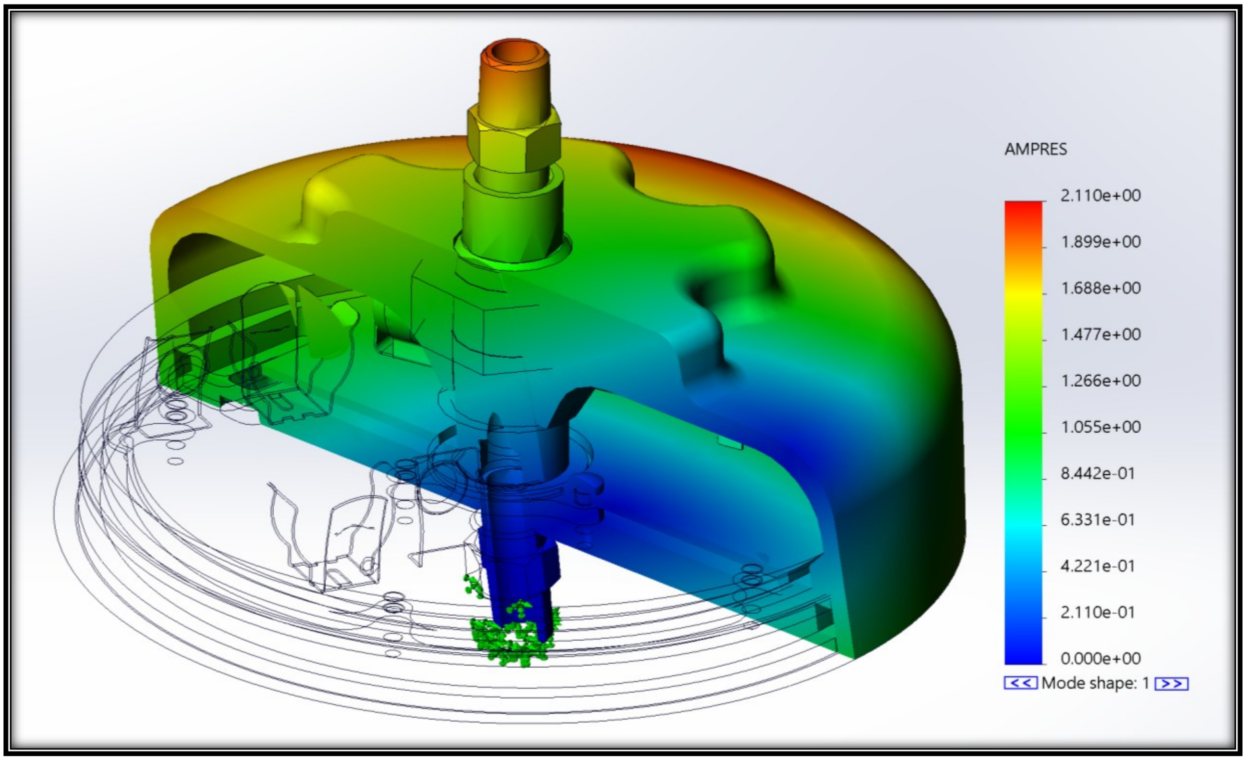


Figure 4.20 - Example modal analysis result showing the AMPRES within the sensor

From the results, only the first few modes, between 0 and 4 kHz, show significant effective mass participation factors (EMPF). The EMPF at those frequencies cumulate to 85 % of the EMPF. The AMPRES for those modes, as measured under the accelerometer, are provided in Figure 4.21.

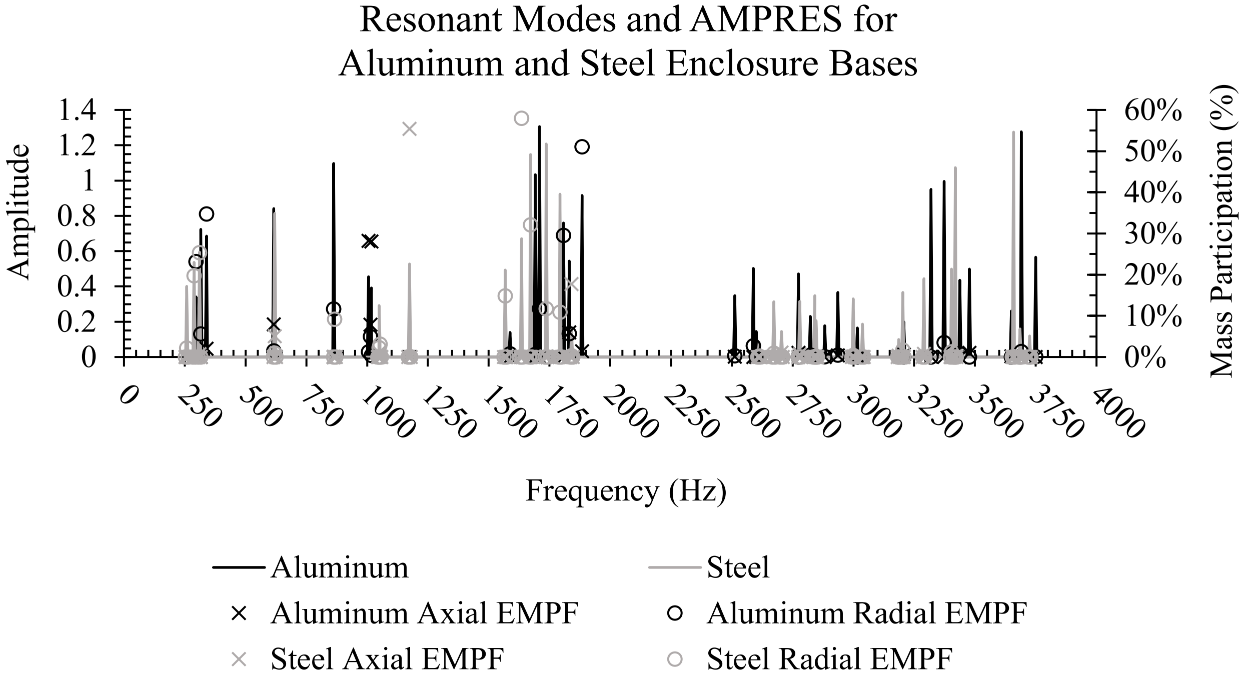


Figure 4.21 - Modes, AMPRES and EMPFs for Aluminum and Steel Enclosure Bases

The modes shown (including their EMPFs) are not much different between the steel and aluminum enclosures. This points to the brass fitting between the enclosure and the MUT being the biggest contributor to low-frequency resonance in the enclosure, and the enclosure base material being of little consequence. This confirms that aluminum is the better choice for the material of the enclosure (see section 4.3.1). To show this, the vibration simulation is repeated for a stainless-steel fitting. The results are shown in Figure 4.22. In this figure, the use of a stainless-steel fitting significantly increases the modal frequencies of the device. However, the modes remain well within the bandwidth of the MachMoS device, such that the use of a stainless-steel fittings would only contribute an added cost to the device.

It should be noted that AMPRES is a unitless measure of relative amplitude within a mode. Testing will ultimately show if the amplitude of the resonance encountered is significant enough to result in measurable noise.

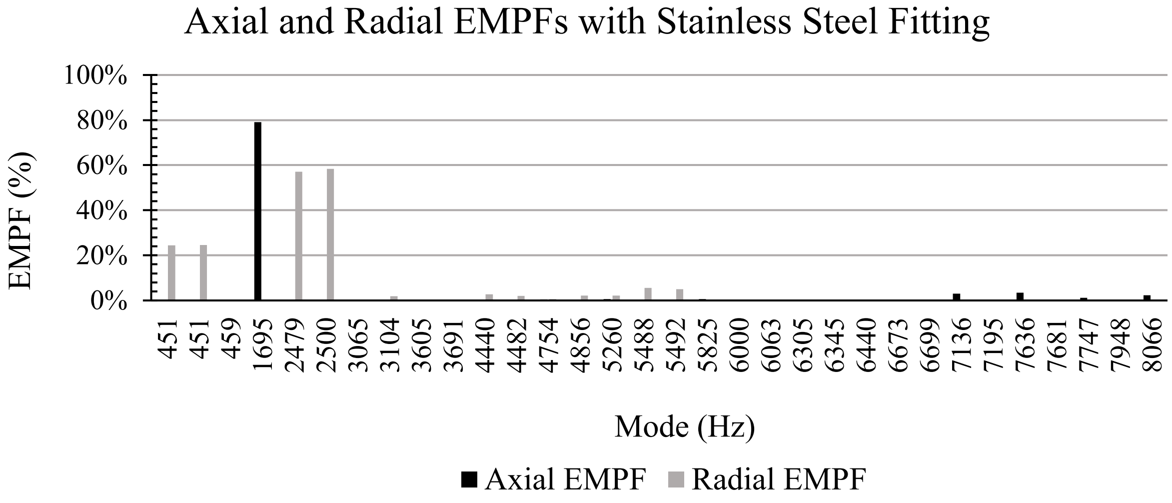


Figure 4.22 - EMPFs with a Stainless-Steel Fitting

4.8 Construction

The PCBs of the MachMoS node were manufactured by JLCPCB using the manufacturing data exported from the ECAD software. The individual electrical components were purchased and soldered to the PCB by the author.

The base of the MachMoS node was machined using a Computer Numerical Control (CNC) 3-axis mill (*Tormach* PCNC 1100). The tool-pathing for the CNC mill was generated by the author and post-processed into g-code using Computer-Aided Manufacturing (CAM) software. The stem of the MachMoS node was machined on a manual lathe. All machining was performed by the author at the University of Ottawa's Brunsfield Centre. The lid of the MachMoS node was 3D-printed in the DDL using a *Bambu Labs* X1 Carbon printer. The manufacturing drawings are provided in Appendix D.

Chapter 5 Results and Discussion

5.1 Temperature Sensor Accuracy Testing

The results for the temperature sensor accuracy test are shown in Figure 5.1. The figure shows the expected behaviour, where a plateau is discernable at 0°C. The plateau is situated between -0.25°C and 0°C and shows that the test was successful. Such variation can be attributed to the error propagation calculated in Appendix A.2 (±0.25°C at 0°C). This indicates that the MachMoS node's temperature measurements are accurate within ±0.25°C and meet specification S.5. The test can also be used as partial validation of the MachMoS node's operational temperature range (S.4), as the test was conducted down to -15°C (not shown in Figure 5.1), with no interruption of the device's data logging.

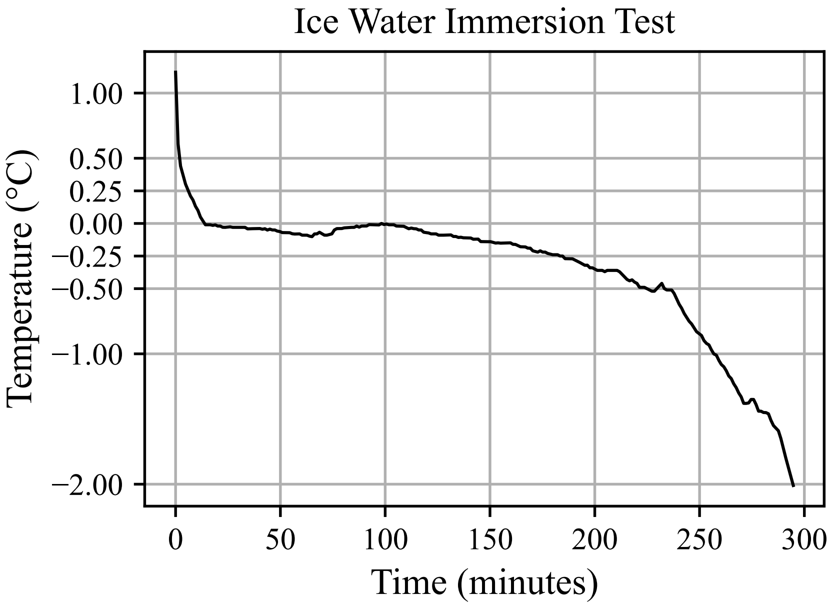


Figure 5.1 - Ice Water Immersion Test

5.2 Accelerometer and Microphone

The following section will discuss the testing of the accelerometer device (KX143-1211) and the microphone device (ICS-43434).

5.2.1 Maximum Achievable Data Rate

It was found during data rate testing that the theoretical maximum data rate of 25.6 kHz for the accelerometer, and 51.2 kHz could not be achieved by the MachMoS node. This is most likely due to limited RAM and an insufficient data transfer rate to the node's flash memory. In addition to this, the maximum achievable data rate is also determined by the code running the MachMoS node. The code was developed by a person contracted by the DDL. Tests were performed by this person as per section 3.2.2.

The maximum achievable data rate of the MachMoS device was found to be 12.8 kHz for the accelerometer, and 21.8 kHz for the microphone. This data rate is sufficient to perform vibration testing and meets target specification S.1 of this work. However, it does not reflect the full potential of the components in the MachMoS node.

A detailed discussion on data generation and transfer rates is provided in Appendix B.3. The appendix recommends an SPI RAM IC or multiple flash ICs be included in the node to increase its maximum data rate and storage capacity if the software of the device cannot be further optimized to accommodate for higher data rates.

5.2.2 Anti-Aliasing Testing

When choosing the KX134-1211 (MachMoS accelerometer) and ICS-43434 (MachMoS microphone) sensors, care was taken to ensure that these were equipped with anti-aliasing filters. While the ICS-43434 was clear about its implementation of the filter before decimation, it was not entirely clear if the KX134-1211 was implementing its low-pass filter before or after digitisation (and decimation), stressing the importance of the anti-aliasing test. On the accelerometer, all tests were conducted using the default low-pass filter which has a -3 dB cut-off of ODR/9 (1.42 kHz). On the microphone, this filter cannot be modified, and its -3 dB cut-off is set to 41.7 % of the ODR (equivalent to ODR/2.4).

Using the maximum achievable sampling rate (f_s) of 12.8 kHz for the accelerometer, the maximum frequency that can be captured ($f_{\max}$) as determined by the Nyquist-Shannon theorem is 6.4 kHz ($f_{\max} = f_s/2$). In the results, the exciter was set to 600 Hz higher than $f_{\max}$, or to 7 kHz. The input voltage amplitude was set to 10 V peak to peak (pk-pk). As per the Nyquist-Shannon theorem, an unfiltered signal that is 600 Hz higher in frequency than $f_{\max}$ causes an aliased peak to be folded into the captured signal 600 Hz below $f_{\max}$. In this case, at 5.8 kHz. Figure 5.2 shows that the input frequency is set properly to 7 Hz.

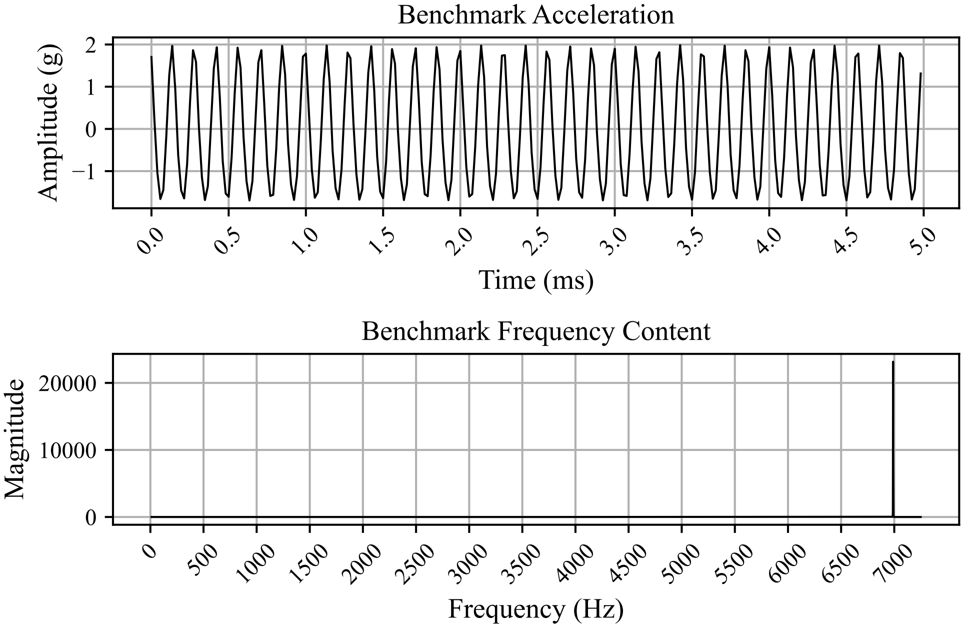


Figure 5.2 - Aliasing Testing: Benchmark

For all three axes of the MachMoS capture, a folded peak was observed at 5.899 kHz. For the X and Y axes, the peak amplitude was found to be smaller than the noise floor. Therefore, it is reasonable to ignore it. This small peak is shown in Figure 5.3.

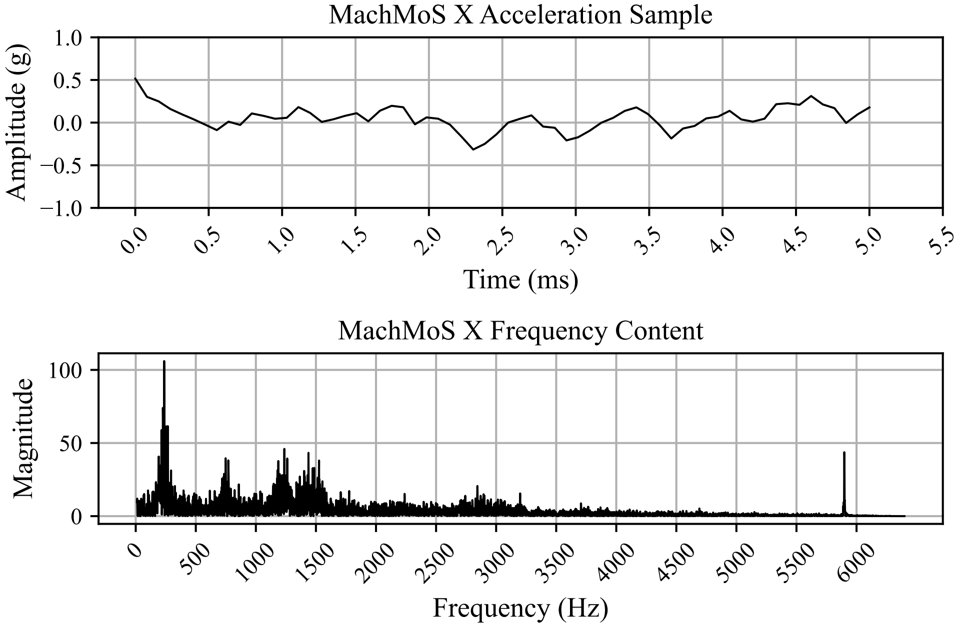


Figure 5.3 - Aliasing Testing: X-Axis. Y-Axis similar (not shown).

However, Figure 5.4 shows that the peak in the Z-axis is much higher in magnitude than the noise floor.

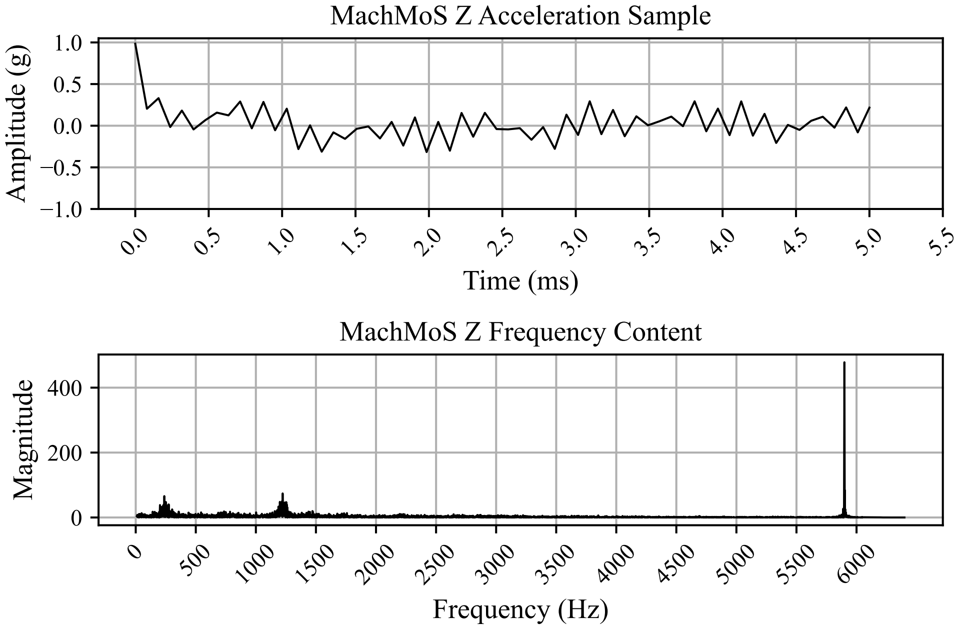


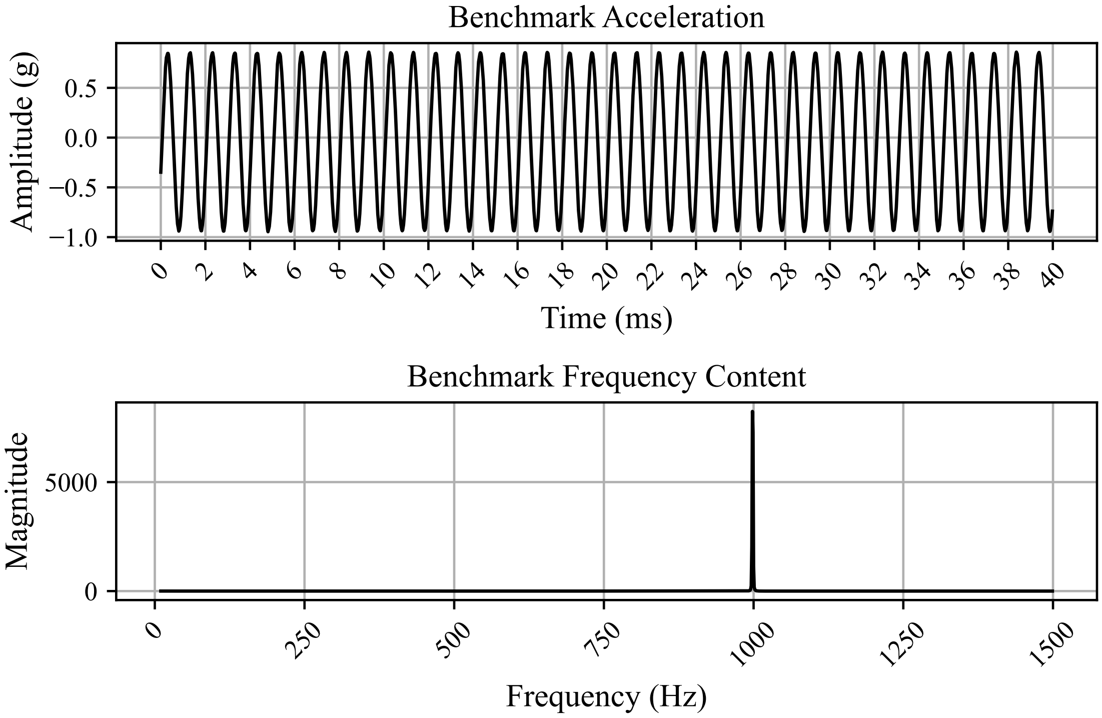
Figure 5.4 - Aliasing Testing: Z-Axis

In fact, the aliasing effect was more pronounced in a test that was conducted at 9.6 kHz, in which a folded peak at 3.35 kHz was clearly observable. This is a sign that the KX134-1211 does not filter its measured signal before digitization. It should be noted that tests conducted with 25.6 kHz sampling rates (not included in this work) do not suffer the same aliasing effects as the ones conducted at 12.8 kHz. This is most likely attributable to the fact that the KX134-1211 itself is not physically sensitive to vibrations beyond its maximum ODR. The microphone (Nyquist frequency of 10.9 kHz, filter cut-off of 9.1 kHz) was tested at 12.6 kHz and showed no aliasing, as expected. As such, it is reasonable to assume that once the node is optimised to allow for the maximum sampling rate to be used, the risk of aliasing will be mitigated.

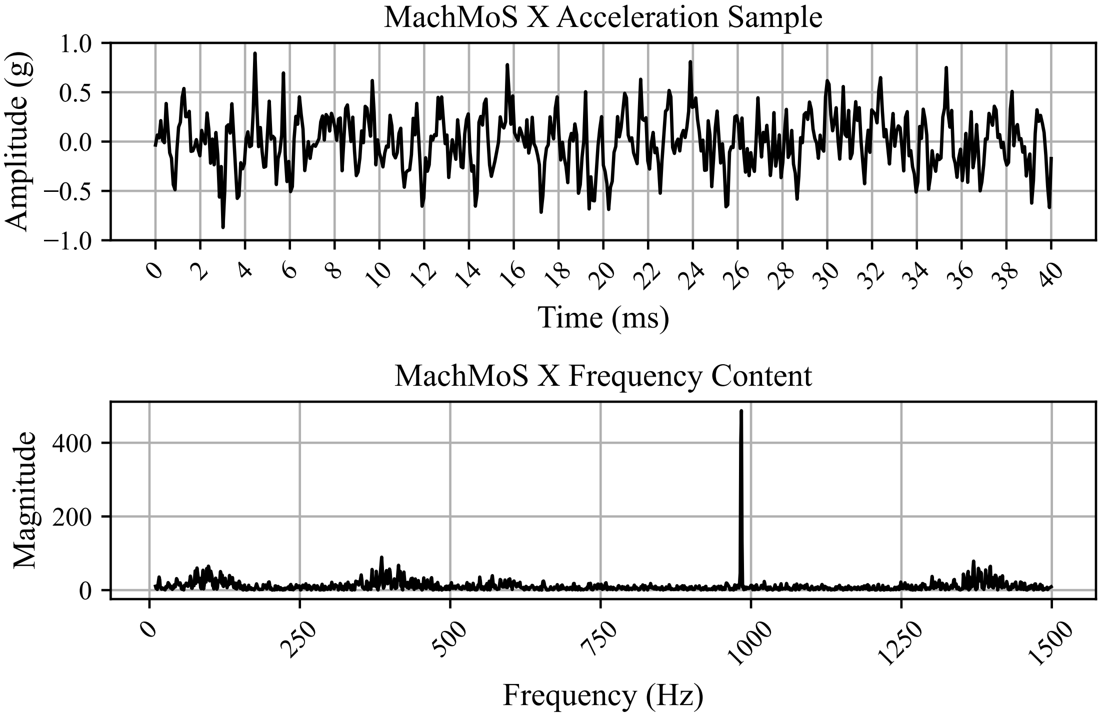
The discrepancy between the folded peak frequency bin and the expected folded peak is approximately 1.5 %, as observed in the tests that follow (subsection 5.2.3). The results of the microphone test are shown in Appendix E.

5.2.3 Vibro-Acoustic Frequency Range and Accuracy

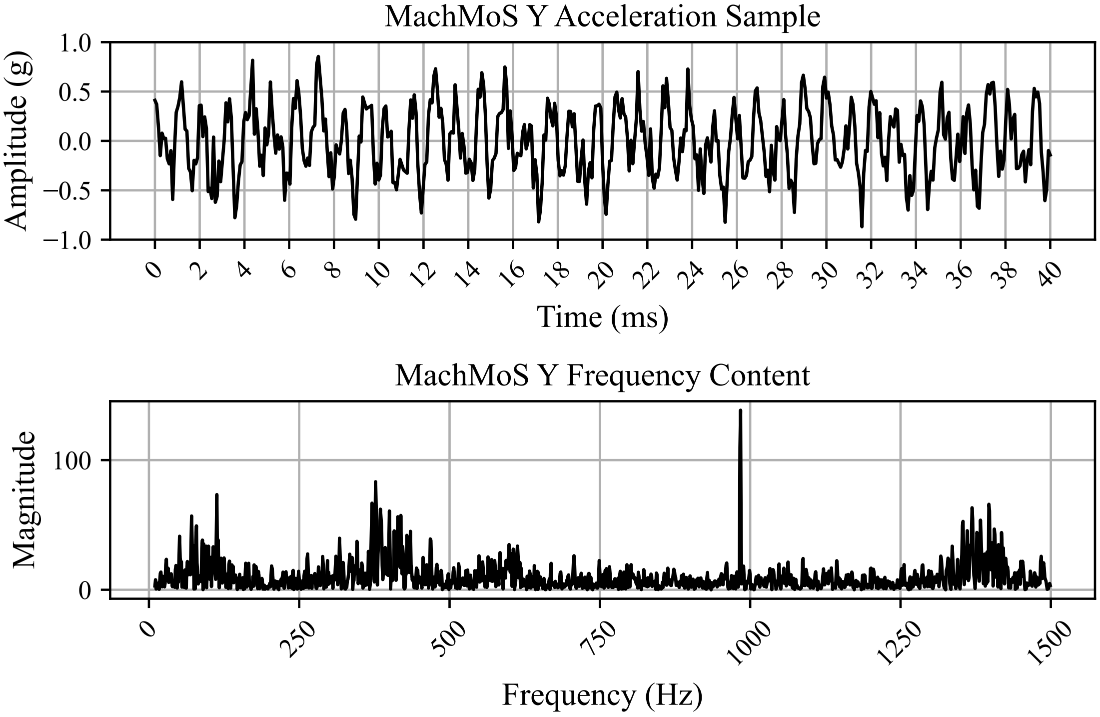
As computed in subsection 5.2.2, the maximum frequency that can be captured by the node ($f_{\max}$) is 6.4 kHz. As such, these tests were conducted with vibrator/exciter frequencies of 20 Hz, 50 Hz, 100 Hz, 500 Hz, 1 kHz, 2 kHz, 3 kHz and 6 kHz. Results for the test conducted at 1 kHz are provided in Figure 5.5 and Figure 5.6. In the figures, the raw time-series data and corresponding signal frequency content are shown. Results for the tests conducted at the other frequencies are provided in Appendix F.



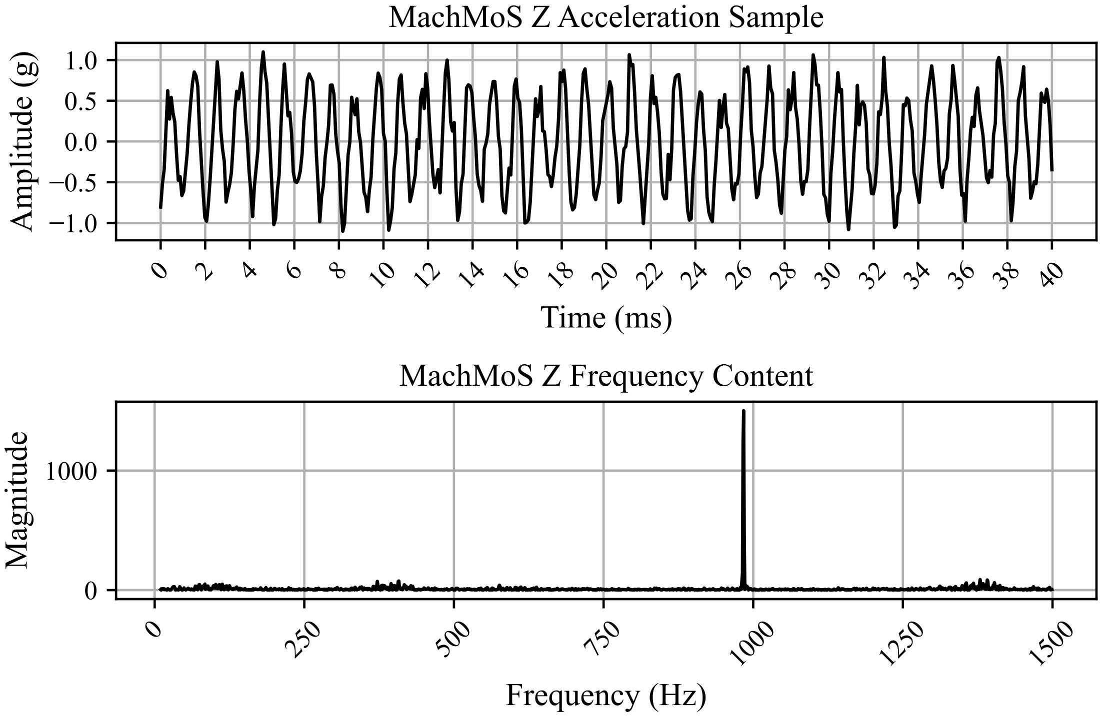
(a) Benchmark Accelerometer (along X, normalized)



(b) MachMoS Accelerometer X Axis (normalized)



(c) MachMoS Accelerometer Y Axis (normalized)

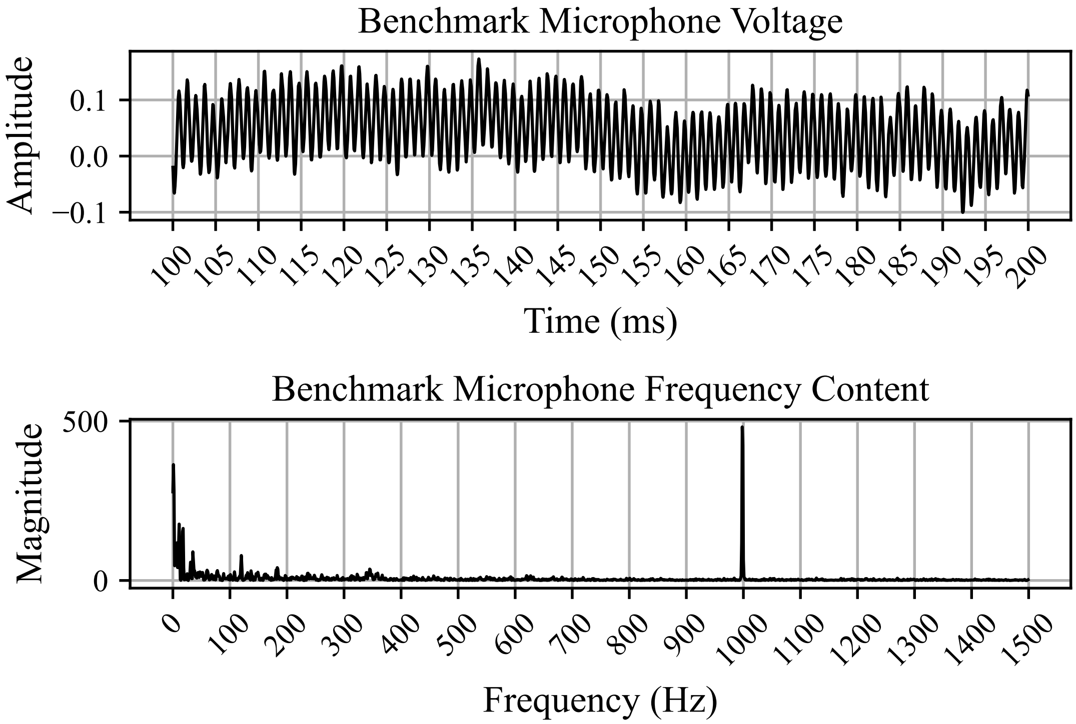


(d) MachMoS Accelerometer Z Axis (normalized)

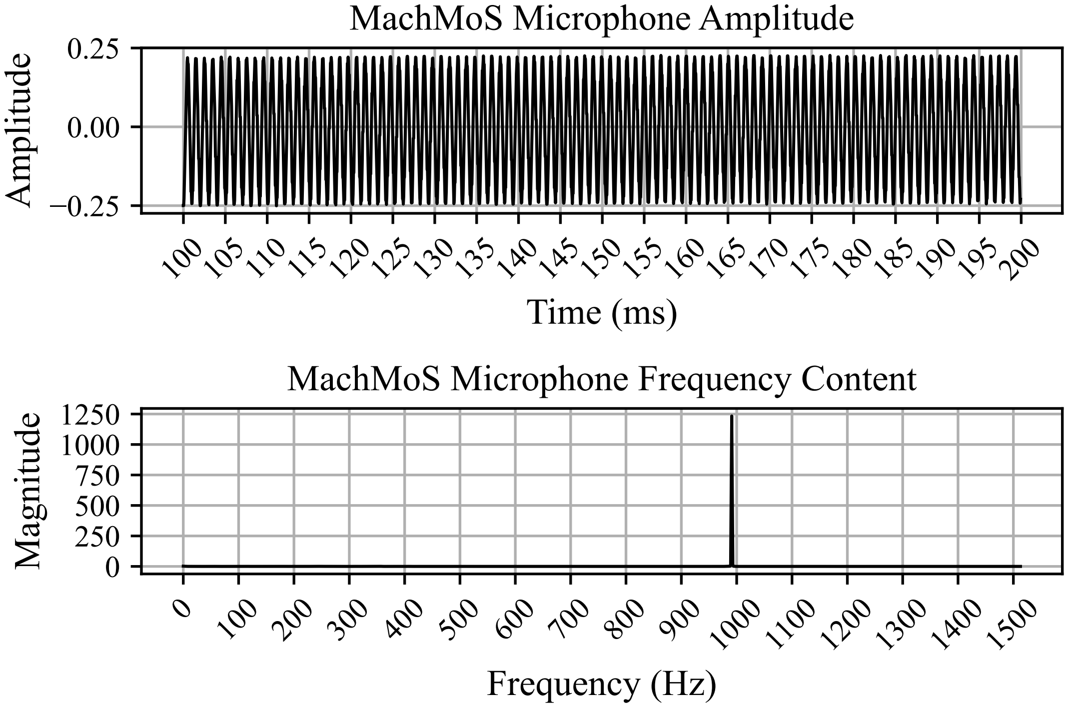
Figure 5.5 - Accelerometer Results. All data normalized and trimmed.

It can be seen in Figure 5.5(a) through (d) that the benchmark accelerometer data is much smoother than that of the accelerometer used in the MachMoS node (KX134-1211). This is to be expected, considering the differences in noise density of the benchmark accelerometer and the KX134-1211 (i.e. 7 μg/√Hz at 10 Hz sampling rate vs. 300 μg/√Hz at 50 Hz sampling rate).

The RMS of the noise for the benchmark and MachMoS signals can be obtained by multiplying the noise density by the square-root of the system $f_{\max}$ (assuming the noise density is constant over the bandwidth). The expected pk-pk value can be obtained by multiplying the RMS value by 6. This includes noise up to 3 standard deviations away from the mean (99.7 % of noise). At the sampling frequency used for the KX134-1211 accelerometer (12.8 kHz), the RMS and pk-pk noise can be determined to be 0.024 g and 0.144 g, respectively. This amount of noise is large compared to the measured signal and explains why the MachMoS measurement is visibly much noisier. Using the same analysis, the RMS and pk-pk noise of the benchmark (sampled at 51.2 kHz) are 0.0011 g and 0.0066 g, respectively.



(a) Benchmark Microphone



(b) MachMoS Microphone

Figure 5.6 - Microphone Results. All data normalized and trimmed.

The microphone data is much cleaner than that of the accelerometer (Figure 5.6(b)). It is comparable in quality to that of the benchmark microphone, though the latter shows significant baseline wander (Figure 5.6(a)), which is likely due to an insufficient signal conditioner settling time having been provided when the tests were conducted.

5.2.3.1 Frequency Content at a 1 kHz Input Frequency

Despite the noise in the MachMoS node's accelerometer signal, the frequency content of the output is conclusive as to the dominant frequency in the measured signal. In this test, each axis of the MachMoS device has a discernable peak at, or near 1 kHz. In this specific capture, the benchmark accelerometer measured a peak at 998.4 kHz, while the MachMoS node's three axes measured a peak at 983.70 ± 0.02 Hz (using quadratic interpolation to obtain the true peak between frequency bins). This represents a -1.5 % discrepancy from the benchmark. The microphone data's frequency peak is lower as well, but only by 0.7 % (991 Hz vs. 998 Hz).

Tests conducted at 6.3 kHz did not lead to a large enough peak in the frequency domain to draw conclusions. This is expected (and desired) behaviour, in which signals that are close to the Nyquist rate (6.4 kHz) are attenuated by a LP filter in the accelerometer.

As noted earlier, the discrepancy in the microphone's frequency peak is 0.7 % at 1 kHz. One possible source of this discrepancy is an inaccuracy in the I^{2}S clock frequency generated by the ESP32-C3's fractional scaler, as described in section 4.1.7. However, if the fractional scaler analysis presented in Appendix B.1 is performed using a decimation ratio of 384 (the parameter used during testing), the resulting scaling error is 0.62 ppm. This error is negligeable over a one-second capture interval and cannot account for the observed frequency deviation. This discrepancy is further discussed in subsection 5.2.3.3.

5.2.3.2 Resonance and Frequency Content with a Variable Excitation Voltage (3 kHz)

The MachMoS device was tested with vibration/exciter voltage inputs of 4 V and 8 V at 3 kHz. The results are shown in Figure 5.7. In this figure, a portion of the time series data from the benchmark and MachMoS node are shown. The frequency contents are also provided, showing the difference between both sensors.

The SNR of the accelerometer signals were computed using the frequency content shown in Figure 5.7. The SNR for the benchmark was found to be 33 dB at 4 V and 25 dB at 8 V. The SNR for the MachMoS node X-axis was 1.6 dB at 4 V and 6.0 dB at 8 V.

It can be assumed that the 2x increase in the vibrator/exciter input from 4 V to 8 V increases the vibrator/exciter signal's spectral power by 4x (6.0 dB). This is expected to also increase the signal power of the accelerometer measurements by 6.0 dB. This effect was not observed in the benchmark data but is consistent with the increase in SNR of the MachMoS node accelerometer at 4.4 dB.

As mentioned in the previous subsection, the spectral noise density of the benchmark and MachMoS node accelerometers are 7 µg/√Hz and 300 µg/√Hz, respectively. From the noise density, the pk-pk baseline noise, without excitation, was previously determined to be 6.6 mg and 144 mg for the benchmark and MachMoS node accelerometers, respectively.

In the 4 V, 3 kHz test, the pk-pk acceleration measured by the benchmark is approximately 1 g. This pk-pk acceleration also represents the acceleration that the MachMoS accelerometer experienced. The 144 mg pk-pk baseline noise of the MachMoS accelerometer is equivalent to 14.4 % of the pk-pk acceleration experienced by the sensor and explains why the time series data of the MachMoS node is very noisy. In the 8 V test, the pk-pk acceleration approximately doubles as the voltage is increased, and the time series data is much cleaner.

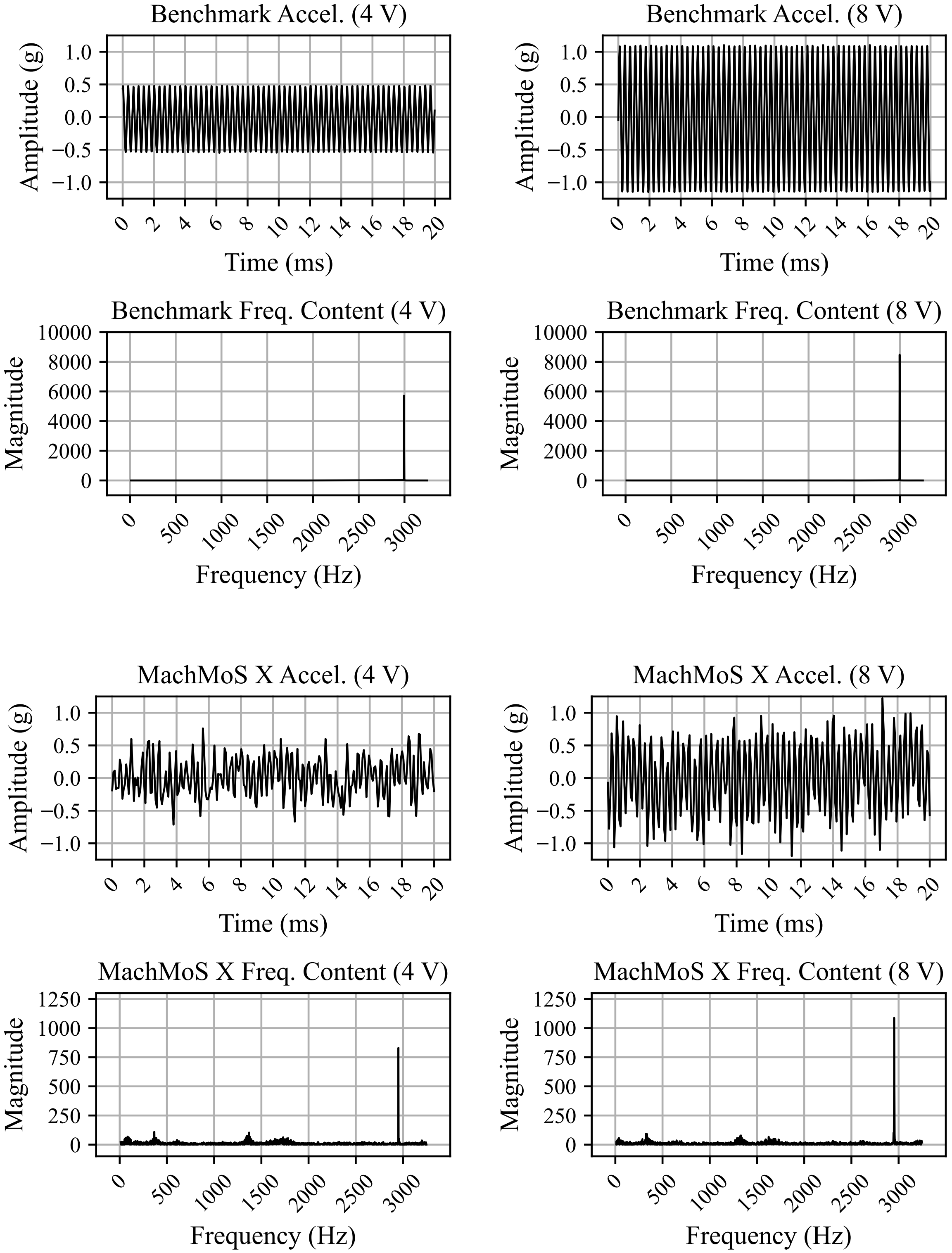


Figure 5.7 - Accelerometer Readings for 4 V and 8 V Exciter Inputs at 3 kHz

As seen in Figure 5.7, small peaks are present in the noise floor of the frequency content of the KX134-1211 accelerometer data. These peaks are most pronounced around 350 Hz, 1.3 kHz, and 1.7 kHz. The radial frequency modes of the enclosure were previously modeled in subsection 4.7.2 by using a SolidWorks vibration simulation. The results for the radial components of the resonance were selected from Figure 4.21 and presented in Figure 5.8 for clarity. In this case, modeling predicted two major groups of peaks, centered at 325 Hz and 1.8 kHz. These are present in the experimental results, shown in Figure 5.7. However, the group of peaks in Figure 5.7 centered on 1.3 kHz were not predicted by the model and could have been introduced by any other component of the test setup. This validates the use of SolidWorks vibration simulations to predict the resonant modes of a vibration sensor.

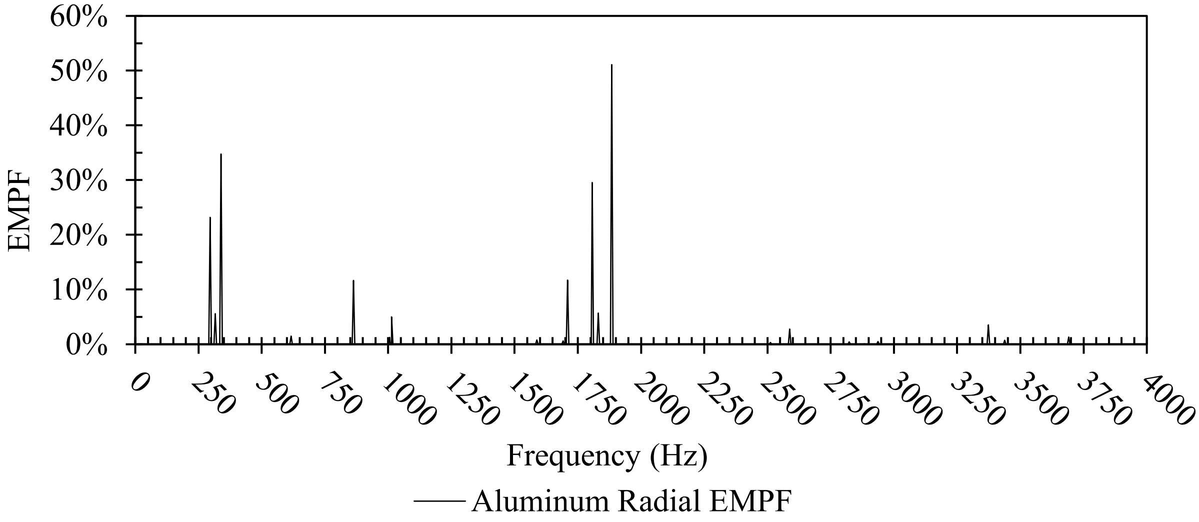


Figure 5.8 - Resonant Modes and EMPF for the Aluminum Enclosure Base

Despite the noise introduced by resonance, the SNR of the signal remains high. It is important to point out that this noise is absent in most of the microphone captures. This can be seen in the plots shown in Appendix E.

It is important to compare these results with the vibration amplitudes found in industry. From subsection 2.1.1.6 of the literature review, it is possible to compute the vibration limits for a frequency of 3 kHz, assuming a sinusoidal vibration signal.

First, the velocity RMS value (m/s) must be converted to a pk-pk value by multiplying by $2\sqrt{2}$. Then, the pk-pk acceleration (m^2/s) is obtained by multiplying the pk-pk velocity by $2\pi f$ where f

is the vibration frequency. Finally, the acceleration is divided by 9.81 m/s^2/g to obtain the pk-pk acceleration limit for the given frequency.

At a frequency of 3 kHz, the acceptable acceleration limits for pumps > 15 kW on a rigid foundation are less than 7.61 g pk-pk for new and reconditioned machines, 12.5 to 15.2 g pk-pk for machines that can operate indefinitely, and 19.0 to 24.5 g pk-pk for machines that do not operate for a long time. In addition to this, if the vibration is above 38.6 g pk-pk, damage is expected.

Likewise, at a frequency of 3 kHz, the acceptable acceleration limits for electric motors with a power rating < 15 kW are as follows: less than 3.9 g pk-pk for good, 6.1 to 9.8 g pk-pk for satisfactory, 15.2 to 24.5 g pk-pk for unsatisfactory, and over 38.6 g pk-pk for unacceptable operation.

The accelerations tested in this work were approximately 2 g pk-pk and are within the limits for new and reconditioned pumps > 15 kW as well as those for the good operation of electric motors having a power rating < 15 kW. The full-scale acceleration range for the KX134 -1211 accelerometer used in the MachMoS node is ±64 g and is suitable to monitor all the levels of vibration severity listed above.

5.2.3.3 Summary of Frequency Content Test Results

The discrepancy on the frequency peak for either sensor is consistent across all tested frequencies. The results and the discrepancy on the peak are provided in Table 5.1 and Table 5.2. To ensure that the test setup was not the cause of the discrepancy, additional tests were performed with a node attached directly above the vibrator/exciter in the absence of the benchmark. The same discrepancy of -1.5 % at 1 kHz was obtained for the accelerometer measurements. The source of the discrepancy remains unknown and would require further optimisation of the node, which remains beyond the scope of this thesis.

Table 5.1 - Tested Frequencies and Associated Measurement Errors (Accelerometer)

Input Freq. (set using oscilloscope)	Input Voltage (V)	Measured Freq. (Benchmark, Hz)	Error Benchmark vs. Input Freq. (%)	Measured Freq. (Node X-axis, Hz)	Error Node vs. Input Freq. (%)	Error Node vs. Benchmark (%)
20	1	19.98	-0.10%	19.69	-1.55%	-1.47%
50	4	49.94	-0.12%	49.17	-1.66%	-1.57%
100	4	99.88	-0.12%	98.31	-1.69%	-1.60%
500	4	499.2	-0.16%	492.0	-1.60%	-1.46%
1000	4	998.4	-0.16%	983.7	-1.63%	-1.49%
2000	4	1996.9	-0.15%	1968.9	-1.56%	-1.42%
3000	4	2995.2	-0.16%	2951.8	-1.61%	-1.47%
3000	8	2995.2	-0.16%	2953.3	-1.56%	-1.42%
6000	8	5990.4	-0.16%	5905.6	-1.57%	-1.44%

Table 5.2 - Tested Frequencies and Associated Measurement Errors (Microphone)

Input Freq. (set using oscilloscope)	Input Voltage (V)	Measured Freq. (Benchmark, Hz)	Error Benchmark vs. Input Freq. (%)	Measured Freq. (Node, Hz)	Error Node vs. Input Freq. (%)	Error Node vs. Benchmark (%)
20	1	20.14	0.71%	19.88	-0.63%	-1.32%
50	4	49.93	-0.13%	49.53	-0.94%	-0.81%
100	4	99.88	-0.12%	99.07	-0.93%	-0.81%
500	4	499.2	-0.17%	495.6	-0.89%	-0.73%
1000	4	998.4	-0.16%	991.1	-0.89%	-0.73%
2000	4	1996.9	-0.16%	1982.2	-0.89%	-0.73%
3000	4	2995.2	-0.16%	2973.3	-0.89%	-0.73%
3000	8	2995.2	-0.16%	2973.3	-0.89%	-0.73%
6000	8	5990.4	-0.16%	5946.6	-0.89%	-0.73%

5.2.3.4 Accelerometer Amplitude

In these tests, the benchmark accelerometer was aligned with the X axis of the MachMoS accelerometer. From Figure 5.5(a), it was determined that the benchmark accelerometer measured a pk-pk amplitude of 1.8 g in the 1 kHz test. As such, the expected pk-pk accelerometer measurement for the node's accelerometer X-axis is also 1.8 g. The Y-axis of the accelerometer should measure the same acceleration, as both axes are oriented at the same angle from the vertical (shown in Figure 3.3). Vector math is used to determine that the MachMoS node's Z-axis underwent 2.5 g of acceleration pk-pk during the 1 kHz test.

The measurement data shows that the Z-axis of the node measured a 2.6 g pk-pk value in the test (0.1 g larger than expected). The X and Y axes, 2.0 g (0.2 g larger than expected) and 2.1 g (0.3 g larger than expected), respectively. The error between the benchmark and node can be attributed to the pk-pk noise in the accelerometer data as well as baseline drift in the MachMoS data.

To determine if sensor noise is too large for the application, the findings of subsection 5.2.3.2 can be used to apply the same analysis for a rotational speed of 3600 RPM (60 Hz) commonly found in industry. At a frequency of 60 Hz, the acceleration limits for electric motors with a power rating < 15 kW are as follows: less than 77 mg pk-pk for *good*, 122 to 186 mg pk-pk for *satisfactory*, 304 to 489 mg pk-pk for *unsatisfactory*, and over 772 mg pk-pk for *unacceptable* operation. The pk-pk noise for the KX134-1211 used in the MachMoS node was found to be 144 mg at a sampling rate of 12.6 kHz. The accelerometer is, therefore, found to be suitable for detecting the *satisfactory* operation of small machinery (< 15 kW) as well as its transition from *satisfactory* to *unsatisfactory*. However, the SNR must also be considered.

Acceptable pk-pk sensor noise for an accelerometer can be proposed at one decade lower than the 77 mg pk-pk vibration amplitude for a *good* operating condition. Assuming a sampling rate of 25.6 kHz and a corresponding sensor bandwidth of 12.8 kHz, the appropriate noise density for an accelerometer would be 68 µg/√Hz, which is less than 300 µg/√Hz. This indicates that if *good* machine operation is to be distinguished from *satisfactory* operation, a more sensitive accelerometer should be used in this application. This is further discussed in section 6.2.

5.2.4 Accelerometer and Microphone Test Conclusions

In conclusion, it is possible to qualify acceptable performance of the MachMoS node for signals at or under 6 kHz. If the accelerometer ODR is set to 25.6 kHz, however, the aliasing risk is mitigated, but this is not currently supported. While the SNR and measured amplitudes of the accelerometer captures are not as high as that of the benchmark, it should be reiterated that the peaks are well defined in the frequency domain, showing that the data is appropriate for condition monitoring. Finally, the ICS-43434 microphone has shown complete immunity to aliasing in this test.

5.3 Ingress Protection

The main PCB of the node was completely removed for the ingress protection test to ensure the device's electronics were not damaged if water infiltrated the enclosure. A sheet of absorbent towel was used to detect water ingress during the test. As they are essential to ensuring the seal of the enclosure, the O-rings and USB jack were assembled in the enclosure. A ballast was manufactured from aluminum stock and tapped with a 1/8-NPT thread to ensure the node would submerge in the appropriate orientation during the test. Using this ballast, the node was first positioned upright, then the test repeated with the node upside-down. The microphone orifice in the node enclosure (see subsection 4.3.5.2) was sealed using vacuum sealant tape (tacky tape) of the type used in the fabrication of composites through vacuum-assisted resin transfer moulding. During normal operation, this orifice is protected using an acoustic vent that meets the requirement of the IP standard (as per manufacturer testing already conducted).

To confirm the effectiveness of the vacuum tape, and to detect any voids in the 3D printed shell, the MachMoS node was initially tested in a stainless-steel vat having a depth of approximately 0.2 m. There was no leakage into the enclosure during this test.

The device was then moved to a 1.3 m-deep water-filled concrete vat. The depth being more than 1 m, a successful test would therefore qualify the device for IPX8.

Due to shrinkage during the 3D printing process, the enclosure did not seal fully, and the first test was unsuccessful. Shrinkage tests conducted on the printer used to manufacture the enclosure (*Bambu Labs* X1 Carbon) found that the specific ASA material used (EL3D® ASA Filament [199]) had a 1 % shrinkage coefficient.

The geometry of the enclosure lid was, therefore, modified at the interface to accommodate for the shrinkage. Other issues with printing, such as warping and voids, were also mitigated by modifying the geometry of the enclosure to avoid thin branching of the geometry, reducing the print speed, and increasing the build chamber temperature.

The final cover was printed successfully using a 0.4 mm nozzle, 0.16 mm layer height, 4 wall loops (to ensure a full print at the O-ring interface), and a cubic infill of 15 % density. The enclosure was retested as described in subsection 3.2.3. There was no sign of ingress during this test. It can be concluded that the MachMoS enclosure is IPX8 rated, per the requirements set out in IEC 60529. This enclosure was used in all experiments reported in this thesis.

5.4 Charge Circuitry

Testing of the charge circuitry was successfully conducted as outlined in subsection 3.2.4. Isolation of the two battery packs in the device during charging was successful. No errors were triggered during charging, and the battery temperature did not rise significantly during the charge cycle. The battery pack reached a full charge within the pre-calculated 4-hour charge time.

5.5 Energy Usage

The results for the energy usage test are provided in Table 5.3. These results are for the per-cycle energy usage of the node and were conducted expecting 1 capture cycle per hour and 1 s of data captured per cycle. The measurement method is described in subsection 3.2.5. During the test, it was noticed that the node had a considerably larger current draw when connecting to the gateway than when it was transmitting measurement data. In the results, the *send* task is, therefore, split in two: the *connect* and *send* tasks. The respective current draws and time required for those tasks are listed in Table 5.3.

Table 5.3 - Actual MachMoS Node Per-Cycle Energy Usage

Task	Voltage (V)	Current Draw (mA)	Time (s)	Energy (µWh/cycle)
Capture	2.653	54.4	1	40
Connect / Send	2.653	85.0 / 64.1	3 / 11	708 (total)
Sleep	2.653	0.057	3586	151
			Total:	899

* The time required for a transfer from one corner to the other of a busy laboratory with multiple wooden and metal obstacles. Tests at shorter distances (3 m, with concrete and wooden obstacles) show a *send* time reduction to 7 s (519 µWh).

The total energy usage per cycle listed in Table 5.3 is lower than 1049 μWh/cycle, the energy usage calculated in subsection 4.2.3.1. As such, the MachMoS node is expected to have more than 1 year of life between charges and meets specification S.2.

5.6 External Sensor

An external analog sensor (DRV425 fluxgate sensor) was tested with the MachMoS node device. The test showed that a magnetic field sensed by the sensor could be measured by the MachMoS node. The signal was noiseless, despite being transferred over a 2-meter-long cable.

However, the analog signal could not be read with some USB cable models. The signal is transferred using the TX1/RX1 and TX2/RX2 lines of the USB-C cable. It is suspected that some of the ICs used in modern USB-C cables, which serve the purpose of managing power delivery functionality, disable the TX and RX lines of the cable when it detects that a legacy USB 2.0 device is connected. It is also possible that the USB-C cable's ICs require power to allow communication through those lines. The MachMoS device does not supply 5 V to the USB cable as it is not equipped with the required boost regulator.

5.7 Wireless Range

The wireless range test was conducted as described in subsection 3.2.6. The MachMoS node successfully connected to a gateway device and transferred data at a LoS distance of at most 70 m. At this distance, the time required to transfer 1 s of data averaged 12 s. While this range is less than the desired 100 m range specified under specification S.8, there are options within the ESP32-C3 code that allow for a higher transmission power to be used (up to 9 dBm).

Using reference [200], it can be calculated that the observed range is within the expected theoretical range, showing proper integration of the ESP32-WROOM device within the node. Using a 3 dBm transmit power (the ESP32-C3 default) and 3 dBi antenna gain (ESP-WROOM PCB antenna per [201]) with a 0 dBm receiver antenna gain and when using a receiver sensitivity of -91 dBm, which is within the typical range for 1M BLE PHY, the expected signal range is 68 to 96 meters. Using the same model for estimating transmission range, the expected range should increase to a minimum of 103 m when a 9 dBm transmitter power is used. While the increased transmission power allows the node to meet specification S.8, the increased transmission power would negatively affect the node life between charges.

5.8 Electro Magnetic Interference and Compatibility

EMI/EMC testing is beyond the scope of this work as the facilities required for these tests were not available. These interference and compatibility tests will need to be performed in the future if the MachMoS node is to be implemented in practice.

Chapter 6 Conclusions and Future Work

6.1 Conclusions

This thesis detailed the design and validation of a novel, low-cost vibro-acoustic sensor for use in machine condition monitoring.

The results of this work demonstrated that a low-cost sensor node suitable for vibration monitoring applications can be produced to generate open-access vibro-acoustic and temperature data. The device meets specifications developed using a benchmarking process that included various existing commercial solutions.

The main conclusions of this work are listed as follows:

- The accuracy of the ratio-metric RTD measurement circuit was shown to be within ±0.25°C. The device was found to operate at temperatures as low as -15°C.
- The maximum data rate of the device using the ESP32-C3 MCU with flash memory was shown to be limited to 12.6 kHz of tri-axial 16-bit accelerometer data and 21.8 kHz of 24-bit microphone data.
- The anti-aliasing filter of the ICS-43434 microphone was effective, but the filter of the KX134-1211 accelerometer was not. The KX134-1211 is not sensitive to vibrations beyond its maximum ODR (25.6 kHz) and does not require an anti-aliasing filter when this ODR is used.
- The device exhibits reliable vibro-acoustic and temperature sensing capabilities. Testing confirmed the device's ability to capture vibration signals up to 6 kHz and acoustic signals beyond 7 kHz, with a discrepancy of approximately -1.5 % on frequency.
- Vibration simulations using SolidWorks provided good predictions of the main resonant modes of the enclosure.
- An enclosure meeting the ingress protection standard can be produced using fused filament fabrication when material shrinkage and print geometry are considered.
- In addition, energy consumption testing showed that a sensor node that uses rechargeable NiMH batteries can be designed to offer at least 1 year between charging cycles for

heavy use. The charge circuitry for such cells can be integrated into the sensor and can provide a full charge within 4 hours using a USB cable.

- Analog signals can be transferred through the TX/RX pins of a full-featured USB-C device, though this practice is not recommended due to variability in the circuitry within USB-C cables.
- The node can successfully connect and transfer data at up to 70 m line of sight using the BLE protocol and the wireless antenna contained in the ESP32-WROOM device.
- The node can be manufactured using the facilities available in research institutions.

6.2 Future Work

While the performance of the node remains robust, it is recommended that future work focus on optimizing the embedded code running on the node to allow for greater sampling frequencies. It is also recommended that solutions for improving clock accuracy should be investigated. Future testing is required to enhance MachMoS node capabilities. Recommended tests are listed as follows:

- IP6X testing.
- Complete operational temperature range testing from -40°C to 85°C.
- EMC/EMI testing.
- Testing in an industrial setting.

Future designs and prototypes of the MachMoS device would benefit from using an accelerometer with a noise density of at most 68 µg/√Hz. To further reduce cost and device complexity, an LDO is recommended to regulate the USB voltage used to charge the device, as the efficiency of this regulator is not as important as that of the boost regulator used to regulate the battery voltage. The 24-pin full-featured *Amphenol* MUSBR USB-C jack could be replaced with a low-cost, waterproof, compact alternative.

References

[1] D. Acemoglu and P. Restrepo, “Automation and New Tasks: How Technology Displaces and Reinstates Labor,” *J. Econ. Perspect.*, vol. 33, no. 2, pp. 3–30, May 2019, doi: 10.1257/jep.33.2.3.

[2] “Jobs of the future: Jobs lost, jobs gained | McKinsey.” Accessed: Sep. 27, 2025. [Online]. Available: https://www.mckinsey.com/featured-insights/future-of-work/jobs-lost-jobs-gained-what-the-future-of-work-will-mean-for-jobs-skills-and-wages

[3] “Machine Condition Monitoring Market | Industry Report, 2032.” Accessed: Sep. 27, 2025. [Online]. Available: https://www.fortunebusinessinsights.com/machine-condition-monitoring-market-112654

[4] “Machine Condition Monitoring Market Size Report, 2030.” Accessed: Sep. 27, 2025. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/machine-condition-monitoring-market-report

[5] “Machine Condition Monitoring Market Size to Reach USD 6.98 billion by 2034.” Accessed: Sep. 27, 2025. [Online]. Available: https://www.precedenceresearch.com/machine-condition-monitoring-market

[6] N. Tandon and A. Choudhury, “A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings,” *Tribol. Int.*, vol. 32, no. 8, pp. 469–480, Aug. 1999, doi: 10.1016/S0301-679X(99)00077-8.

[7] W. Zhou, T. G. Habetler, and R. G. Harley, “Bearing Condition Monitoring Methods for Electric Machines: A General Review,” in *2007 IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives*, Sep. 2007, pp. 3–6. doi: 10.1109/DEMPED.2007.4393062.

[8] R. R. Schoen, “On-line current-based condition monitoring of three-phase induction machines,” Ph.D., Georgia Institute of Technology, United States -- Georgia, 1994. Accessed: Oct. 22, 2025. [Online]. Available: https://www.proquest.com/docview/304107299/abstract/E45528427E7A4A5DPQ/1

[9] T. Goktas, M. Zafarani, K. W. Lee, B. Akin, and T. Sculley, “Comprehensive Analysis of Magnet Defect Fault Monitoring Through Leakage Flux,” *IEEE Trans. Magn.*, vol. 53, no. 4, pp. 1–10, Apr. 2017, doi: 10.1109/TMAG.2016.2617318.

[10] *ISO 17359*, Jan. 2018. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/71194.html

[11] *ISO 10817-1*, Dec. 1998. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/1175.html

[12] *ISO 22096*, Jul. 2007. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/40686.html

[13] *ISO 13379-1*, Oct. 2025. Accessed: Oct. 30, 2025. [Online]. Available: https://www.iso.org/standard/88027.html

[14] *ISO 13379-2*, Apr. 2015. Accessed: Oct. 30, 2025. [Online]. Available: https://www.iso.org/standard/52258.html

[15] *ISO 13381-1*, Sep. 2025. Accessed: Oct. 29, 2025. [Online]. Available: https://www.iso.org/standard/88029.html

[16] *ISO 13373-1*, Feb. 2002. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/21831.html

[17] *ISO 13373-2*, Jan. 2016. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/68128.html

[18] *ISO 13373-3*, Sep. 2015. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/40840.html

[19] *ISO 13373-4*, Nov. 2021. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/62201.html

[20] *ISO 13373-5*, Jun. 2020. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/62202.html
[21] *ISO 13373-7*, Aug. 2017. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/62204.html
[22] *ISO/AWI 13373-8*. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/78145.html
[23] *ISO 13373-9*, Oct. 2017. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/62190.html
[24] *ISO 13373-10*, Jun. 2024. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/82072.html
[25] *ISO/PWI 13373-11*, Nov. 2022. Accessed: Oct. 28, 2025. [Online]. Available: https://genorma.com/en/standards/iso-pwi-13373-11
[26] *ISO 20816-1*, Nov. 2016. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/63180.html
[27] *ISO 20816-2*, Jul. 2017. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/70047.html
[28] *ISO 20816-3*, Oct. 2022. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/78311.html
[29] *ISO 20816-4*, Jun. 2018. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/74450.html
[30] *ISO 20816-5*, Jul. 2018. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/67919.html
[31] *ISO 20816-8*, Aug. 2018. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/75440.html
[32] *ISO 20816-9*, Jun. 2020. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/75670.html
[33] *ISO 20816-21*, May 2025. Accessed: Oct. 28, 2025. [Online]. Available: https://www.iso.org/standard/84280.html
[34] M. Baltazar, B. Ramírez, and G. Kemper, "A Low-Complexity Algorithm for Diagnosis of Three-Phase Induction Motors," in *Proceedings of the 6th Brazilian Technology Symposium (BTSym'20)*, vol. 233, Y. Iano, O. Saotome, G. Kemper, A. C. Mendes De Seixas, and G. Gomes De Oliveira, Eds., in Smart Innovation, Systems and Technologies, vol. 233. , Cham: Springer International Publishing, 2021, pp. 929–948. doi: 10.1007/978-3-030-75680-2_102.
[35] N. Jamil, M. F. Hassan, S. K. Lim, and A. R. Yusoff, "Predictive maintenance for rotating machinery by using vibration analysis," *J. Mech. Eng. Sci.*, vol. 15, no. 3, pp. 8289–8299, Sep. 2021, doi: 10.15282/jmes.15.3.2021.07.0651.
[36] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review," *IEEE Access*, vol. 8, pp. 29857–29881, 2020, doi: 10.1109/ACCESS.2020.2972859.
[37] J. Antoni *et al.*, "On the design of Optimal Health Indicators for early fault detection and their statistical thresholds," *Mech. Syst. Signal Process.*, vol. 218, p. 111518, Sep. 2024, doi: 10.1016/j.ymssp.2024.111518.
[38] Y. Lv, W. Zhao, Z. Zhao, W. Li, and K. K. H. Ng, "Vibration signal-based early fault prognosis: Status quo and applications," *Adv. Eng. Inform.*, vol. 52, p. 101609, Apr. 2022, doi: 10.1016/j.aei.2022.101609.
[39] A. Moshrefzadeh, "Condition monitoring and intelligent diagnosis of rolling element bearings under constant/variable load and speed conditions," *Mech. Syst. Signal Process.*, vol. 149, p. 107153, Feb. 2021, doi: 10.1016/j.ymssp.2020.107153.
[40] N. Belrose, Q. Pope, L. Quirke, A. Mallen, and X. Fern, "Neural Networks Learn Statistics of Increasing Complexity," arXiv.org. Accessed: Apr. 21, 2026. [Online]. Available: https://arxiv.org/abs/2402.04362v3

[41] O. Ure, S. Demir, and Z. Dogan, “Learning Beyond the Gaussian Data: Learning Dynamics of Neural Networks on an Expressive and Cumulant-Controllable Data Model,” arXiv.org. Accessed: Apr. 21, 2026. [Online]. Available: https://arxiv.org/abs/2602.02153v1

[42] “Bearing Data Center | Case School of Engineering | Case Western Reserve University,” Case School of Engineering. Accessed: May 09, 2022. [Online]. Available: https://engineering.case.edu/bearingdatacenter

[43] J. Hendriks, P. Dumond, and D. A. Knox, “Towards better benchmarking using the CWRU bearing fault dataset,” *Mech. Syst. Signal Process.*, vol. 169, p. 108732, Apr. 2022, doi: 10.1016/j.ymssp.2021.108732.

[44] M. Sehri and P. Dumond, “Optimizing Rolling Element Bearing Data Collection and Algorithm Hyperparameters for Machine Learning,” 2024, *SSRN*. doi: 10.2139/ssrn.4949181.

[45] “Bearing Data Center | Case School of Engineering.” Accessed: Oct. 21, 2025. [Online]. Available: https://engineering.case.edu/bearingdatacenter

[46] K. Alqunun *et al.*, “An efficient bearing fault detection strategy based on a hybrid machine learning technique,” *Sci. Rep.*, vol. 15, no. 1, p. 18739, May 2025, doi: 10.1038/s41598-025-02439-4.

[47] W. A. Smith and R. B. Randall, “Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study,” *Mech. Syst. Signal Process.*, vol. 64–65, pp. 100–131, Dec. 2015, doi: 10.1016/j.ymssp.2015.04.021.

[48] S. J. Kim *et al.*, “Motor-current-based electromagnetic interference de-noising method for rolling element bearing diagnosis using acoustic emission sensors,” *Measurement*, vol. 193, p. 110912, Apr. 2022, doi: 10.1016/j.measurement.2022.110912.

[49] “Konstruktions- und Antriebstechnik (KAt) - Data Sets and Download (Universität Paderborn).” Accessed: May 09, 2022. [Online]. Available: https://mb.uni-paderborn.de/en/kat/main-research/datacenter/bearing-datacenter/data-sets-and-download

[50] “Data Challenge – PHM Asia Pacific 2023.” Accessed: Oct. 22, 2025. [Online]. Available: https://phmap.jp/program-data/

[51] P. Nectoux *et al.*, “PRONOSTIA : An experimental platform for bearings accelerated degradation tests.,” in *Conference on Prognostics and Health Management.*, Denver, Colorado, United States: IEEE Catalog Number : CPF12PHM-CDR, Jun. 2012, pp. 1–8. Accessed: Oct. 22, 2025. [Online]. Available: https://hal.science/hal-00719503

[52] “DASHlink - Gearbox Fault Detection Dataset, PHM Data Challenge 2009.” Accessed: Oct. 22, 2025. [Online]. Available: https://c3.ndc.nasa.gov/dashlink/resources/997/

[53] “PHM North America 2024 Conference Data Challenge,” PHM Society Data Repository. Accessed: Oct. 22, 2025. [Online]. Available: https://data.phmsociety.org/phm2024-conference-data-challenge/

[54] “PHM North America 2025 Conference Data Challenge,” PHM Society Data Repository. Accessed: Oct. 22, 2025. [Online]. Available: https://data.phmsociety.org/phm-north-america-2025-conference-data-challenge/

[55] M. Sehri, P. Dumond, and M. Bouchard, “University of Ottawa constant load and speed rolling-element bearing vibration and acoustic fault signature datasets,” *Data Brief*, vol. 49, p. 109327, Aug. 2023, doi: 10.1016/j.dib.2023.109327.

[56] J. Zarei and J. Poshtan, “Bearing fault detection using wavelet packet transform of induction motor stator current,” *Tribol. Int.*, vol. 40, no. 5, pp. 763–769, May 2007, doi: 10.1016/j.triboint.2006.07.002.

[57] O. Gazi, “Sampling of Continuous Time Signals,” in *Understanding Digital Signal Processing*, O. Gazi, Ed., Singapore: Springer, 2018, pp. 1–70. doi: 10.1007/978-981-10-4962-0_1.

[58] “Fourier Transform and Signal Spectrum,” in *Digital Signal Processing*, John Wiley & Sons, Ltd, 2005, pp. 79–100. doi: 10.1002/0470034009.ch4.

[59] J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex Fourier series,” *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965, doi: 10.1090/S0025-5718-1965-0178586-1.

[60] J. Mais, “Spectrum Analysis: The key features of analyzing spectra.” SKF USA Inc., May 2002. Accessed: Oct. 23, 2025. [Online]. Available: https://cdn.skfmediahub.skf.com/api/public/0901d1968024acef/pdf_preview_medium/0901d1968024acef_pdf_preview_medium.pdf

[61] The Timken Company, “Bearing Dynamics and Sound.” Timken, 2017. Accessed: Nov. 04, 2025. [Online]. Available: https://www.timken.com/fr/resources/a-technical-review-of-bearing-dynamics-and-sound/

[62] J. Mais, “Spectrum Analysis.” SKF USA, Inc., May 2002. Accessed: Oct. 28, 2025. [Online]. Available: https://cdn.skfmediahub.skf.com/api/public/0901d1968024acef/pdf_preview_medium/0901d1968024acef_pdf_preview_medium.pdf

[63] J. R. Stack, T. G. Habetler, and R. G. Harley, “Fault classification and fault signature production for rolling element bearings in electric machines,” in *4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003.*, Aug. 2003, pp. 172–176. doi: 10.1109/DEMPED.2003.1234568.

[64] “Neural-network-based motor rolling bearing fault diagnosis | IEEE Journals & Magazine | IEEE Xplore.” Accessed: Oct. 22, 2025. [Online]. Available: https://ieeexplore-ieee-org.proxy.bib.uottawa.ca/document/873214

[65] D. Procter, “Frequency is Everything,” CBM CONNECT®. Accessed: Oct. 23, 2025. [Online]. Available: https://www.cbmconnect.com/frequency-is-everything/

[66] R. B. Randall and J. Antoni, “Rolling element bearing diagnostics—A tutorial,” *Mech. Syst. Signal Process.*, vol. 25, no. 2, pp. 485–520, Feb. 2011, doi: 10.1016/j.ymssp.2010.07.017.

[67] H. Zumbahlenas, *Linear circuit design handbook*. Amsterdam Boston: Elsevier/Newnes Press, 2008.

[68] Acoem India, “Vibration Analysis Basics – Time waveform acquisition,” Acoem India. Accessed: Sep. 09, 2025. [Online]. Available: https://www.acoem.com/india/blog/vibration-analysis-basics-time-waveform-acquisition/

[69] Rebecca, “Vibration Analysis Data Quality: Sampling, Resolution Range.” Accessed: Sep. 09, 2025. [Online]. Available: https://www.engineeranalysis.com/sampling-time-resolution-dynamic-range/

[70] “FFT Setup - IVS Expert Wiki.” Accessed: Sep. 09, 2025. [Online]. Available: https://www.industrialvibes.com/wiki/index.php/FFT_Setup

[71] K. V. Rangarao and R. K. Mallik, *Digital Signal Processing: A Practitioner’s Approach*, 1st ed. Wiley, 2005. doi: 10.1002/0470034009.

[72] F. Ohnhäuser, *Analog-Digital converters for industrial applications including an introduction to digital-analog converters*. Heidelberg: Springer, 2015.

[73] “Analog-to-digital converters (ADCs) product selection | TI.com.” Accessed: Jul. 10, 2025. [Online]. Available: https://www.ti.com/data-converters/adc-circuit/products.html

[74] J. Wu and L. R. Carley, “Electromechanical Delta-Sigma modulation with high-Q micromechanical accelerometers and pulse density modulated force feedback,” *IEEE Trans. Circuits Syst. Regul. Pap.*, vol. 53, no. 2, pp. 274–287, Feb. 2006, doi: 10.1109/TCSI.2005.857084.

[75] P. Malcovati and A. Baschirotto, “The Evolution of Integrated Interfaces for MEMS Microphones,” *Micromachines*, vol. 9, no. 7, p. 323, Jun. 2018, doi: 10.3390/mi9070323.

[76] C. E. Shannon, “Communication in the Presence of Noise,” *Proc. IRE*, vol. 37, no. 1, pp. 10–21, Jan. 1949, doi: 10.1109/JRPROC.1949.232969.

[77] “Rotationlf | Predictive Mainteance Software | Predictive Mainteance,” Nanoprecise. Accessed: Oct. 26, 2025. [Online]. Available: https://nanoprecise.io/rotationlf/

[78] Baker Hughes, “System 1 Asset Health Management Software.” Accessed: Oct. 26, 2025. [Online]. Available: https://www.bakerhughes.com/bently-nevada/system-1-condition-monitoring-software

[79] “Echo® Wireless Vibration Monitoring System | PCB Piezotronics.” Accessed: Oct. 27, 2025. [Online]. Available: https://www.pcb.com/industrial-sensors/accelerometers/wireless-vibration-monitoring-system/landingpages/echoplus
[80] “Treon Apex,” Treon. Accessed: Oct. 26, 2025. [Online]. Available: https://treon.fi/our-products/treon-apex/
[81] National Control Devices, LLC, “NCD Base Station,” NCD.io. Accessed: Oct. 26, 2025. [Online]. Available: https://ncd.io/start/
[82] “Low-code programming for event-driven applications : Node-RED.” Accessed: Oct. 26, 2025. [Online]. Available: https://nodered.org/
[83] “openPHM.” Accessed: Feb. 12, 2024. [Online]. Available: https://openphm.org/
[84] J. Hendriks, M. Azarm, and P. Dumond, “Structured Data Ontology for AI in Industrial Asset Condition Monitoring,” *J. Sens. Actuator Netw.*, vol. 13, no. 2, Art. no. 2, Apr. 2024, doi: 10.3390/jsan13020023.
[85] E. Than, “SBOA277A: Diode-Based Temperature Measurements (Rev. A),” Texas Instruments Application Notes. Accessed: Sep. 09, 2025. [Online]. Available: https://www.ti.com/lit/an/sboa277a/sboa277a.pdf?ts=1757451670207&ref_url=https%253A%252F%252Fwww.google.com%252F
[86] “TMP102 data sheet, product information and support | TI.com.” Accessed: Sep. 09, 2025. [Online]. Available: https://www.ti.com/product/TMP102
[87] Texas Instruments, “Temperature Sensing with Thermistors (Rev. A).” Jan. 2020. Accessed: Oct. 07, 2025. [Online]. Available: https://www.ti.com/lit/wp/slay054a/slay054a.pdf?ts=1759799669471
[88] F. P. Incropera and D. P. DeWitt, *Fundamentals of heat and mass transfer*. New York [etc.: J. Wiley & Sons, 2002.
[89] Micro-Epsilon, “Basics of non contact temperature measurement,” Micro-Epsilon. Accessed: Sep. 09, 2025. [Online]. Available: https://www.micro-epsilon.com/fileadmin/download/products/dat--infrared-basics--en.pdf
[90] “What is an IR Thermometer?” Accessed: Sep. 09, 2025. [Online]. Available: https://www.fluke.com/en-ca/learn/blog/temperature/what-is-an-ir-thermometer
[91] “Analog Front-End Design for RTD Measurements | Analog Devices.” Accessed: Feb. 11, 2024. [Online]. Available: https://www.analog.com/en/resources/analog-dialogue/articles/afe-design-considerations-rtd-ratiometric.html
[92] J. Wu, “A Basic Guide to RTD Measurements.” Texas Instruments, Jun. 2018. Accessed: Jul. 10, 2025. [Online]. Available: https://www.ti.com/lit/an/sbaa275a/sbaa275a.pdf?ts=1752126590369
[93] “4 Types Of Temperature Sensors,” Atlas Scientific. Accessed: Sep. 10, 2025. [Online]. Available: https://atlas-scientific.com/blog/types-of-temperature-sensors/
[94] “Thermocouple Types | Watlow.” Accessed: Sep. 10, 2025. [Online]. Available: https://www.watlow.com/resources-and-support/engineering-tools/reference-data/thermocouple-types
[95] “reliabilitydirectstore.com/kb_results.asp?ID=30.” Accessed: Oct. 22, 2025. [Online]. Available: https://www.reliabilitydirectstore.com/kb_results.asp?ID=30
[96] “Accelerometers vs. Velocity Sensors | Key Differences Explained.” Accessed: Oct. 22, 2025. [Online]. Available: https://www.metrixvibration.com/applications/accelerometers-versus-velocity-sensors
[97] linfeng, “A Detailed Comparison of Analog vs. Digital MEMS Microphone Design - SILICON SOURCE.” Accessed: Oct. 23, 2025. [Online]. Available: https://sistc.com/blog-analog-vs-digital-mems-microphone-design/
[98] “Micro-Electro-Mechanical Systems Microphones: A Brief Review Emphasizing Recent Advances in Audible Spectrum Applications - PMC.” Accessed: Oct. 23, 2025. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10972232/

[99] T. Kite, “Understanding PDM Digital Audio.” Audio Precision, 2012. [Online]. Available: https://users.ece.utexas.edu/~bevans/courses/rtdsp/lectures/10_Data_Conversion/AP_Understanding_PDM_Digital_Audio.pdf

[100] NXP, “I2S bus specification.” NXP Semiconductor, 2022. [Online]. Available: https://www.nxp.com/docs/en/user-manual/UM11732.pdf

[101] “How a fluxgate works,” Imperial College London. Accessed: Oct. 22, 2025. [Online]. Available: https://www.imperial.ac.uk/a-z-research/space-and-atmospheric-physics/research/areas/space-magnetometer-laboratory/space-instrumentation-research/magnetometers/fluxgate-magnetometers/how-a-fluxgate-works/

[102] A. Kobayashi, K. Nakamura, and T. Ono, “Measuring the Operating Condition of Induction Motor Using High-Sensitivity Magnetic Sensor,” *Sensors*, vol. 25, no. 14, p. 4471, Jan. 2025, doi: 10.3390/s25144471.

[103] G. A. Canada, “A Guide To Canada’s Export Controls - Groups 3 to 7 and Index,” GAC. Accessed: Sep. 09, 2025. [Online]. Available: https://www.international.gc.ca/controls-controles/about-a_propos/expor/guide-2013-2.aspx?lang=eng

[104] “Types.” Accessed: Nov. 05, 2025. [Online]. Available: https://www.sciencedirect.com/science/chapter/monograph/pii/B9780081023914000058

[105] J. Zhang and D. Zhang, “The Calculation of Thermal Resistance for Forced Air Cooling,” presented at the 2015 International Symposium on Material, Energy and Environment Engineering, Atlantis Press, Nov. 2015, pp. 622–625. doi: 10.2991/ism3e-15.2015.149.

[106] G. Boothroyd, P. Dewhurst, and W. A. Knight, *Product Design for Manufacture and Assembly, Revised and Expanded*, 2nd ed. Boca Raton: CRC Press, 2001. doi: 10.1201/9780824741587.

[107] InvenSense, Inc., “Recommendations for Mounting and Connecting InvenSense MEMS Microphones.” Accessed: Feb. 11, 2024. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2015/02/Recommendations-for-Mounting-and-Connecting-InvenSense-MEMS-Microphones.pdf

[108] R. Anslow and A. Devices, “How to Design a Good Vibration Sensor Enclosure Using Modal Analysis”.

[109] T. T. To, A. Al Mahmud, and C. Ranscombe, “A framework for integrating additive manufacturing into engineering education: perspectives of students and educators,” *Eur. J. Eng. Educ.*, vol. 50, no. 2, pp. 298–319, Mar. 2025, doi: 10.1080/03043797.2024.2358368.

[110] R. Lachmayer, T. Ehlers, and R. B. Lippert, “Introduction,” in *Design for Additive Manufacturing*, R. Lachmayer, T. Ehlers, and R. B. Lippert, Eds., Berlin, Heidelberg: Springer, 2024, pp. 1–6. doi: 10.1007/978-3-662-68463-4_1.

[111] K. V. Selivanov, A. O. Yakimov, G. A. Volkov, and D. S. Lyskov, “Prospects of Design of Electronic Equipment Enclosures by Means of Additive Technologies,” in *2024 6th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*, Feb. 2024, pp. 1–5. doi: 10.1109/REEPE60449.2024.10479765.

[112] Y. Dong and H. Hansen, “Design of an Internet of Things (IoT)-Based Photosynthetically Active Radiation (PAR) Monitoring System,” *AgriEngineering*, vol. 6, no. 1, pp. 773–785, Mar. 2024, doi: 10.3390/agriengineering6010044.

[113] “Experimental Investigation of Hybrid Propulsion Mechanism for Robotic Fish | FEDSM | ASME Digital Collection.” Accessed: Nov. 05, 2025. [Online]. Available: https://asmedigitalcollection-asme-org.proxy.bib.uottawa.ca/FEDSM/proceedings/FEDSM2022/85833/V001T03A039/1147110

[114] K. Jardine, E. Augusto, S. D. Levine, A. Sunder, S. Som, and J. Chambers, “Development of a lightweight, portable, waterproof, and low power stem respiration system for trees,” *MethodsX*, vol. 10, p. 101986, Jan. 2023, doi: 10.1016/j.mex.2022.101986.

[115] *Degrees of protection provided by enclosures (IP Code)*, IEC60529, Aug. 2013. Accessed: Jan. 02, 2025. [Online]. Available: https://webstore.iec.ch/en/publication/2452

[116] J. Y. Khan, “Introduction to IoT Systems,” in *Internet of Things (IoT)*, Jenny Stanford Publishing, 2019.

[117] “6 Types of Wireless IoT Technology.” Accessed: Oct. 23, 2025. [Online]. Available: https://www.zipitwireless.com/blog/5-types-of-wireless-iot-technology

[118] F. Wu, M. S. Arefin, J.-M. Redoute, and M. R. Yuce, “Low-Power Design Considerations for IoT Sensors,” in *Internet of Things (IoT)*, Jenny Stanford Publishing, 2019.

[119] “The Bluetooth® Mesh Networking Primer,” Bluetooth® Technology Website. Accessed: Oct. 23, 2025. [Online]. Available: https://www.bluetooth.com/bluetooth-mesh-networking-primer/

[120] “What is Zigbee? Learn About Zigbee Wireless Mesh Technology.” Accessed: Oct. 23, 2025. [Online]. Available: https://www.digi.com/solutions/by-technology/zigbee-wireless-standard

[121] “47 CFR 15.109 -- Radiated emission limits.” Accessed: Sep. 08, 2025. [Online]. Available: https://www.ecfr.gov/current/title-47/part-15/section-15.109

[122] I. Government of Canada, “ICES-003 — Information Technology Equipment (including Digital Apparatus).” Accessed: Sep. 07, 2025. [Online]. Available: https://ised-isde.canada.ca/site/spectrum-management-telecommunications/en/devices-and-equipment/interference-causing-equipment-standards-ices/ices-003-information-technology-equipment-including-digital-apparatus

[123] “47 CFR 15.209 -- Radiated emission limits; general requirements.” Accessed: Sep. 08, 2025. [Online]. Available: https://www.ecfr.gov/current/title-47/part-15/section-15.209

[124] I. Government of Canada, “RSS-Gen — General Requirements for Compliance of Radio Apparatus.” Accessed: Sep. 08, 2025. [Online]. Available: https://ised-isde.canada.ca/site/spectrum-management-telecommunications/en/devices-and-equipment/radio-equipment-standards/radio-standards-specifications-rss/rss-gen-general-requirements-compliance-radio-apparatus

[125] M. F. Acevedo, “Single-Board Computers and Microcontrollers,” in *Real-Time Environmental Monitoring*, CRC Press, 2015.

[126] Arm Ltd., “What is RISC?,” Arm | The Architecture for the Digital World. Accessed: Nov. 05, 2025. [Online]. Available: https://www.arm.com/glossary/risc

[127] Compaq *et al.*, *Universal Serial Bus Specification*, Apr. 2000.

[128] Apple Inc. *et al.*, *Universal Serial Bus 3.2 Specification*, Jun. 2022.

[129] J. M. Fiore, *Operational amplifiers and linear integrated circuits: theory and applications*, Version 3.2.11. Utica: James M. Fiore, 2024.

[130] A. Castaldo, “Switching regulator fundamentals.” Texas Instruments, 2012. Accessed: Nov. 05, 2025. [Online]. Available: https://www.ti.com/lit/an/snva559c/snva559c.pdf

[131] Texas Instruments, “TPS6300x High-Efficient Single Inductor Buck-Boost Converter With 1.8-A Switches datasheet (Rev. C).” Oct. 2015. Accessed: Nov. 05, 2025. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps63001.pdf

[132] EVE Energy, “EVE Primary Batteries.” Accessed: Jun. 09, 2025. [Online]. Available: https://www.novitronic.com/Datenbl%C3%A4tter/Akkus%20%20%26%20Batterien/EVE%20Primary%20Batteries.pdf

[133] Saft, “Datasheet-Li-SOCI2-Battery-MSR175B54.pdf,” Primary lithium battery: LS 26500. Accessed: Jun. 04, 2025. [Online]. Available: https://www.msr.ch/media/pdf/Datasheet-Li-SOCI2-Battery-MSR175B54.pdf

[134] Y. Sun, X. Qin, L. Li, Y. Zhang, J. Zhang, and J. Qi, “The Impact of Temperature on the Performance and Reliability of Li/SOCl2 Batteries,” *Energies*, vol. 17, no. 13, p. 3063, Jan. 2024, doi: 10.3390/en17133063.

[135] Guangzhou Ser Battery Technology Co., “High temperature capability – LiSOCL2 battery can be designed to withstand temperatures up to 165 °C.” Accessed: Oct. 23, 2025. [Online]. Available: https://serui-battery.com/News/gongsidongtai/high-temperature-battery-lisocl2-battery-200c.html

[136] “Cylindrical Primary Lithium: Handbook and Application Manual.” Energizer Brands, LLC. Accessed: Oct. 23, 2025. [Online]. Available: https://data.energizer.com/pdfs/lithiuml91l92_appman.pdf

[137] Energizer, “L91 Ultimate Lithium Datasheet.” Accessed: Jun. 04, 2025. [Online]. Available: https://data.energizer.com/pdfs/l91.pdf

[138] W. Sun, X. Tang, and Y. Wang, “Multi-metal–Organic Frameworks and Their Derived Materials for Li/Na-Ion Batteries,” *Electrochem. Energy Rev.*, vol. 3, no. 1, pp. 127–154, Mar. 2020, doi: 10.1007/s41918-019-00056-0.

[139] muRata, “Cylindrical Type Lithium Ion Secondary Battery: US18650VTC6.” Accessed: Jun. 09, 2025. [Online]. Available: https://www.murata.com/-/media/webrenewal/products/batteries/cylindrical/datasheet/us18650vtc6-product-datasheet.ashx?la=en-gb&cvid=20220207015421000000

[140] “Review Of Comparative Battery Energy Storage Systems (Bess) For Energy Storage Applications In Tropical Enviroments,” ResearchGate. Accessed: Oct. 23, 2025. [Online]. Available: https://www.researchgate.net/publication/327966044_Review_Of_Comparative_Battery_Energy_Storage_Systems_Bess_For_Energy_Storage_Applications_In_Tropical_Enviroments

[141] “Test of Eneloop AA BK-3MCCE 1900mAh (White).” Accessed: Sep. 06, 2025. [Online]. Available: https://lygte-info.dk/review/batteries2012/Eneloop%20AA%20BK-3MCCE%201900mAh%20(White)%20UK.html

[142] “eneloop - Consumer - Panasonic Energy Co., Ltd.” Accessed: Oct. 23, 2025. [Online]. Available: https://www.panasonic.com/global/energy/products/eneloop/en/lineup/eneloop.html

[143] “AA battery tests incl. freezer test,” Candle Power Flashlight Forum. Accessed: Oct. 23, 2025. [Online]. Available: https://www.candlepowerforums.com/threads/aa-battery-tests-incl-freezer-test.241606/

[144] Shenzhen Grepow Battery Co., “Wide-Temperature Rechargeable NiMH Battery,” Grepow Rechargeable Batteries. Accessed: Oct. 23, 2025. [Online]. Available: https://www.grepow.com/nimh-battery/wide-temperature-rechargeable-nimh-battery.html

[145] N. Brylski, “State of Charge Estimation Using Smart Battery Charger.” Texas Instruments, 2021.

[146] C. Sestok, “Dynamic Z-Track^{TM} Technology: An Advanced Battery Gauging Algorithm for Dynamic Load Applications.” Texas Instruments, 2025.

[147] D. Nku, “Powering a Wireless Sensor Network for Machine Condition Monitoring,” Jul. 04, 2022, *Université d’Ottawa / University of Ottawa*. doi: 10.20381/RUOR-27964.

[148] M. Arjovsky and L. Bottou, “Towards Principled Methods for Training Generative Adversarial Networks,” *ArXiv170104862 Cs Stat*, Jan. 2017, Accessed: May 09, 2022. [Online]. Available: http://arxiv.org/abs/1701.04862

[149] “Model TO670A01 | PCB Piezotronics.” Accessed: Aug. 27, 2025. [Online]. Available: https://www.pcb.com/products?m=to670a01

[150] “MachineDoctor | Predictive Maintenance with Wireless IOT Sensors,” Nanoprecise. Accessed: Aug. 27, 2025. [Online]. Available: https://nanoprecise.io/solutions/machinedoctor/

[151] School of Business Management, Bandung Institute of Technology, Indonesia, P. Aldo Wahjudi, H. Wandebori, and School of Business Management, Bandung Institute of Technology, Indonesia, “Marketing Strategy for Online Condition Monitoring (Case Study: Nanoprecise Machine Doctor),” *Int. J. Curr. Sci. Res. Rev.*, vol. 07, no. 08, Aug. 2024, doi: 10.47191/ijcsrr/V7-i8-13.

[152] S. Vedula, “Nanoprecise: Automated Predictive Maintenance,” Jan. 2020. Accessed: Aug. 30, 2025. [Online]. Available: https://iadc.org/wp-content/uploads/2020/01/02-Sunil-Vedula-Benchmark_Nanoprecise_Presentation.pdf

[153] “New Products for Engineers | IoT - MACHINE DOCTOR - LTE - Wireless - Vibration | Nanoprecise Sci Corp.” Accessed: Aug. 30, 2025. [Online]. Available: https://gspplatform.cfemedia.com/pe/productProfile/60dae9aa82bcfa234affa78a

[154] Spin Doctors Reliability, “Battery Life (for Upload Rate vs No. of Uploads/day),” The MachineDoctor^{TM} 6-in-1 Sensor. Accessed: Aug. 30, 2025. [Online]. Available: https://static1.squarespace.com/static/5ea9b62dfb35a3394b1decf1/t/5fbda049fa04221c71da77be/1606262858705/Battery+Life+%28for+upload+rate+vs+no+of+uploads+per+day%29.pdf

[155] “Treon Industrial Node 6,” Treon.fi. Accessed: Aug. 27, 2025. [Online]. Available: https://www.treonstore.com/treon-industrial-node-6.html

[156] “Treon Industrial Node 6 Default Configuration,” Treon Knowledgebase. Accessed: Aug. 30, 2025. [Online]. Available: https://kb.treon.fi/pathmesh/default_configuration/in6_config/

[157] “Cutting Costs with Wireless Condition Monitoring Sensors.” Accessed: Aug. 27, 2025. [Online]. Available: https://www.bakerhughes.com/bently-nevada/blog/cutting-cord-and-cost-wireless-sensors

[158] “Ranger Pro Wireless Condition Monitoring Device User Manual.” Baker Hughes.

[159] “Phantom Gen 3,” ERBESSD INSTRUMENTS. Accessed: Aug. 28, 2025. [Online]. Available: https://www.erbessd-instruments.com/phantom-generation-3/

[160] “Wireless High Range Triaxial Accelerometer for Condition Monitoring Sensor,” Electro Mechanical S. Accessed: Aug. 30, 2025. [Online]. Available: https://www.mechsol.ca/product-page/wireless-high-range-triaxial-accelerometer-for-condition-monitoring-sensor-pha

[161] “Phantom System Setup Guide v4.” Erbessd Instruments, Feb. 2024. Accessed: Aug. 30, 2025. [Online]. Available: https://www.erbessd-instruments.com/wp-content/uploads/2024/02/Phantom-Setup-Guide-v4-2-low.pdf?srsltid=AfmBOop4KahM4P2nol46fQfc0DGcsy-x9peryKcsrru51UZ7vUEQpSVP

[162] National Control Devices, LLC, “Wireless Vibration Temperature Sensor | MEMS 3-Axis Vibration Sensor,” NCD.io. Accessed: Aug. 30, 2025. [Online]. Available: https://store.ncd.io/product/industrial-iot-wireless-smart-vibration-sensor-gen4/

[163] V. Yordanov, “Standalone Smart Vibration Sensor Gen4 Datasheet,” NCD.io. Accessed: Aug. 30, 2025. [Online]. Available: https://ncd.io/blog/standalone-smart-vibration-sensor-v4-datasheet/

[164] P. L. Walter, “The history of the accelerometer 1920s-1996 - Prologue and Epilogue, 2006,” *Sound Vib.*, vol. 21, no. 1, pp. 84–92, 2007.

[165] “History | PCB Piezotronics.” Accessed: Sep. 30, 2025. [Online]. Available: https://www.pcb.com/about/history

[166] “USB-6212 Specifications - NI,” https://www.ni.com. Accessed: Sep. 20, 2025. [Online]. Available: https://www.ni.com/docs

[167] “Model M623C01 | PCB Piezotronics.” Accessed: Oct. 06, 2025. [Online]. Available: https://www.pcb.com/products?m=m623c01

[168] “Model 130F20 | PCB Piezotronics.” Accessed: Oct. 06, 2025. [Online]. Available: https://www.pcb.com/products?m=130f20

[169] “About DigiKey,” DigiKey. Accessed: Oct. 01, 2025. [Online]. Available: https://www.digikey.ca/en/resources/about-digikey

[170] “Sensors, Transducers | Motion Sensors | Accelerometers,” DigiKey Electronics. Accessed: Jul. 21, 2025. [Online]. Available: https://www.digikey.ca/en/products/filter/motion-sensors/accelerometers/515

[171] “IMUs (Inertial Measurement Units) | Motion Sensors | Electronic Components Distributor DigiKey.” Accessed: Jul. 21, 2025. [Online]. Available: https://www.digikey.ca/en/products/filter/imus-inertial-measurement-units/567

[172] “KX134-1211 - Data Sheet, Product Detail | ROHM.com.” Accessed: Jul. 25, 2025. [Online]. Available: https://www.rohm.com/products/sensors-mems/accelerometer-ics/kx134-1211-product#productDetail

[173] “KX132-1211 - Data Sheet, Product Detail | ROHM.com.” Accessed: Jul. 25, 2025. [Online]. Available: https://www.rohm.com/products/sensors-mems/accelerometer-ics/kx132-1211-product#productDetail

[174] “ADS1120 data sheet, product information and support | TI.com.” Accessed: Jul. 30, 2025. [Online]. Available: https://www.ti.com/product/ADS1120

[175] “Speed Optimization - ESP32-C3 - — ESP-IDF Programming Guide v5.4.1 documentation.” Accessed: May 06, 2025. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32c3/api-guides/performance/speed.html

[176] “Hardware Reference - ESP32-C3 - — ESP-IDF Programming Guide v5.5 documentation.” Accessed: Aug. 14, 2025. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32c3/hw-reference/index.html

[177] P. Virtanen *et al.*, “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nat. Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020, doi: 10.1038/s41592-019-0686-2.

[178] Apple, Inc. *et al.*, “Universal Serial Bus Type-C Cable and Connector Specification Release 2.4.” USB Implementers Forum, Inc., Oct. 2024. [Online]. Available: https://www.usb.org/document-library/usb-type-cr-cable-and-connector-specification-release-24

[179] “Deep sleep wake up time - ESP32 Forum.” Accessed: Oct. 04, 2025. [Online]. Available: https://esp32.com/viewtopic.php?t=1504

[180] “ICS-43434 Datasheet,” TDK InvenSense. Accessed: Sep. 05, 2025. [Online]. Available: https://invensense.tdk.com/download-pdf/ics-43434-datasheet/

[181] “Eneloop rechargeable batteries. Tests and more in 2025,” https://eneloop101.com/. Accessed: Sep. 06, 2025. [Online]. Available: https://eneloop101.com/batteries/eneloop-test-results/

[182] “Panasonic BK3HCCA8BA Eneloop Pro Rechargeable AA Batteries, 8 pk : Amazon.ca: Electronics.” Accessed: Nov. 16, 2025. [Online]. Available: https://www.amazon.ca/Panasonic-BK-3HCCA8BA-Pre-Charged-Rechargeable-Batteries/dp/B00MXCIK32

[183] “Panasonic BK-3MCCA16FA eneloop AA 2100 Cycle Ni-MH Pre-Charged Rechargeable Batteries, 16 Pack : PANASONIC: Amazon.ca: Health & Personal Care.” Accessed: Nov. 16, 2025. [Online]. Available: https://www.amazon.ca/Panasonic-BK-3MCCA16FA-Pre-Charged-Rechargeable-Batteries/dp/B00JHKSN4O

[184] “Power Designer.” Accessed: Sep. 06, 2025. [Online]. Available: https://webench.ti.com/power-designer/switching-regulator?powerSupply=0

[185] “TPS61022 data sheet, product information and support | TI.com.” Accessed: Sep. 06, 2025. [Online]. Available: https://www.ti.com/product/TPS61022

[186] “TPS564242 data sheet, product information and support | TI.com.” Accessed: Sep. 15, 2025. [Online]. Available: https://www.ti.com/product/TPS564242

[187] “Ultimate 3D Printing Material Properties Table.” Accessed: Aug. 20, 2025. [Online]. Available: https://www.simplify3d.com/resources/materials-guide/properties-table/

[188] M. Coletti, “Implementation of Relative Humidity Control in DMA Measurements,” *AIP Conf. Proc.*, vol. 1255, no. 1, pp. 82–84, Jun. 2010, doi: 10.1063/1.3455673.

[189] “Water Absorption of Nylons Pros, Cons and Busting Myths.” Accessed: Aug. 20, 2025. [Online]. Available: http://www.dotmar.com.au/insights/educational/water-absorption-of-nylons-pros-cons-and-busting-myths

[190] “McMaster-Carr - Wikipedia.” Accessed: Oct. 03, 2025. [Online]. Available: https://en.wikipedia.org/wiki/McMaster-Carr

[191] “Nitrile Rubber (Buna-N Material),” Denver Rubber Company. Accessed: Oct. 07, 2025. [Online]. Available: https://drcfirst.com/materials/sheet-rubber-and-elastomerics/nitrile/

[192] “Data Sheet: GORE AcousticVents Provide Reliable Protection for Microphones | Gore.” Accessed: Sep. 15, 2025. [Online]. Available: https://www.gore.com/resources/data-sheet-gore-acousticvents-provide-reliable-protection-microphones

[193] “McMaster-Carr.” Accessed: Oct. 06, 2025. [Online]. Available: https://www.mcmaster.com/

[194] B. Schweizer, “The low-frequency spectrum of small Helmholtz resonators,” *Proc. R. Soc. Math. Phys. Eng. Sci.*, vol. 471, no. 2174, p. 20140339, Feb. 2015, doi: 10.1098/rspa.2014.0339.

[195] “Custom O-Ring Manufacturer | Marco Rubber & Plastics.” Accessed: Oct. 03, 2025. [Online]. Available: https://www.marcorubber.com/about-us.htm

[196] “O-Ring Groove Design Guides - Marco Rubber.” Accessed: Oct. 03, 2025. [Online]. Available: https://www.marcorubber.com/o-ring-groove-design-directory.htm

[197] ASME, “Preferred Limits and Fits for Cylindrical Parts,” American Society of Mechanical Engineers, Standard ANSI/ASME B4.1-1967 (R2009), 2009.

[198] “PCB Manufacturing & Assembly Capabilities - JLCPCB.” Accessed: Nov. 05, 2025. [Online]. Available: https://jlcpcb.com/capabilities/pcb-capabilities
[199] “EL3D® ASA Filament, Black, 1Kg, 1.75,” Envirolaser3D. Accessed: Oct. 16, 2025. [Online]. Available: https://www.envirolaser3d.com/products/el3d%c2%ae-asa-filament-black-1kg-1-75
[200] “Understanding Bluetooth Range,” Bluetooth® Technology Website. Accessed: Nov. 14, 2025. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/
[201] Espressif Systems, “ESP32-C3-WROOM-02 User Manual.” Espressif Systems, Jul. 06, 2023. Accessed: Nov. 14, 2025. [Online]. Available: https://fcc.report/FCC-ID/2AC7Z-ESPC3WROOM/5875713.pdf
[202] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.
[203] J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.
[204] “resample — SciPy v1.16.1 Manual.” Accessed: Aug. 10, 2025. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample.html
[205] Espressif Systems, “ESP32-WROOM-32,” ESP32 Series of Modules. Accessed: Oct. 04, 2025. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
[206] W. Kester, *Practical design techniques for power and thermal management*. Norwood, MA: Analog Devices, 1998.

Appendix A Sensor and Enclosure Analysis

A.1 Average RTD Cost

The average cost of RTDs from a filtered Digi-Key Search is included here to use as a metric to evaluate the cost of potential device options against the cost of most RTDs.

As per section 4.1.4, the Digi-Key filters used to narrow down the list of RTDs to those desired in the MachMoS design are the following:

- Current: All available values between 0.05 mA and 1.5 mA
- Operating Temperature: All ranges containing -40°C to 120°C
- Mounting Type: *Free-Hanging, Through-Hole*
- Product Status: *Active*
- Stocking Options: *In Stock*
- Exclude: Marketplace *Products*
- Environmental Options: *RoHS Compliant*

These filters brought the count to 109 results. In this list of devices, the prices vary greatly, from 2$ CAD to 125$ CAD, with an average price of 16$ CAD (median of 5$ CAD). Considering the large gap between the median and average price, there is a large possibility of outliers. If RTD prices can be assumed to be a right-skewed population, it is possible to fit a χ^2 distribution to the log-prices distribution, and to calculate the price threshold for the 90th percentile RTD. To increase the population size and to target Pt RTDs more accurately, RTDs with other mounting types were reintroduced, and the *RTD Material* field was set to filter Pt RTDs only. The population size with these filters was 209. This analysis is done using Python code, using the *numpy* [202], *matplotlib* [203] and *scipy* [177] libraries:

```python
# Library imports
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2

# RTD price data
prices = np.log(np.array([…])) # Price data

# Fit chi-squared distribution to the data. The location parameter’s set to 0
df_fit, loc_fit, scale_fit = chi2.fit(prices, floc=0)
```

Using these prices, the fitted degree of freedom (DoF) is 11.16, and the scaling factor is 0.193. Once the initial fit is done, outliers in the 90th percentile and higher are removed. The χ^2 is re-fit, and a Kolmogorov-Smirnov goodness of fit test is performed on the new model to ensure the population follows a χ^2 distribution. The log-price data is plotted on a histogram with the χ^2 probability density function (PDF) overlayed for visual evaluation. This is done with the following code:

```python
# Calculate 90th percentile cutoff
cutoff_high = stats.chi2.ppf(0.90, df_fit, loc=loc_fit, scale=scale_fit)

# Remove outliers (prices in the 90th percentile and higher)
newlist = [log_price for price in prices if item <= cutoff_high]

# Re-fit chi-squared distribution to the data
df_fit, loc_fit, scale_fit = stats.chi2.fit(newlist, floc=0)

# Calculate how the price of the chosen RTD compares to the population
percentile_rtd_price = stats.chi2.cdf(np.log(5.55), df_fit, loc=loc_fit,
scale=scale_fit)

scaled_data = newlist/scale_fit
ks_stat, p_value = stats.kstest(scaled_data, 'chi2', args=(df_fit,0.0))
print(f"CDF percentile of RTD price: {percentile_rtd_price}")
print(f"P-value, K-S Stat: {p_value}, {ks_stat}")

# Plotting
x = np.linspace(0, max(newlist) + 20, 500)
pdf_fitted = stats.chi2.pdf(x, df_fit, loc=loc_fit, scale=scale_fit)

values, bins, bars = plt.hist(newlist, bins=20, density=True, alpha=0.6, color='grey',
label='Log-Price Data')

plt.plot(x, pdf_fitted, 'k-', label=f'Chi-squared Fit (df={df_fit:.2f})')
plt.xlabel('Log-Price ($ CAD)')
plt.ylabel('Density')
plt.title('Chi-Squared Fit to RTD Prices, Outliers Removed')
plt.grid(True)
plt.savefig("./RTD_prices_chi-pdf.svg")
```

from which a new fit using a DoF of 20.81 and a scaling factor of 0.0873 is obtained. The fit has $N=134$, $p=0.95$. The overlayed histogram and fitted χ^2 are shown in Figure A.1. From this code, it can be determined that the chosen RTD is more expensive than 46.7 % of RTDs (obtained from the cumulative distribution function, CDF, associated with the distribution).

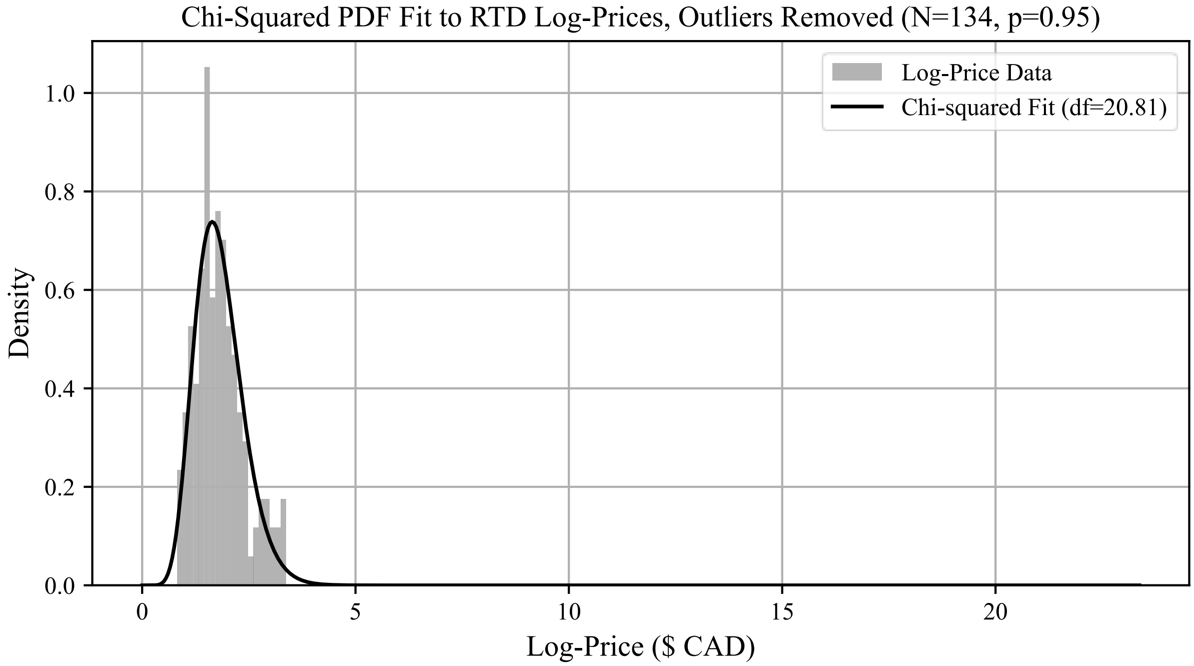


Figure A.1 - Histogram of RTD Costs, with χ^2 PDF Overlayed

A.2 Error Propagation on RTD Readings

An error propagation analysis is necessary to calculate the bounds of the expected accuracy of the MachMoS node RTD readings. During a temperature measurement, the ADC reads a voltage value across a variable resistor, the RTD, and compares it to a reference resistor. The value of the resistance of the RTD is computed using Equation (10) from section 4.1.4 (a simple ratio of ADC code to full-scale, multiplied by the reference resistor value). The corresponding temperature is then computed using the CVD equation (Equation (9) from section 2.8.1.1). For convenience, these equations are repeated below:

$$R_{\text{RTD}}(T)\begin{cases} R_{\text{RTD}}(0)\cdot(1+AT+BT^2) & \text{if } T \geq 0 \\ [1+AT+BT^2+C\cdot(T-100)T^3] & \text{if } T < 0 \end{cases}$$

where $R_{\text{RTD}}(T)$ is the resistance of the device, in Ohms, at temperature, T, in °C. A, B, C are coefficients with values $A = 3.908 \times 10^{-3}$, $B = -5.775 \times 10^{-7}$, $C = -4.183 \times 10^{-12}$ for platinum RTDs. In this measurement, there are two main components that have a tolerance attached, that is, the RTD, which has an A-rating (± 0.06 %), and the reference resistor of 1500 **Ω** which has a tolerance of ± 0.1 %, and a deviation of 25 ppm/°C. Assuming the ADC does indeed feature 16 noise-free bits, the assumption is made that the reading made by the ADC suffers only quantization noise (no value averaging) and gain error, that is, ±0.5 codes and 0.015 %. Additionally, since the ADC measurements are done ratiometrically, it means that error on the exact amount of excitation current flowing through the system is not important, as the currents cancel out (the same current flows through both the RTD and the reference as these are in series in the circuitry). That is, the voltages across the system are the important values, and the voltage across the RTD, referenced to the reference resistor voltage, is the value that is read by the ADC:

$$V_{\mathrm{RTD}} = I_{\mathrm{excitation}} R_{\mathrm{RTD}}, V_{\mathrm{REF}} = I_{\mathrm{excitation}} R_{\mathrm{REF}} \rightarrow \frac{V_{\mathrm{REF}}}{R_{\mathrm{REF}}} = \frac{V_{\mathrm{RTD}}}{R_{\mathrm{RTD}}}$$

$$\therefore R_{\mathrm{RTD}} = \frac{R_{\mathrm{REF}} V_{\mathrm{RTD}}}{V_{\mathrm{REF}}} = \frac{R_{\mathrm{REF}} N_{\mathrm{RTD}}}{N_{\mathrm{FS}}}$$

where $V_{\mathrm{RTD}}, R_{\mathrm{RTD}}$ and $V_{\mathrm{REF}}, R_{\mathrm{REF}}$ are the RTD and reference resistor voltages and resistances, respectively. N_{RTD} is the ADC reading, and N_{FS} is the full-scale of the ADC (2^{16}-1). Using the general law of uncertainty propagation, the quantisation uncertainty δ_{ADC}, gain uncertainty δ_{GAIN}, and the reference resistor δ_{REF} uncertainty leads, in this calculation step, to an RTD resistance uncertainty δ_{RTD} of:

$$\frac{\delta_{\mathrm{RTD}}}{R_{\mathrm{RTD}}} = \sqrt{\left(\frac{\delta_{\mathrm{ADC}}}{N_{\mathrm{RTD}}}\right)^2 + \left(\frac{\delta_{\mathrm{REF}}}{R_{\mathrm{REF}}}\right)^2 + \left(\frac{\delta_{\mathrm{GAIN}}}{N_{\mathrm{FS}}}\right)^2} = \sqrt{\left(\frac{0.5}{N_{\mathrm{RTD}}}\right)^2 + (0.1\ \%)^2 + (0.015\ \%)^2}$$

$$\therefore \delta_{\mathrm{RTD}} = R_{\mathrm{RTD}} \sqrt{1.0225\mathrm{E}^{-6} + \left(\frac{0.5}{N_{\mathrm{RTD}}}\right)^2}$$

Assuming linearity in the reference resistor's behaviour (meaning a 0.0375/°C), and using the CVD equation (shortened to $R_{\mathrm{RTD}}(T)$), the tolerance on the reading can be re-written as a function of the temperature, T (°C), instead of the resistances and ADC codes:

$$\delta_{\mathrm{RTD}}(T) = R_{\mathrm{RTD}}(T) \sqrt{1.0225\mathrm{E}^{-6} + \left(\frac{0.5}{(2^{16} - 1)\, R_{\mathrm{RTD}}(T)/(1500 + 3.75\mathrm{E}^{-2} T)}\right)^2}$$

When computed, this equation shows an uncertainty ranging from of ±0.812 Ω to ±1.515 Ω in the -50°C to 130°C range. This uncertainty can then be used in the computation of uncertainty in temperature measurement when used with the CVD equation. Since temperature in the CVD equation is a function of RTD resistance, and the RTD resistance at 0°C, the uncertainty on temperature δT as a function of the uncertainty on resistance δR can be written as:

$$\delta T = \sqrt{\left(\frac{\delta T}{\delta R}\delta R\right)^2 + \left(\frac{\delta T}{\delta R_0}\delta R_0\right)^2}$$

in which $\delta R = \delta R_{\mathrm{RTD}}$ and R_0 is the resistance of the RTD at 0°C. In general, the derivative of CVD with respect to (w.r.t.) R is rather complicated to obtain, especially for temperatures below 0°C, where the CVD is a 4th order equation, which does not have a closed-form solution. However, the inverse derivative can be used to obtain δT for this specific case in which the only interest is for small uncertainty propagation, in which small changes in R and R_0 cause proportional changes in T, scaled by $\frac{dT}{dR}$ (first-order propagation):

$$\frac{dT}{dR} = \frac{1}{dR/dT} \rightarrow \delta T = \frac{\delta R_{\mathrm{RTD}}}{\left(\frac{dR_{\mathrm{RTD}}}{dT}\right)} \text{ and}$$

$$\frac{dR_{\mathrm{RTD}}}{dT} = \begin{cases} R_{\mathrm{RTD}}(0) \cdot (A + 2BT) & \text{if } T \geq 0 \\ R_{\mathrm{RTD}}(0) \cdot (A + 2BT + 3C(T-100)T^2 + CT^3) & \text{if } T < 0 \end{cases}$$

As for the derivative of R_0 w.r.t. T, CVD is re-written as $R(T) = R_0F(T)$ where $F(T)$ is the polynomial in the CVD. The same inverse derivative principle, with first order propagation, is used:

$$R(T) = R_0F(T) \rightarrow \frac{dR(T)}{dT} = \frac{dR_0}{dT}F(T) + R_0\frac{dF(T)}{dT} \rightarrow 0 = \frac{dR_0}{dT}F(T) + R_0\frac{dF(T)}{dT}$$

$$-\frac{dR_0}{dT}F(T) = R_0\frac{dF(T)}{dT} \rightarrow \frac{dT}{dR_0} = -\frac{F(T)}{R_0\frac{dF(T)}{dT}} \rightarrow \delta T = -\delta R_0\frac{F(T)}{R_0\frac{dF(T)}{dT}} \text{ and}$$

$$\frac{dF(T)}{dT} = \begin{cases} A + 2BT & \text{if } T \geq 0 \\ A + 2BT + 3C(T-100)T^2 + CT^3 & \text{if } T < 0 \end{cases}$$

Finally, the uncertainty on temperature can be computed as:

$$\delta T = \sqrt{\left(\delta R_{\mathrm{RTD}} \frac{1}{\left(\frac{dR_{\mathrm{RTD}}}{dT}\right)}\right)^2 + \left(\delta R_0 \frac{F(T)}{R_0 \frac{dF(T)}{dT}}\right)^2}$$

Since the denominators are the same (once expanded):

$$\delta T = \frac{\sqrt{\delta R_{\mathrm{RTD}}^2 + \delta R_0^2 F(T)^2}}{R_0 \frac{dF(T)}{dT}}$$

and the final equation for RTD reading uncertainty can be obtained:

$$\delta T = \begin{cases} \dfrac{\sqrt{\delta R_{\mathrm{RTD}}^2 + \delta R_0^2 \cdot (1 + AT + BT^2)^2}}{R_0 \cdot (A + 2BT)} & \text{if } T \geq 0 \\ \dfrac{\sqrt{\delta R_{\mathrm{RTD}}^2 + \delta R_0^2 \cdot [1 + AT + BT^2 + C(T-100)T^3]^2}}{R_0 \cdot (A + 2BT + 3C(T-100)T^2 + CT^3)} & \text{if } T < 0 \end{cases} \tag{17}$$

A plot of uncertainty on temperature was generated from equation (17) and is shown in Figure A.2. This plot shows that the uncertainty increases as temperature increases, with a maximum for this application at (130°C, ±0.4032°C).

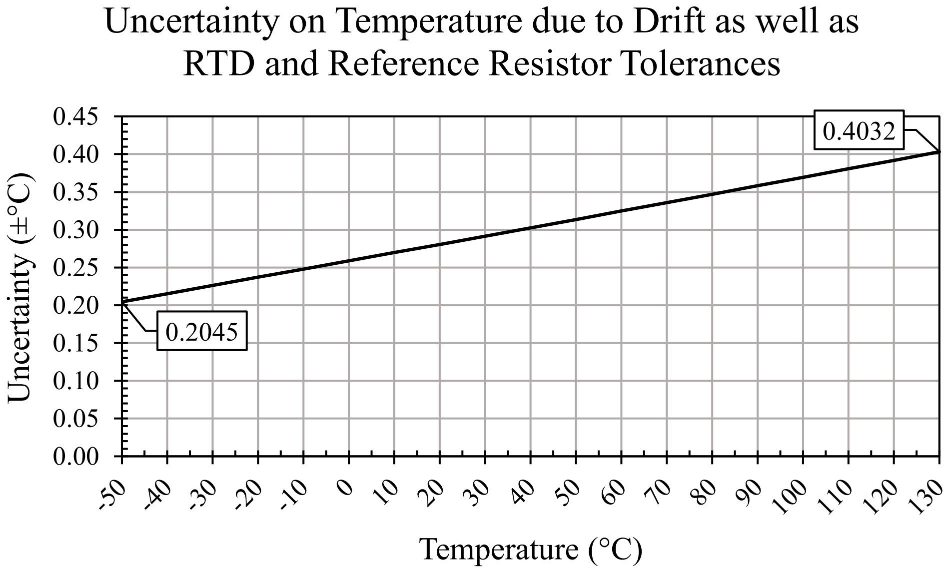


Figure A.2 - Tolerance on ADC Readings

A.3 I^{2}S Clock Maximum Frequency Accuracy

In this appendix, the accuracy on output frequency achievable by the ESP32-C3's I^{2}S subsystem and the error on clock frequency are calculated. For a 51.2 kHz output with a decimation ratio of 192, the ESP32-C3 must output a 9.8304 MHz clock, which requires a scaling factor of 0.06144 from the ESP32-C3 160 MHz system clock. In the case of the ESP32-C3, the input clock is scaled using a fractional divider, where the output is scaled by the ratio given in equation (18).

$$S = \frac{1}{N + b/a},\ a \neq 0 \tag{18}$$

where N, b and a are integers, and S the scaling factor [176]. Values of a and b generate values for registers within the I^{2}S subsystem fractional scaler: $x, y, z,$ and y_{n1}. These values are calculated using the equations in Table A.1 ($\lfloor \cdot \rfloor$ denotes the floor function, and % the modulo function).

Table A.1 - Fractional Scaler Coefficients and Calculations

Condition	x	y	z	y_{n1}
$b \leq \frac{a}{2}$	$\left\lfloor \frac{a}{b} \right\rfloor - 1$	$a\%b$	b	0
$b > \frac{a}{2}$	$\left\lfloor \frac{a}{a-b} \right\rfloor - 1$	$a\%(a-b)$	$a-b$	1

The calculated values are stored in 8-bit registers, such that these can range from 0 to 255 (except for y_{n1}, which only needs one bit). Similarly, a and b must be integers from 0 (exclusive) to 255, as per the equations for z. It must also be true that $a > b$ for x or y to have valid values (and for N not to be redundant). To obtain $S = 0.06144$, one can take $N = 16$ and $\frac{b}{a} = 0.2760417$, as $S^{-1} = 16.2760417$. To obtain the most accurate value possible for $\frac{b}{a}$, the Fractions module in Python can be used:

```
from fractions import Fraction
S = 0.06144 # Clock scaling ratio
best_fraction = Fraction(1.0/S - int(1.0/S)).limit_denominator(255)
```

which returns $b = 53$, $a = 192$. These values give an ideal fractional scaler, with no error on the desired output frequency. This calculation can be repeated for each valid accelerometer sampling rate and I^{2}S decimation ratios, with a resulting error on the I^{2}S output frequency. These are provided in Table A.2. The $x, y, z,$ and y_{n1} values are not included in the table.

Table A.2 - Recommended MachMoS Microphone and Accelerometer Sampling Rates

Microphone Rate (kHz)	Accelerometer ODR (kHz)	I^{2}S Scaling	Sys. Clock Ratio	N	b	a	Error (ppm)
6.40	6.40	192	0.00768	130	5	24	0.00
		384	0.01536	65	5	48	0.00
		576	0.02304	43	29	72	0.00
		768	0.03072	32	53	96	0.00
		1152	0.04608	21	101	144	0.00
12.8	12.8	192	0.01536	65	5	48	0.00
		384	0.03072	32	53	96	0.00
		576	0.04608	21	101	144	0.00
		768	0.06144	16	53	192	0.00
		1152	0.09216	10	188	221	1.45
19.2	6.40	192	0.02304	43	29	72	0.00
		384	0.04608	21	101	144	0.00
		576	0.06912	14	101	216	0.00
		768	0.09216	10	188	221	1.45
		1152	0.13824	7	18	77	4.16
25.6	25.6	192	0.03072	32	53	96	0.00
		384	0.06144	16	53	192	0.00
		576	0.09216	10	188	221	1.45
		768	0.12288	8	33	239	6.69
		1152	0.18432	5	94	221	1.45
32.0	6.40	192	0.0384	26	1	24	0.00
		384	0.0768	13	1	48	0.00
		576	0.1152	8	49	72	0.00
		768	0.1536	6	49	96	0.00
		1152	0.2304	4	49	144	0.00
38.4	12.8	192	0.04608	21	101	144	0.00
		384	0.09216	10	188	221	1.45
		576	0.13824	7	18	77	4.16
		768	0.18432	5	94	221	1.45
		1152	0.27648	3	95	154	4.16
44.8	6.40	192	0.05376	18	101	168	0.00
		384	0.10752	9	52	173	1.85
		576	0.16128	6	51	254	62.99
		768	0.21504	4	106	163	1.96
		1152	0.32256	3	1	10	64.00
51.2	25.6	192	0.06144	16	53	192	0.00
		384	0.12288	8	33	239	6.69
		576	0.18432	5	94	221	1.45
		768	0.24576	4	2	29	11.03
		1152	0.36864	2	62	87	11.03

A.4 Heat Transfer Simulations

In order to know the expected temperature delta between the machine under test and the measured temperature, and to determine if an aluminum or a steel enclosure base presents a smaller temperature difference, a series of heat transfer simulations are performed. This appendix will discuss the calculations performed as well as the simulation setup and the comparison. The Fundamentals of Heat and Mass Transfer book by Bergman, T. and Lavine, A. [88], chapter 9 on Free Convection describes approximate solutions for the same case as the MachMoS node encounters: a flat hot plate with natural convection at various orientations. The assumption that there is no heat transfer between the plate and the cover is made since the top side of the base is surrounded by a plastic cover. An assumption that there is insignificant radiative heat transfer from the plate is also made, as the bottom of the enclosure faces the MUT. Therefore, the temperature difference between the surface that would be irradiated and the enclosure temperature is minimal. The average convective heat transfer coefficient in a natural convection case is given by equation (29).

$$\bar{h} = \frac{k\overline{Nu}_L}{L} \tag{19}$$

where $\bar{h}$ is the average convective heat transfer coefficient over the plate, k the thermal conductivity of the interfacing gas, $\overline{Nu}_L$ is the Nusselt number, which, for an isothermal hot plate where laminar flow is assumed ($\mathrm{Ra}_L \lesssim 10^9$), is given by:

- Equation (20) for the bottom side of a horizontal plate,
- Equation (21) for a vertical plate,
- Equation (22) for the top side of a horizontal plate.

$$\overline{Nu}_L = 0.52Ra_L^{1/5} \quad (10^4 \lesssim \mathrm{Ra}_L \lesssim 10^9,\ Pr \gtrsim 0.7) \tag{20}$$

$$\overline{Nu}_L = 0.68 + \frac{0.670Ra_L^{1/4}}{[1 + (0.492/Pr)^{9/16}]^{4/9}} \quad (\mathrm{Ra}_L \lesssim 10^9) \tag{21}$$

$$\overline{Nu}_L = 0.54Ra_L^{1/4} \quad (10^4 \lesssim \mathrm{Ra}_L \lesssim 10^9,\ Pr \gtrsim 0.7) \tag{22}$$

where Ra_L is the Rayleigh number (non-dimensional) over the plate, given by equation (23).

$$Ra_L = Gr_L Pr = \frac{g \cdot \beta \cdot (T_s - T_\infty) \cdot L^3}{\nu \cdot \alpha} \tag{23}$$

where Gr_L is the Grashof number (dimensionless) over the plate, Pr the Prandtl number (dimensionless), g is the gravitational acceleration constant: 9.81 m/s^2, β is the inverse of the film temperature, T_s is the surface temperature of the plate exposed to air, T_∞ the ambient temperature, L the characteristic length of the plate, and ν, α are physical properties of the air, evaluated at the film temperature. The film temperature T_f and characteristic length L are defined in equation (24).

$$T_f \equiv \beta^{-1} \equiv \frac{T_s + T_\infty}{2} \qquad\qquad L \equiv \frac{A_s}{P} \tag{24}$$

where A_s is the area of the plate, and P its perimeter. Using these equations, as well as air's thermophysical properties at atmospheric pressure, it is possible to compute the Nusselt number (and the corresponding convection coefficient) for various ambient to surface deltas. The applicable thermophysical properties for air are given in Table A.3.

Table A.3 - Thermophysical Properties of Air at Atmospheric Pressure [88]

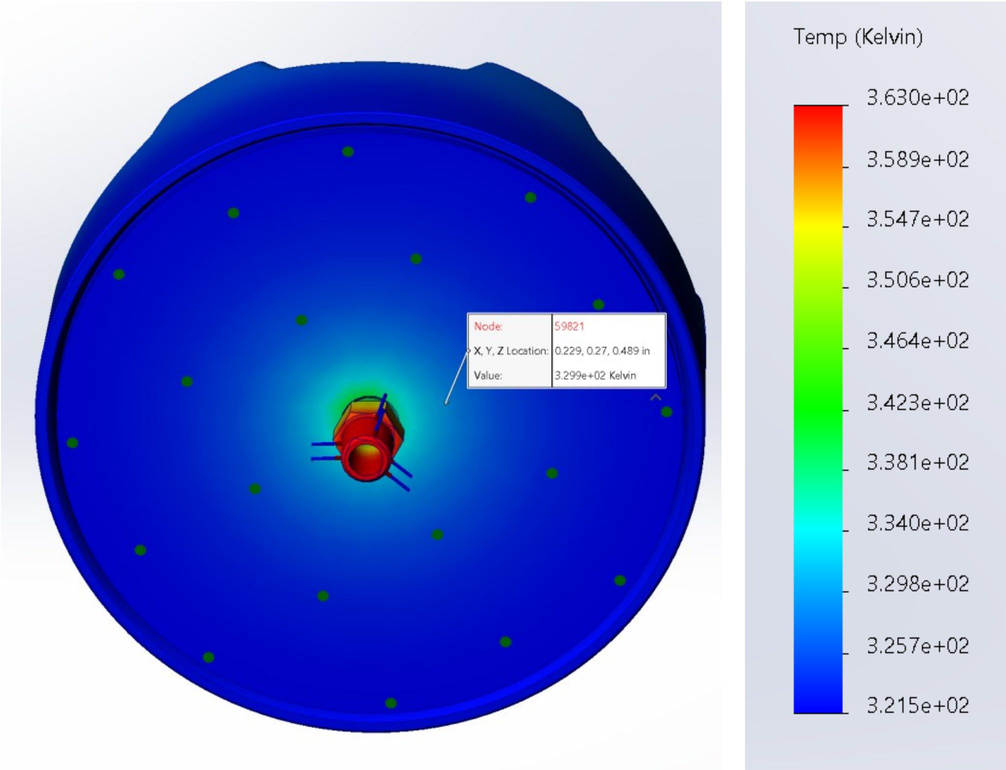
T (K)	ρ (kg/m2)	c_p (kJ/kg·K)	$\mu \cdot 10^6$ (N·s/m^2)	$\nu \cdot 10^6$ (m^2/s)	$k \cdot 10^3$ (W/m·K)	$\alpha \cdot 10^6$ (m^2/s)	Pr
200	1.7458	1.007	13.25	7.590	18.1	10.3	0.737
250	1.3947	1.006	15.96	11.44	22.3	15.9	0.720
300	1.1614	1.007	18.46	15.89	26.3	22.5	0.707
350	0.9950	1.009	20.82	20.92	30.0	29.9	0.700
400	0.8711	1.014	23.01	26.41	33.8	38.3	0.690

Since the models provided are for isothermal plates, they do not match the case of the MachMoS node, where a temperature gradient is expected to take place at steady state, and the plate is not isothermal, but rather has a constant heat flux, q_s''. In these cases, an acceptable approximation is to assume the temperature difference, ΔT, at the plate's mid-point, $L/2$, is the average plate temperature difference [88]. In the case of a round plate, the $L/2$ point would be at $R/2$, where R is the radius of the plate. The characteristic length of the MachMoS node base was computed as 28.4 mm. A SolidWorks simulation of the model can then be built from the assumptions made, and a trial-and-error approach conducted as follows:

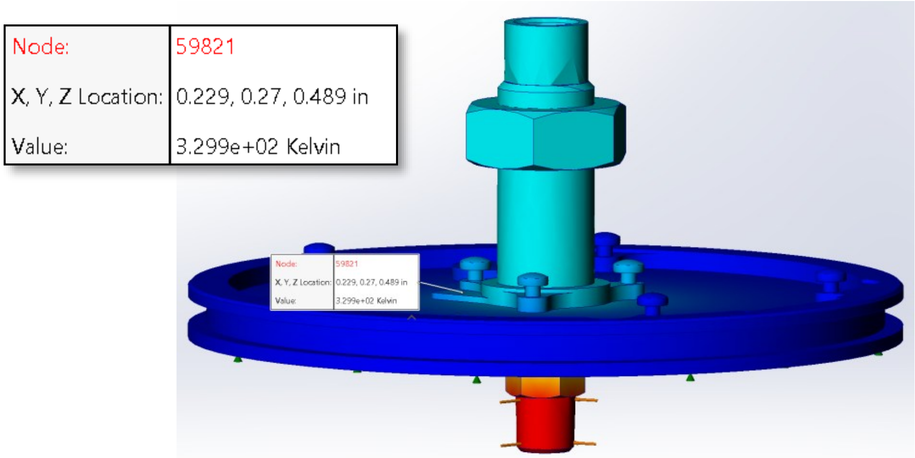
1. A temperature difference is guessed for the plate temperature at the midpoint of the plate.
2. The convective heat transfer coefficient is computed for the isothermal case using either equations (20), (21), or (22), then equation (19).
    a. Linear interpolation is used to obtain air's thermophysical properties using the data from Table A.3
3. The simulation is conducted. From the simulation result, the temperature at the midpoint of the plate is obtained.
4. Steps 1 to 3 are repeated until the surface temperature from the simulation near the plate's midpoint is within 1°C of the assumed average surface temperature from step 1.

The expected temperature difference from MUT to RTD can then be obtained for various MUT casing temperatures and ambient temperatures. An example result of a SolidWorks thermal simulation is shown in Figure A.3. This figure shows the thermal load applied at the bottom of the brass fitting, and the resulting temperature at the RTD platform, within the enclosure. The RTD is bonded to the platform using thermal epoxy.

The code used to automate iterating through the trial-and-error process described above is given in Appendix G.3.



(a) Bottom View



Node: 59821
X, Y, Z Location: 0.229, 0.27, 0.489 in
Value: 3.299e+02 Kelvin

(b) Top View, Cover Hidden to Show RTD Platform Within Enclosure

Figure A.3 - Example Resultant Temperature Heat Map from Thermal Simulations

The results of the simulations are given in Table A.5 to Table A.10. In these tables, T_{MUT} is the temperature of the machine under test (the temperature of the thermal load in the SolidWorks simulation), and T_{RTD} the temperature read by the RTD (the temperature at the RTD platform in the simulation). It is possible to derive models from the simulation results. The temperature discrepancy $\epsilon_T \equiv T_{\text{RTD}} - T_{\text{MUT}}$ is proportional to the difference between the RTD and ambient temperatures, as expressed in equation (25). The error is also inversely proportional to T_∞, as the ambient temperature affects T_s, α, and ν proportionately, which are denominators of Ra_L in equation (23).

$$\epsilon_T \propto a(T_{\text{RTD}} - T_\infty)^b / T_\infty^c \tag{25}$$

This model can be extended to horizontal plates facing down at an incline θ. As described in [88], the Raleigh number calculation for this case can use $g\cos\theta$ between 0° and 60°. Since the Raleigh number has a power of 1/5 in equation (20), a $\cos^{1/5}\theta$ term is added to correct for the incline. The fitted models for aluminum plates are provided in Table A.4 and provide estimates within 4.3 %.

Table A.4 - MUT vs. Measurement Discrepancy Models for Aluminum Plates

Case	Model	*a*	*b*	*c*
Facing Down, angle θ	$\epsilon_T = a\cos^{1/5}(\theta)\,(T_{\text{RTD}} - T_\infty)^b / T_\infty^c$	0.0515	1.19317	0.04560
Vertical	$\epsilon_T = a(T_{\text{RTD}} - T_\infty)^b / T_\infty^c$	0.1977	1.20863	0.18377
Horizontal, Facing Up	$\epsilon_T = a(T_{\text{RTD}} - T_\infty)^b / T_\infty^c$	0.2606	1.21621	0.24286

Table A.5 - Simulation Results: Aluminum Plate, Horizontal, Facing Down

Temperature Differential ($T_{\mathrm{MUT}} - T_{\infty}$)	Ambient Temperature (T_{∞}, °C)					
	-40	-20	0	20	40	50
	Temperature Discrepancy ($T_{\mathrm{MUT}} - T_{\mathrm{RTD}}$)					
10	0.6	0.6	0.6	0.6	0.6	0.6
20	1.3	1.3	1.3	1.3	1.3	1.3
30	2.1	2.1	2.1	2.1	2.1	2.1
40	3.0	3.0	3.0	3.0	2.9	2.9
50	3.9	3.9	3.9	3.9	3.8	3.8
60	4.8	4.8	4.8	4.8	4.8	4.7
70	5.8	5.8	5.7	5.7	5.7	5.7
80	6.7	6.7	6.7	6.7	6.7	
90	7.7	7.7	7.7	7.7		
100	8.8	8.8	8.7	8.7		
110	9.8	9.8	9.7			
120	10.8	10.8	10.8			
130	11.9	11.9				
140	13.0	12.9				
150	14.1					
160	15.2					

Table A.6 - Simulation Results: Steel Plate, Horizontal, Facing Down

Temperature Differential ($T_{\mathrm{MUT}} - T_{\infty}$)	Ambient Temperature (T_{∞}, °C)					
	-40	-20	0	20	40	50
	Temperature Discrepancy ($T_{\mathrm{MUT}} - T_{\mathrm{RTD}}$)					
10	0.7	0.7	0.7	0.7	0.7	0.7
20	1.6	1.6	1.6	1.6	1.6	1.6
30	2.6	2.6	2.6	2.6	2.6	2.6
40	3.7	3.6	3.6	3.6	3.6	3.6
50	4.8	4.7	4.7	4.7	4.7	4.7
60	5.9	5.9	5.9	5.8	5.8	5.8
70	7.0	7.0	7.0	7.0	7.0	
80	8.2	8.2	8.2	8.1	8.1	
90	9.4	9.4	9.4	9.4		
100	10.7	10.7	10.6	10.6		
110	11.9	11.9	11.9			
120	13.2	13.2	13.1			
130	14.5	14.5				
140	15.8	15.7				
150	17.1					
160	18.5					

Table A.7 - Simulation Results: Aluminum Plate, Vertical

Temperature Differential ($T_{\text{MUT}} - T_\infty$)	Ambient Temperature (T_∞, °C)					
	-40	-20	0	20	40	50
	Temperature Discrepancy ($T_{\text{MUT}} - T_{\text{RTD}}$)					
10	1.0	1.0	1.0	1.0	1.0	1.0
20	2.3	2.3	2.3	2.2	2.2	2.2
30	3.8	3.7	3.7	3.6	3.6	3.6
40	5.3	5.2	5.1	5.1	5.0	5.0
50	6.9	6.8	6.7	6.6	6.6	6.5
60	8.5	8.4	8.3	8.2	8.1	8.1
70	10.2	10.1	10.0	9.8	9.8	9.7
80	11.9	11.8	11.7	11.5	11.4	
90	13.7	13.5	13.4	13.2		
100	15.5	15.3	15.1	15.0		
110	17.3	17.2	16.9			
120	19.2	19.0	18.7			
130	21.0	20.8				
140	22.9	22.6				
150	24.8					
160	26.8					

Table A.8 - Simulation Results: Steel Plate, Vertical

Temperature Differential ($T_{\text{MUT}} - T_\infty$)	Ambient Temperature (T_∞, °C)					
	-40	-20	0	20	40	50
	Temperature Discrepancy ($T_{\text{MUT}} - T_{\text{RTD}}$)					
10	1.2	1.2	1.2	1.2	1.2	1.2
20	2.8	2.8	2.7	2.7	2.7	2.6
30	4.5	4.4	4.4	4.3	4.3	4.3
40	6.3	6.2	6.1	6.1	6.0	6.0
50	8.2	8.1	8.0	7.9	7.8	7.8
60	10.1	10.0	9.9	9.8	9.7	9.6
70	12.1	12.0	11.9	11.7	11.6	
80	14.1	14.0	13.8	13.7	13.5	
90	16.2	16.0	15.9	15.7		
100	18.3	18.1	17.9	17.7		
110	20.4	20.2	20.0			
120	22.6	22.4	22.1			
130	24.7	24.5				
140	27.0	26.7				
150	29.2					
160	31.4					

Table A.9 - Simulation Results: Aluminum Plate, Horizontal, Facing Up

Temperature Differential $(T_{\mathrm{MUT}} - T_\infty)$	Ambient Temperature (T_∞, °C)					
	-40	-20	0	20	40	50
	Temperature Discrepancy $(T_{\mathrm{MUT}} - T_{\mathrm{RTD}})$					
10	1.0	1.0	0.9	0.9	0.9	0.9
20	2.3	2.2	2.2	2.2	2.1	2.1
30	3.7	3.6	3.6	3.5	3.5	3.4
40	5.2	5.1	5.0	5.0	4.9	4.9
50	6.8	6.7	6.6	6.5	6.4	6.3
60	8.4	8.3	8.2	8.0	7.9	7.9
70	10.1	9.9	9.8	9.6	9.5	
80	11.8	11.7	11.5	11.3	11.2	
90	13.6	13.4	13.2	13.0		
100	15.4	15.2	14.9	14.7		
110	17.2	17.0	16.7			
120	19.0	18.8	18.5			
130	20.9	20.6				
140	22.8	22.4				
150	24.7					
160	26.6					

Table A.10 - Simulation Results: Steel Plate, Horizontal, Facing Up

Temperature Differential $(T_{\mathrm{MUT}} - T_\infty)$	Ambient Temperature (T_∞, °C)					
	-40	-20	0	20	40	50
	Temperature Discrepancy $(T_{\mathrm{MUT}} - T_{\mathrm{RTD}})$					
10	1.2	1.2	1.1	1.1	1.1	1.1
20	2.7	2.7	2.6	2.6	2.6	2.5
30	4.4	4.3	4.3	4.2	4.1	4.1
40	6.2	6.1	6.0	5.9	5.8	5.8
50	8.1	7.9	7.8	7.7	7.6	7.6
60	10.0	9.8	9.7	9.6	9.4	9.4
70	12.0	11.8	11.7	11.5	11.3	
80	14.0	13.8	13.6	13.4	13.3	
90	16.1	15.9	15.6	15.4		
100	18.2	17.9	17.7	17.4		
110	20.3	20.0	19.7			
120	22.4	22.2	21.8			
130	24.6	24.3				
140	26.8	26.5				
150	29.0					
160	31.3					

Appendix B Data Acquisition and Processing

B.1 Aliasing Example

A common example of aliasing is that of a video of a spinning car wheel. Video cameras capture our world, a continuous system, in a discrete fashion, usually in the range of 20 to 60 frames per second (FPS), that is, taking 20 to 60 pictures (discrete samples) per second. If a camera is used to capture a video of a spinning car wheel on the highway at a sampling rate of 24 FPS, $f_{\text{max}} = 12\ s^{-1}$, and that wheel is spinning at less than 12 revolutions per second (Hz), then it is possible to use the information in the video to determine the exact rotational speed of the tire by taking the difference in the angle of the tire between two frames (which will be less than a half rotation) and dividing by the period between two frames in order to obtain a rotational speed. Alternatively, if that same sampling rate is used to capture a wheel rotating at exactly 12 Hz, every frame captured will display the wheel either at the top or at the bottom of its rotation. In this case, it is impossible to determine the rotational speed (or even the direction) of the wheel, as the wheel could be going forwards (or backwards) at 12 Hz. It should be noted that the wheel could also be going at any odd multiples of that speed (12 Hz, 36 Hz, 60 Hz, etc.). At this car speed (and at higher car speeds), it can be said that the sampled signal suffers from aliasing. If the car accelerates, the tire would now be rotating at more than 12 Hz. In this case, the tire will appear to be spinning at an increasingly slower rate, until it seems to come to a stop at exactly 24 Hz. It will then seem to start reversing at speeds above 24 Hz. This is due to every frame in the video showing a wheel at the same position at every frame when rotating at 24 Hz, and a wheel slightly past a full rotation every frame beyond 24 Hz. Knowing this, an important question arises: how does one know if the tire was turning at less than 12 Hz in the first scenario? This is impossible to know without, for instance, knowing the maximum speed of the road on which the car was travelling (assuming of course traffic laws were followed). If it is known that the car could not have been travelling at more than the speed required to produce 12 rev/s of its wheels, then there is a certainty that the perceived rotational speed of the wheels is under 12 rev/s. This analogy shows the importance of taking aliasing seriously. Once an aliased signal (wheel) is captured, it can be impossible to know if the original signal contained a specific frequency (wheel was turning at a specific rate), or if the frequency perceived (calculated rate of rotation) is simply aliased into the captured signal. By using low pass filters (speed limits) that do not allow for higher frequency signals (fast rotating wheels), one can guarantee that the signals captured do not suffer aliasing.

Using plots of sine waves and their respective frequency contents, aliasing can also be illustrated. In Figure B.1, a fast continuous 10 Hz signal ($f_N = 20$ Hz) is sampled discretely at a rate lower than the Nyquist frequency for this signal ($f_S = 17.5$ Hz). This leads to a perceived signal (shown by the dotted line) which is slower than the real signal. In Figure B.2, the frequency content for each sampling rate is shown. In the figure, it can be observed that the perceived frequency for the signal when captured at 21 Hz, a sampling frequency above the Nyquist rate (20 Hz), is exactly 10 Hz. However, when capturing below the Nyquist rate (17.5 Hz), the perceived signal frequency is 7.5 Hz.

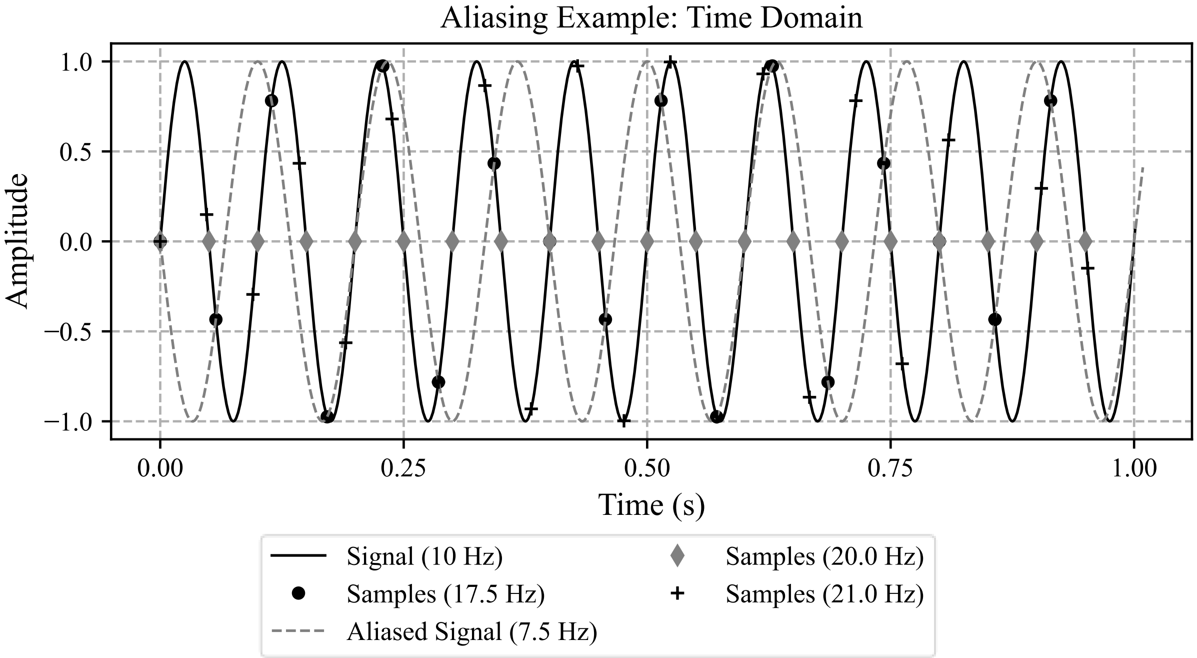


Figure B.1 - Signal and Aliased Capture with Various Other Sampling Rates

The aliased frequency content is often considered a mirror of the actual frequency content, in which the aliased frequency content can be obtained by folding the aliased frequency across the $f_{\max}$ of the sampling system. The phenomenon is exactly observable in Figure B.2, in which the 17.5 Hz sampled signal's frequency content is folded across $f_{\max}$ for a 17.5 Hz sampling system (8.75 Hz).

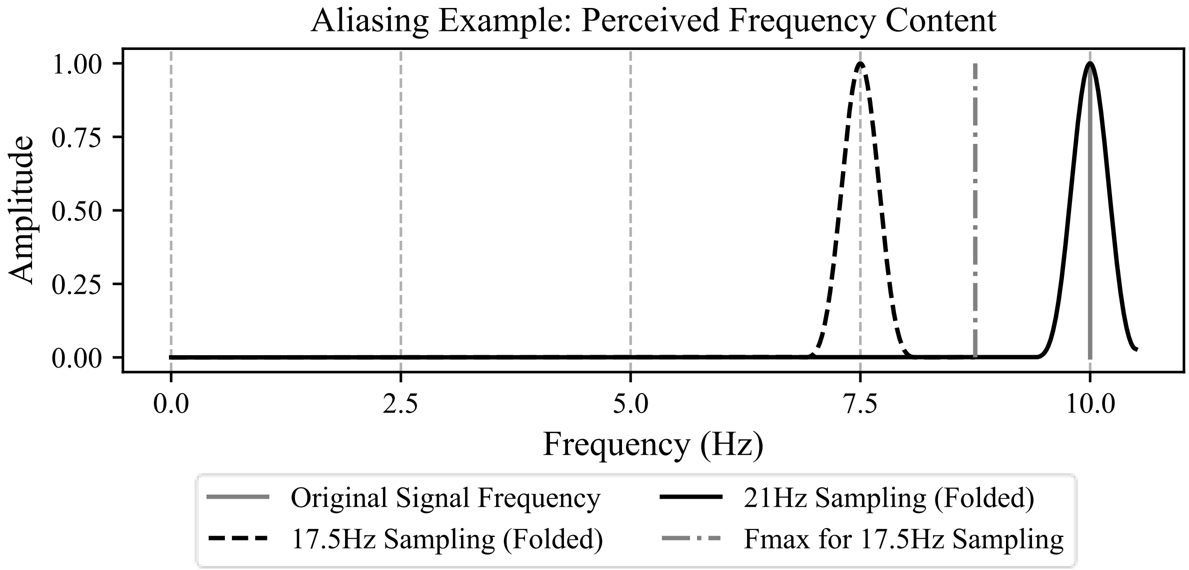


Figure B.2 - Frequency Content of Aliased and Non-Aliased Signal, Showing the Folding

Aliasing is also exactly observable in the car wheel analogy, in which the perceived rotational speed of the wheel increases in the video until it reaches 12 Hz ($f_{\max}$), where it then seems to slow down, as the wheel is accelerating, due to the folding of the actual spinning frequency across $f_{\max}$. Once the wheel reaches exactly 24 Hz, the perceived frequency is 0 Hz, and then the frequency becomes negative beyond 24 Hz, leading to wheels seemingly turning backwards.

B.2 Synchronising Data

Synchronizing data can be done very simply using the resampling functions in Python's *scipy* library. The resampling function takes the Fourier transform of a signal, and then inverses that transform to reconstitute the signal, but with a new sampling rate [204]. For instance a 1 Hz sinusoidal signal is sampled at a 13 Hz sampling rate with one data stream, and the other data stream samples at a sampling rate of 25 Hz:

```
# The signal to analyze
def s(t): return np.sin(t*(2*np.pi))

seconds = 5
total_samples = 10000
t = np.linspace(0, seconds, total_samples)
sine_1Hz = s(t)

# Channel 1 samples at 13Hz
chan1_sampling, chan1 = 13, []
for time in np.arange(0, seconds, 1/chan1_sampling): chan1.append(s(time))

# Channel 2 samples at 25Hz
chan2_sampling, chan2 = 25, []
for time in np.arange(0, seconds, 1/chan2_sampling): chan2.append(s(time))

fig, ax = plt.subplots()
ax.plot(t, sine_1Hz, "k--", label="Sample Signal (1 Hz)", linewidth=0.8)
ax.set_title("Sinusoidal Signal Sampled at Dissimilar Rates")
ax.set_xlabel("Time (s)")
ax.set_ylabel("Amplitude")
ax.plot(np.arange(0, seconds, seconds/len(chan1)), chan1, "s", color="Gray",
label=f"Channel 1 ({chan1_sampling} Hz)")
ax.plot(np.arange(0, seconds, seconds/len(chan2)), chan2, "ko", label=f"Channel 2
({chan2_sampling} Hz)")
ax.legend(loc='upper right'
```

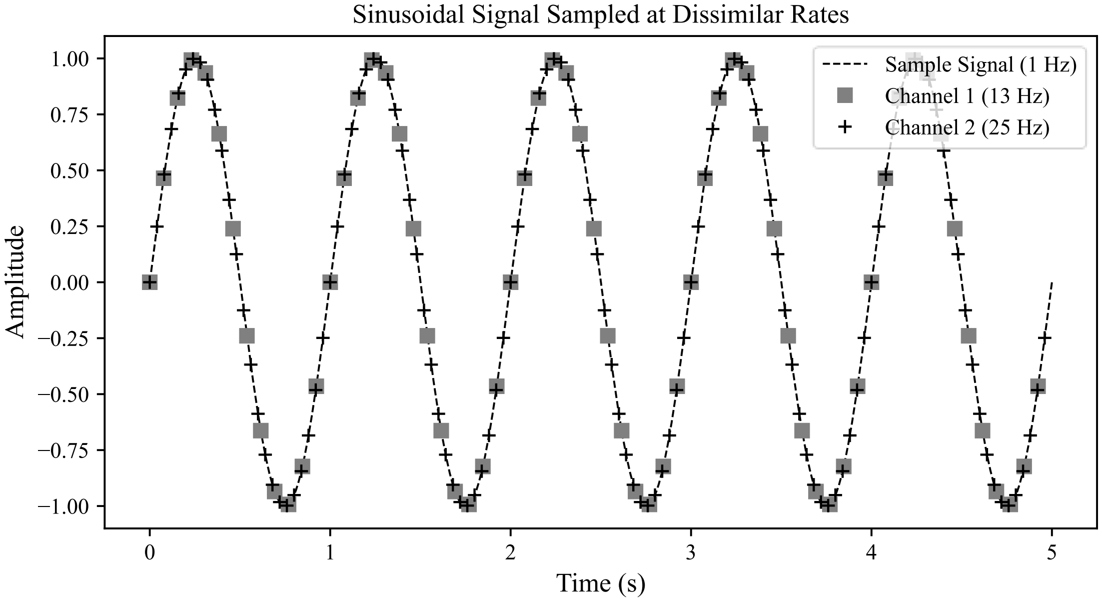


Figure B.1 - 1 Hz Sine Signal Sampled at Dissimilar Rates

As expected, it can be seen in Figure B. that the Channel 1 sampling (13 Hz) is offset from the Channel 2 sampling. Channel 1 can then be resampled to an appropriate number of samples:

```
chan1_resampled = scipy.signal.resample(chan1, chan2_sampling*seconds)
```

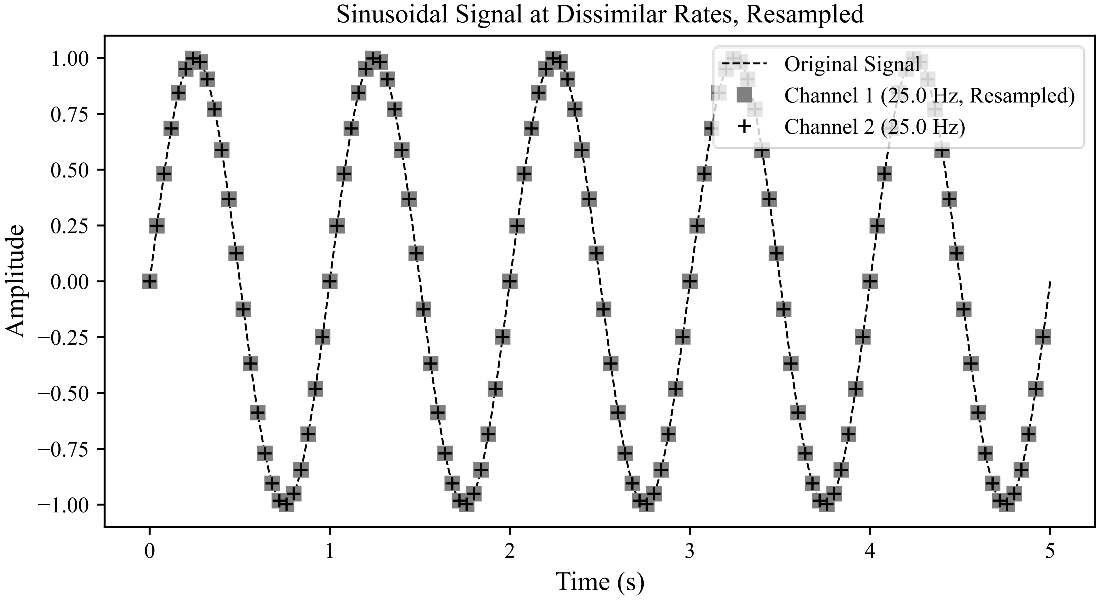


Figure B.2 - 1 Hz Sine Signal Sampled at Dissimilar Rates, Resampled

It can be seen in Figure B.2 that both sampling channels are now overlapping, despite the originally dissimilar sampling rates. Of course, the data on which this method would be used is more complex than a 1 Hz wave, sampled at 13 times that rate. It is important to understand the limitations of the method. As a first test, a more complex signal, with four components, one of them offset. The sampling is done using the same code:

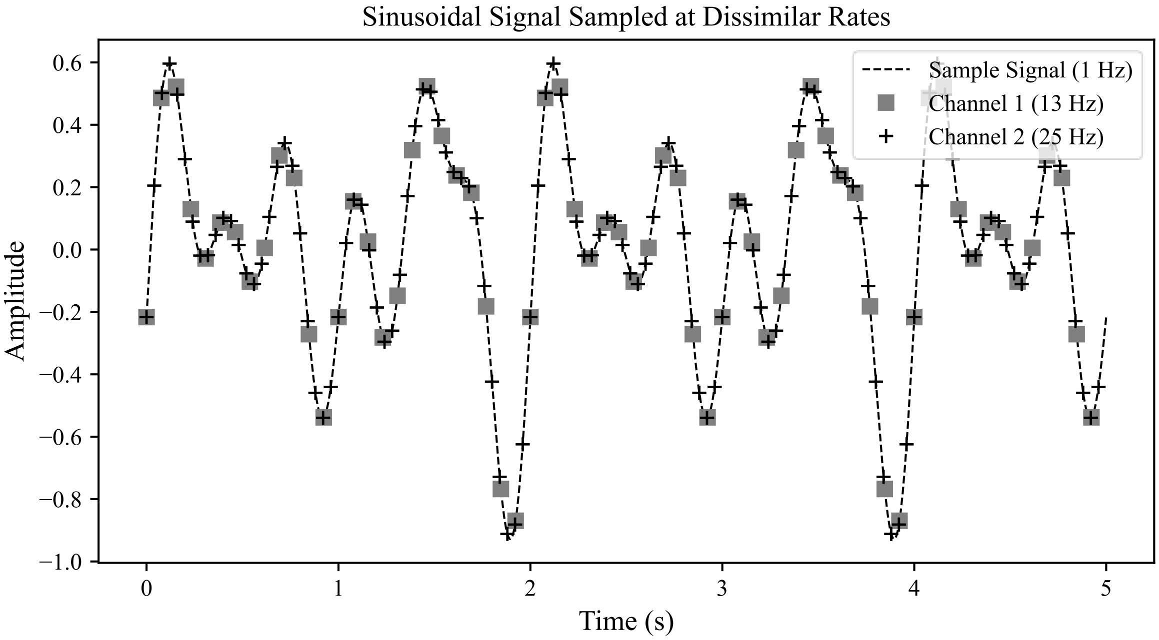


Figure B.3 - Dissimilar Sampling with Four Components

and the resampling is also done the same way:

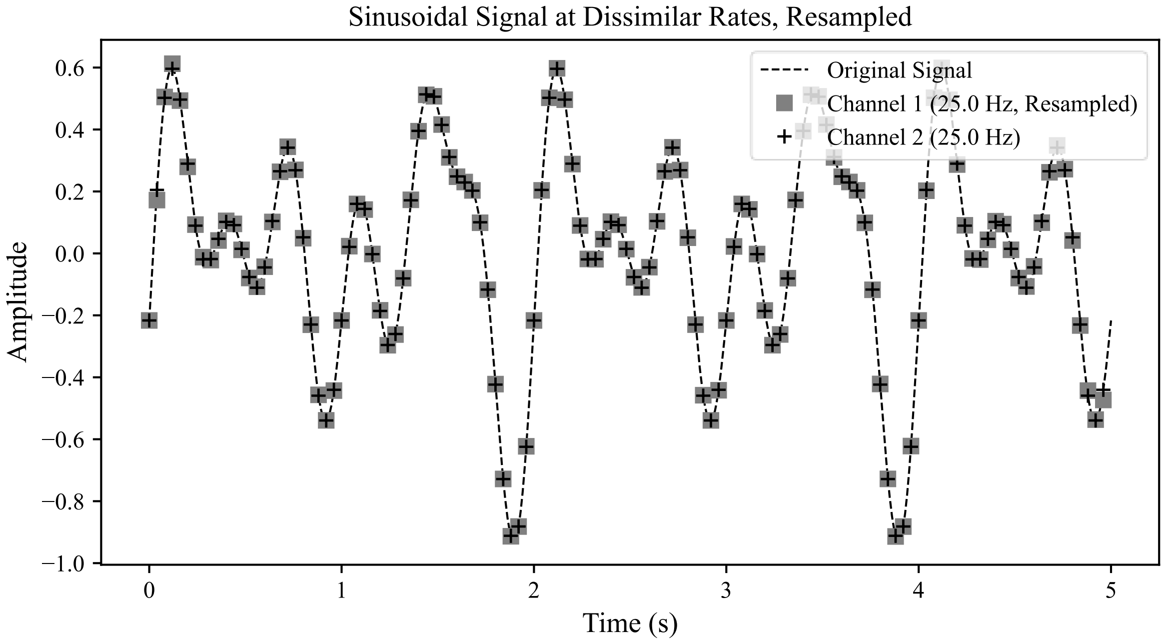


Figure B.4 - Resampled Four-Component Signal

The resampling can be considered very successful. However, the resampling is not as accurate near the start and end of the signal. This can potentially be attributed to spectral leakage, where the start and end of the sampled signal don't seem to match up, making the signal seem less periodic than it is, causing issues during the Fourier Transform and Inverse Fourier Transform. Applying windowing to the method helps, and the effect is attenuated if the original sampling rate is increased, but of course, increasing the sampling rate is not a simple task. Moreover, real applications use sampling rates in the 10 to 50 kHz range, and the effect of small errors at the start and end of the signal can simply be truncated, as necessary, or simply ignored. A quick test in which the absolute error between a Channel 1 sampling rate of 51.6 kHz (the MachMoS microphone maximum rate), and a Channel 2 rate of 25.6 kHz (the MachMoS accelerometer maximum rate), is shown in Figure B.5, where a 10 kHz component is added to the signal to attempt to affect the resampling.

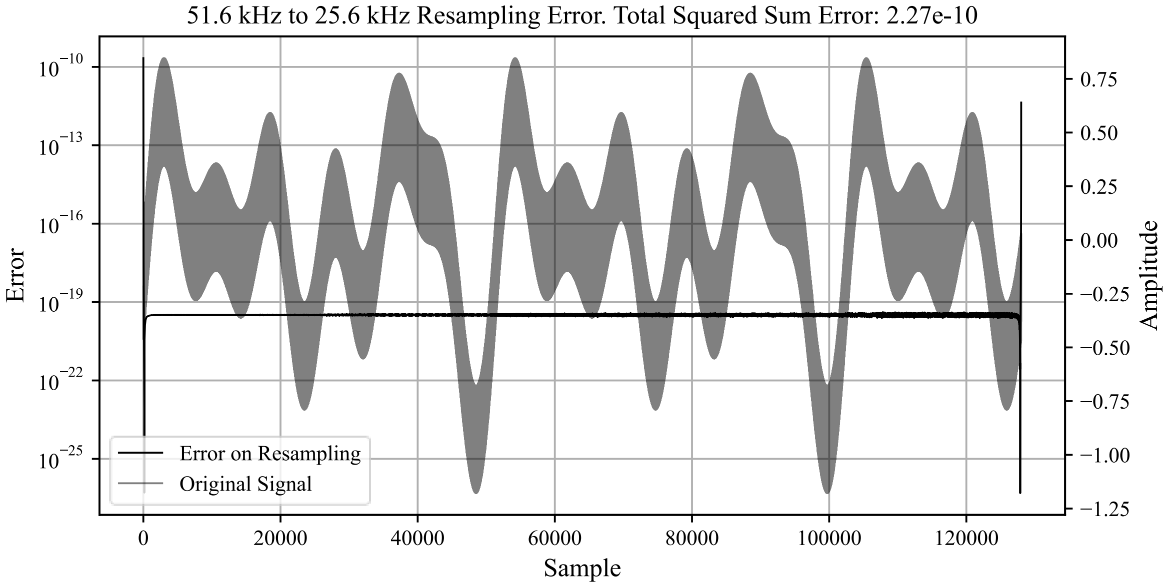


Figure B.5 - Resampling Error, High Sample Count and Frequency

As can be seen, the total error (squared sum) of this signal would be in the nano-g range, with most of the contribution in error being focused on the edges of the signal. Finally, an important note should be made about resampling with original sampling rates lower than the Nyquist frequency. It should be evident that this is not possible, and the resampling will lead to a low pass filtered sample. For instance, using the original low frequency signals (of which one is 3 Hz) and a sampling rate of 6 Hz on the first channel, the up-sampled signal is far from the original signal, as shown in Figure B.6. This is normal, and the behaviour should be perceived as a feature of the method. The result is simply a reconstruction of the raw signal from the frequency content. It should be noted that resampling will not yield the original input signal if that signal is not captured at a frequency higher than the Nyquist frequency.

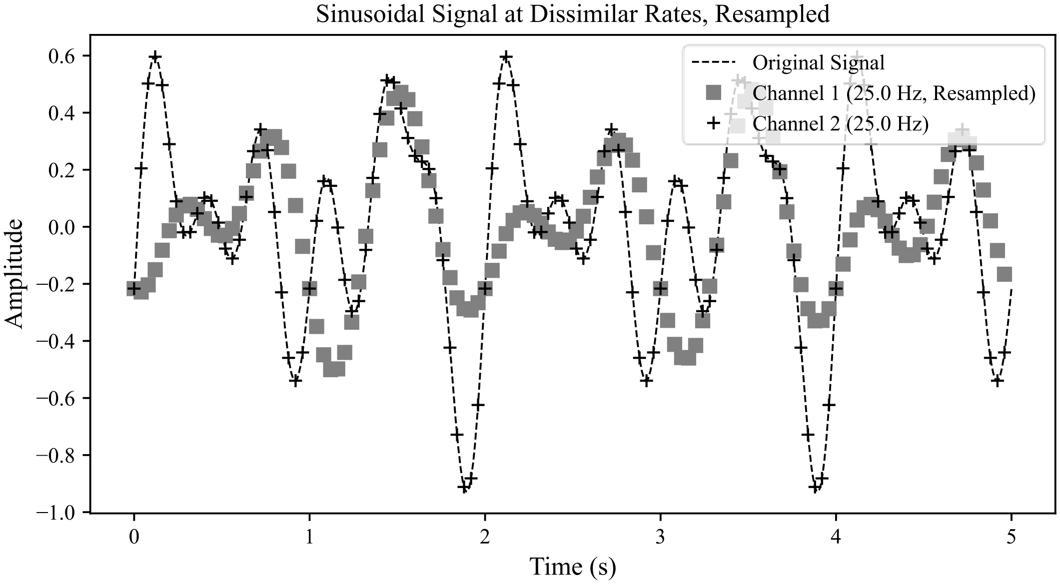


Figure B.6 - Up-Sampled Capture Below the Nyquist Frequency

The same signal, captured at 7 Hz is resampled adequately (with some error at the extremes), as shown in Figure B.7.

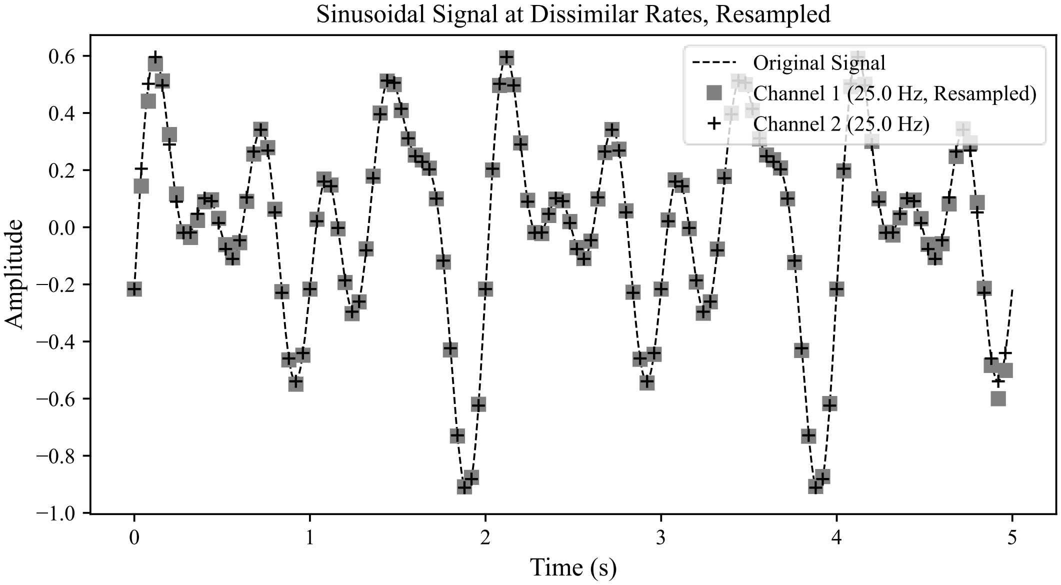


Figure B.7 - Up-Sampled Capture Above the Nyquist Frequency

B.3 Improving the Data Rate of the Node

The ESP32-C3 has a single core, and the transfer of RAM data to flash memory is done through the SPI interface. This requires the MCU's core. That is, the transfer to flash memory cannot happen at the same time as the capture of accelerometer data, as communicating with the accelerometer also uses the SPI interface. Thankfully, the microphone data is stored to RAM using Direct Memory Access (DMA), which does not require the MCU's core, such that microphone and accelerometer data can in fact be captured simultaneously.

The flash device can write data quickly (quad-SPI, 80 MHz), but does so using a page-write program, which supports the writing of only one 256-byte (B) page at a time. Each execution of the page-write program requires some IC down-time, severely reducing the true write speed of the flash memory. A quad-SPI transfer can communicate four bits per clock cycle. At 80 MHz, the quad-SPI data transfer rate is 320 Mbps.

However, the ESP-WROOM's flash IC typically requires 0.8 ms to execute a page-write program. In some cases, the page-write program may take up to 5 ms [205]. As such, the typical continuous write speed to flash using 256-byte pages is 2.56 Mbps (320 kBps). This is slightly faster than the maximum rate at which data can be generated by the sensors in the MachMoS node. At its slowest, the effective transfer rate to flash memory is 0.410 Mbps (51.2 kBps). Of course, most of the transfer time is on the flash IC side, with the actual transfer of 256 B, during which the ESP32-C3's core is busy, taking only 25.6 µs, 3.2 % to 0.5 % of the page write time.

This competition between data generation and storage is of concern as it necessitates reliance on the ESP32-C3's RAM, which is limited to 384 kB. One way to mitigate reliance on the ESP32-C3's RAM is to include multiple small quad-SPI flash ICs on the MachMoS node, or to include an SPI RAM IC. If multiple flash ICs were used, the objective would be to spread the page write program dead time among multiple devices, not to mention the low probability of multiple slow devices. However, this implementation would make reconstituting the data challenging. If a SPI RAM IC were used instead, the objective would be to avoid page-write program dead-time altogether, with the caveat of requiring the RAM to remain powered as long as data must remain accessible to the node.

While these are valid options, the concurrent use of RAM and a single flash device might be acceptable. While data enters the ESP32-C3's RAM, the flash device can recover from its write operations. With an average flash write speed that is up to 256 kBps slower than the sensor data-generating speed, a perfectly coordinated RAM/flash shared capture would run out of RAM after 0.75 seconds of capture in the worst case, and in its average case, the RAM would never run out before the flash IC (assuming 50 % RAM allocation to program data). An average write time of 0.833 ms is the threshold for the MachMoS device requiring some of the data to accumulate in RAM. If this effective transfer speed is assumed to be normally distributed, with the extremes at three standard deviations away (the flash page write program effective transfer speed is of 320.0 ± 268.8 kBps, $\sigma = 89.6$ kBps), then 307.2 kBps includes 56 % of devices. The minimum write speed before the RAM overflows can also be obtained simply by assuming 50 % RAM allocation to program data and a 13.33 s maximum capture length (4 MB flash size). For the free RAM (384 kB) to fill in 13 seconds, the flash write speed must lag behind the capture speed by at most 29.5 kBps, that is, 277.7 kBps, or 68.2 % of devices. If the page program time is in fact measured differently, where the typical and max times are a measurement of each page write operation, it can be assumed that the device will never suffer from running out of RAM before it runs out of flash memory, as the number of page write operations per second of capture (1200 operations) is large enough to consider a normal distribution of runtimes.

To illustrate that the processor duty for writing to flash is insignificant compared to the sensor data read task, a simulation of the capturing cycle can be made. In the cycle, the accelerometer and microphone capture 256 bytes of data (256 samples). Once captured, the ESP32-C3 takes that data and writes a full page to the flash. The write operation takes 25.6 µs, during which the ESP32-C3 cannot read from the accelerometer. During this time, the accelerometer's 512 B buffer has time to fill with only a single sample of data, which the ESP32-C3 can readily catch up on after the SPI channel is freed up (at an SPI frequency of 10 MHz, the accelerometer read time is 8.14 times faster than the accelerometer data generation speed).

Appendix C Electrical

C.1 Restricted Frequency Bands

C.1.1 United States

The following bands are restricted-use bands in the United States per 47 CFR 15.209 [121]:

- 54-72 MHz,
- 76-88 MHz,
- 174-216 MHz, and
- 470-806 MHz

with some exceptions to usage under:

- 47 CFR 15.231 for periodic operation in the bands
  - 40.66-40.70 MHz and
  - above 70 MHz
- 47 CFR 15.241 for biomedical telemetry devices.

C.1.2 Canada

The bands listed in Table C.1 are resctricted in Canada per the RSS general specifications [124].

Table C.1 – Restricted-Use Frequency Bands in Canada

> 30 MHz	400 - 3000 MHz	3 - 10 GHz	> 10 GHz
37.50 - 38.25	399.9 - 410	3.260 - 3.267	10.6 - 12.7
73.0 - 74.6	608 - 614	3.332 - 3.339	13.25 - 13.4
74.8 - 75.2	960 - 1427	3.3458 - 3.3580	14.47 - 14.5
108 - 138	1435 - 1626.5	3.50 - 4.40	15.35 - 16.2
149.90 - 150.05	1645.5 - 1646.5	4.50 - 5.15	17.7 - 21.4
156.52475 - 156.52525	1660.0 - 1710.0	5.35 - 5.46	22.01 - 23.12
156.7 - 156.9	1718.8 - 1722.2	7.25 - 7.75	23.6 - 24.0
162.0125 - 167.1700	2200 - 2300	8.025 - 8.500	31.2 - 31.8
167.72 - 173.2	2310 - 2390	9.0 - 9.2	36.43 - 36.5
240 - 285	2483.5 - 2500.0	9.3 - 9.5	Above 38.6
322.0 - 335.4	2655 - 2900		

C.2 TPS61022 Light Load Efficiency

Since the TPS61022 load efficiency at very light loads (down to 1 µA) is not given, the sleep task efficiency of the MachMoS device cannot be determined. Due to this, it is necessary to model the behaviour of the device at light loads so that light load efficiency can be extrapolated. The efficiency plot for a 5 V output in Figure C.1 is from the TPS61022 datasheet and will be used for this analysis.

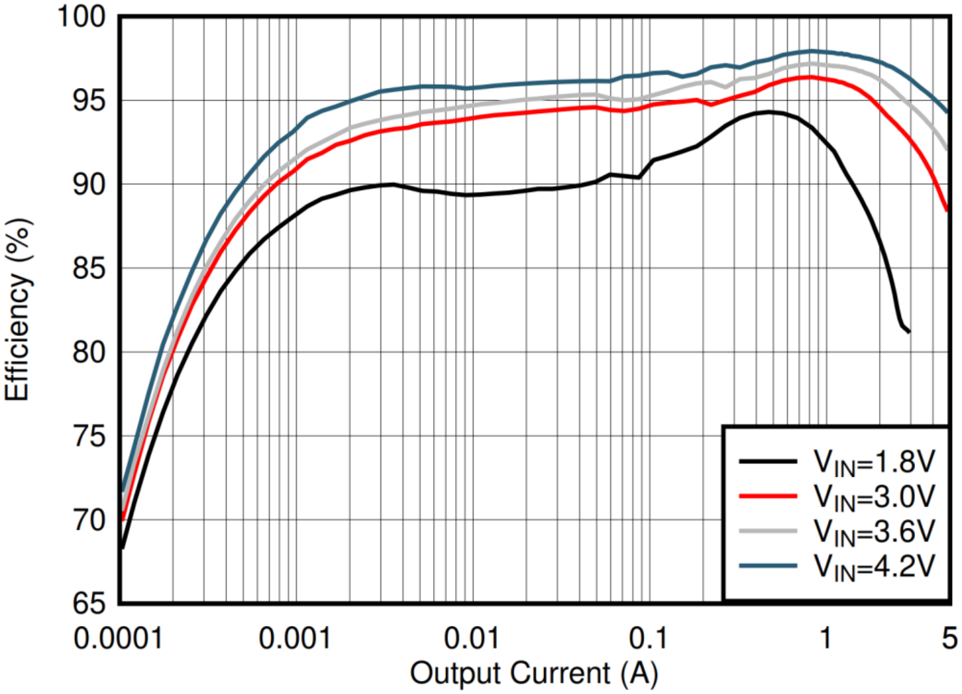


Figure C.1 - TPS61022 Rated Efficiency when Using Auto PFM Mode [185]

In light load applications, the TPS61022 enters PFM mode, in which the output capacitor is intermittently replenished in short bursts. The waveform is shown in Figure C.2, where this effect is observable. In the first 10 µs of the figure, the output capacitors are slowly discharging, until a threshold is reached. In the next 5 µs, the regulator lets current into the inductor in the circuit, and the output capacitor is charged to another threshold.

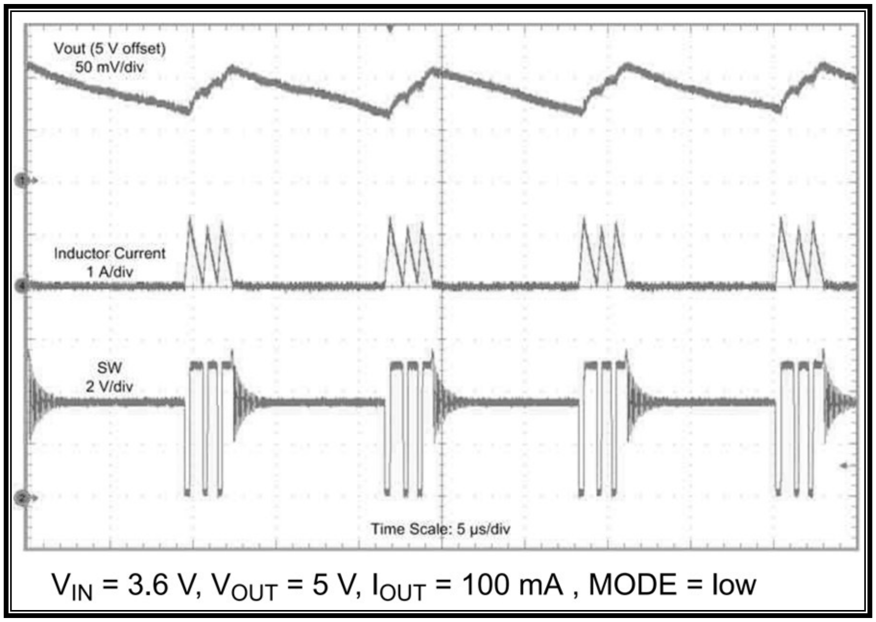


Figure C.2 - Auto PFM Mode Outputs [185]

In typical and heavy load regulator operation, losses in the regulator are related to conduction losses, diode conduction losses, switching losses at the MOSFET in the regulator, and inductor losses [206]. At light loads, however, the effect of these losses is severely attenuated:

- conduction losses scale with the square of the load,
- diode conduction losses are related to the voltage drop at the body diode of the regulator, which scales with the load,
- switching losses as well as the inductor losses scale with the switching frequency of the regulator. In PFM mode, the average frequency is a small fraction of the switching frequency.

At light loads, while the PFM mode is active, the dominant losses are, therefore, related to the energy losses at each burst, E_b. The efficiency η of the regulator can be computed using the simple ratio of power entering the system, P_{in}, to the power leaving the system, P_{out}. Due to energy conservation, $P_{\text{in}} - P_{\text{loss}} = P_{\text{out}}$. From these relations, the power loss can be computed:

$$\eta = \frac{P_{\text{out}}}{P_{\text{in}}} \rightarrow \eta = \frac{P_{\text{out}}}{P_{\text{out}} + P_{\text{loss}}} \rightarrow P_{\text{loss}} = P_{\text{out}}(\eta^{-1} - 1) \tag{26}$$

The power loss is also given by the sum of all the losses in the model:

$$P_{\text{loss}} = f_b E_b + V_{\text{in}} I_q \tag{27}$$

where I_q is the quiescent current and f_b the burst frequency, which scales with the load current. The power output is also given by the product of the output voltage V_o and load current I_o:

$$P_{\text{out}} = V_o I_o \tag{28}$$

Combining equations (26), (27) and (28) gives:

$$V_o I_o(\eta^{-1} - 1) = f_b E_b + V_{\text{in}} I_q \rightarrow E_b f_b = V_o I_o(\eta^{-1} - 1) - V_{\text{in}} I_q \tag{29}$$

Using datasheet efficiency, it is possible to fit $E_b f_b$ against the rated efficiency figures at loads from 0.1 mA to 1 mA. This fit is shown in Figure C.3.

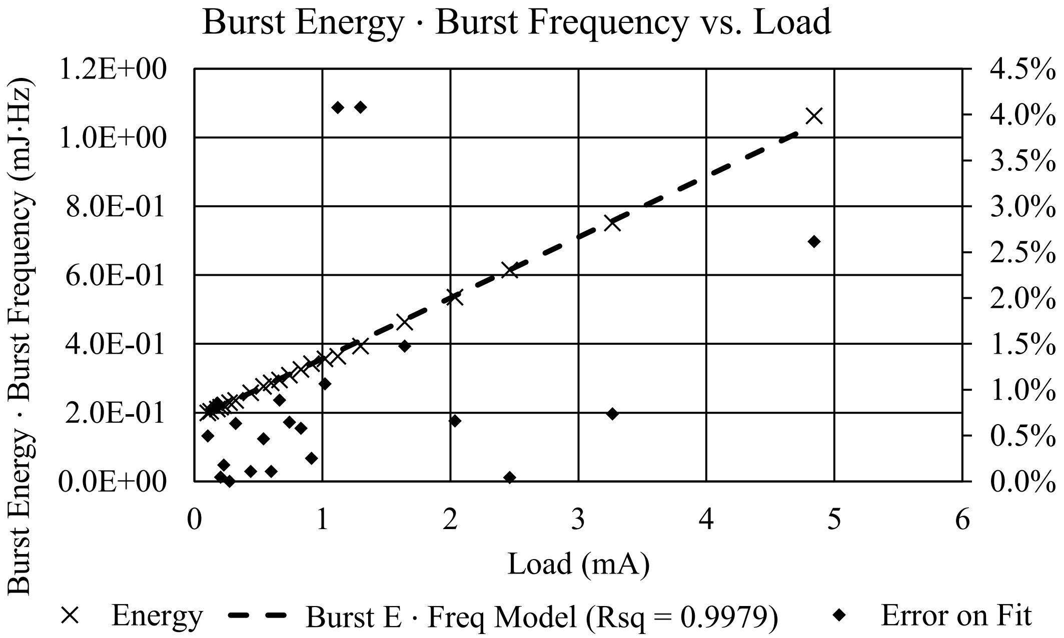


Figure C.3 - $E_b f_b$ as a Function of Load Current

It can be seen in the figure that most of the error on the modeled $E_b f_b$ is under 1 %, with some outliers in the set. The R^2 value for the fit is 99.8 %. The linear fit, which only considers values up to 1 mA, is also valid for extrapolation up to 5 mA, as shown in the figure. The model in equation (30) now has no unknowns and can be used to extrapolate to low-duty operation.

$$\eta = \frac{V_o I_o}{V_o I_o + (f_b E_b + V_{\text{in}} I_q)} \tag{30}$$

The resultant model is plotted in Figure 4.8. Interestingly, the model can successfully estimate efficiency up to a 4A output load, within ±1.47 % of the rated efficiency.

Appendix D Schematics, Manufacturing Drawings

This appendix contains the schematics for the electronics of the MachMoS node and the manufacturing drawings for the mechanical parts.

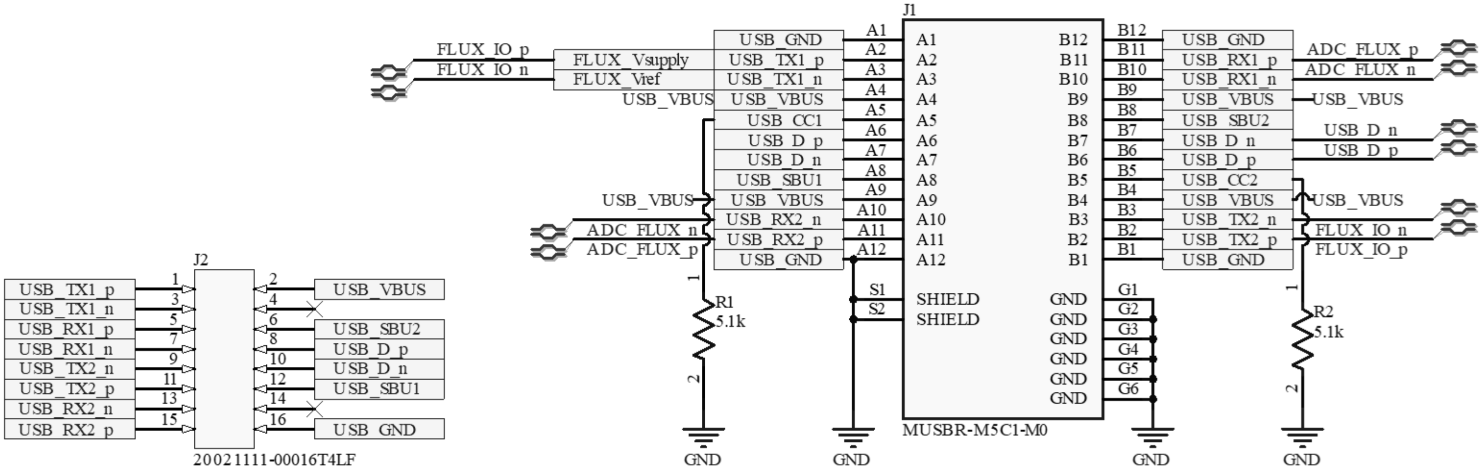


Figure D.1 - MUSBR Adapter

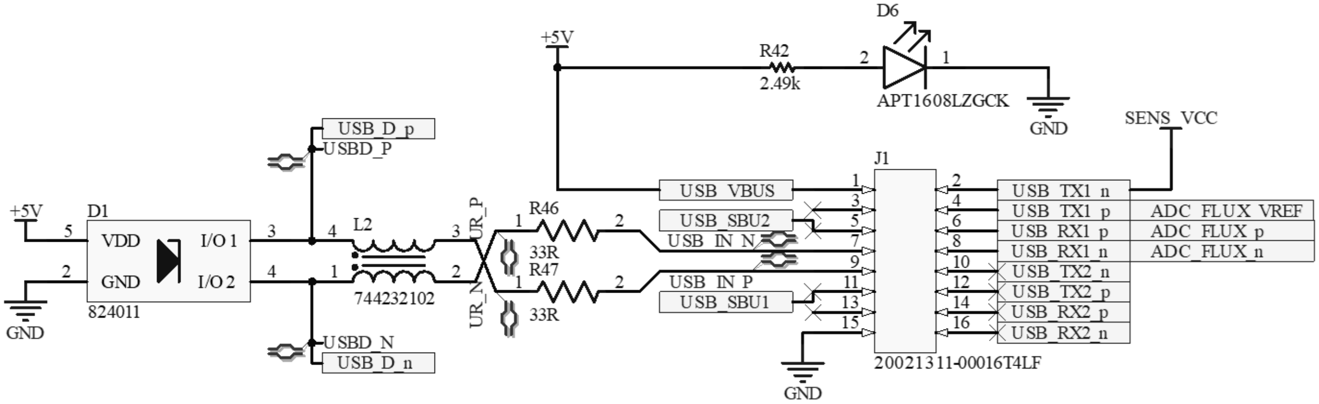
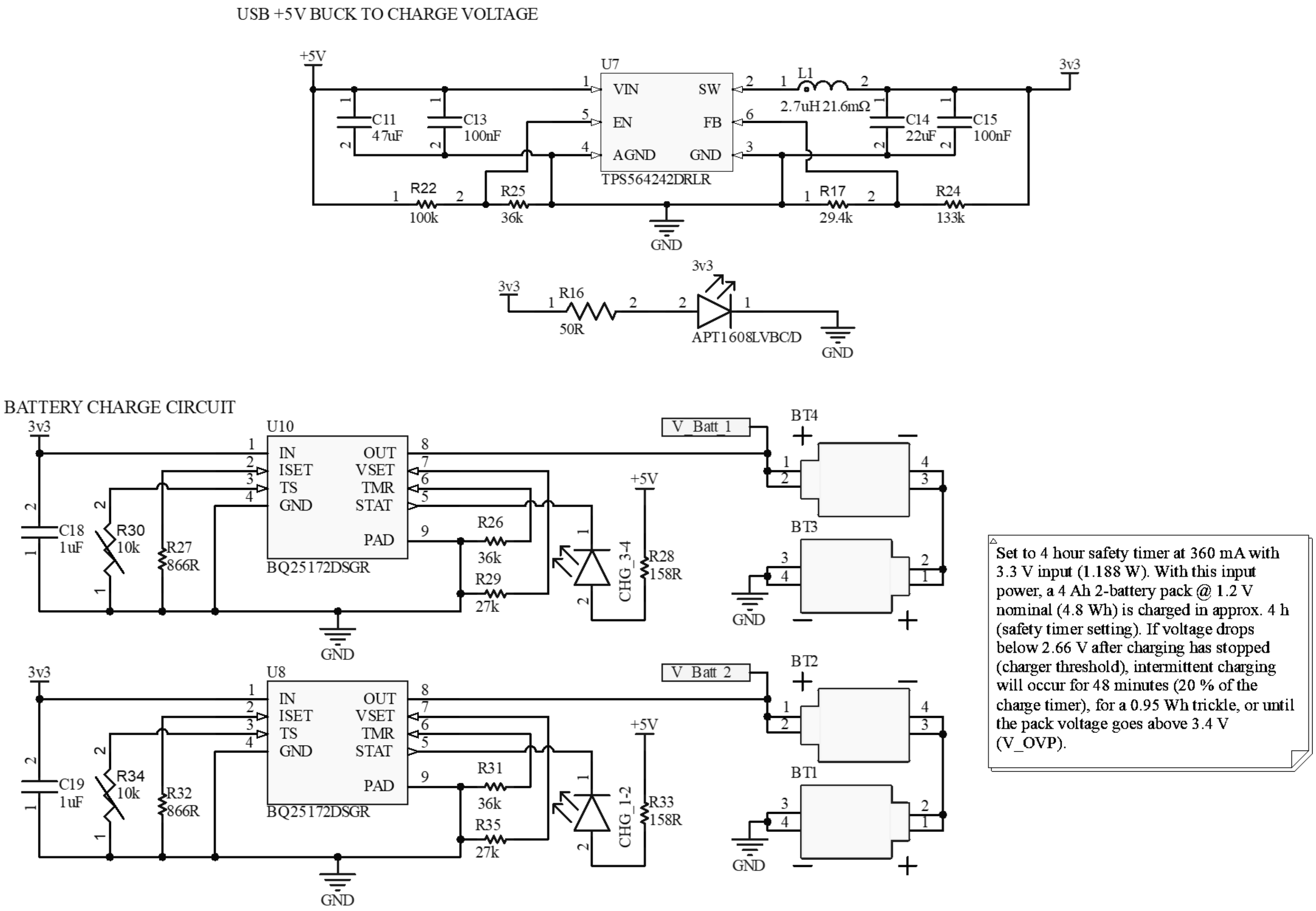


Figure D.2 - Main Board: USB Interface



Set to 4 hour safety timer at 360 mA with 3.3 V input (1.188 W). With this input power, a 4 Ah 2-battery pack @ 1.2 V nominal (4.8 Wh) is charged in approx. 4 h (safety timer setting). If voltage drops below 2.66 V after charging has stopped (charger threshold), intermittent charging will occur for 48 minutes (20 % of the charge timer), for a 0.95 Wh trickle, or until the pack voltage goes above 3.4 V (V_OVP).

Figure D.3 - Main Board: Bus Voltage Regulation and Battery Charging

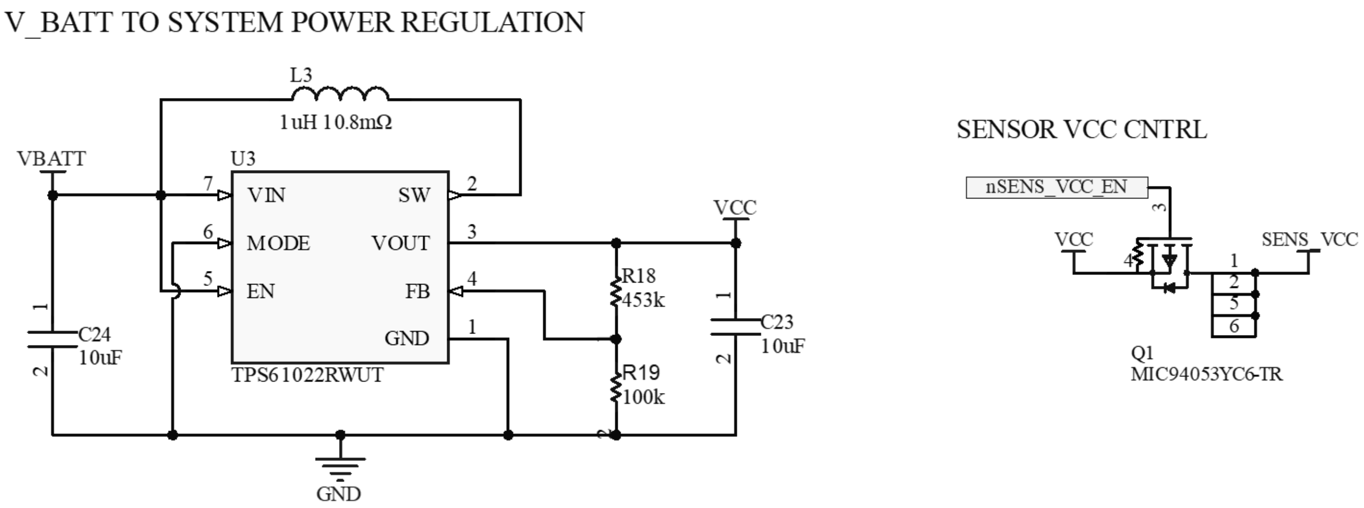


Figure D.4 - Main Board: Battery to System Voltage Regulation, Sensor Power Control

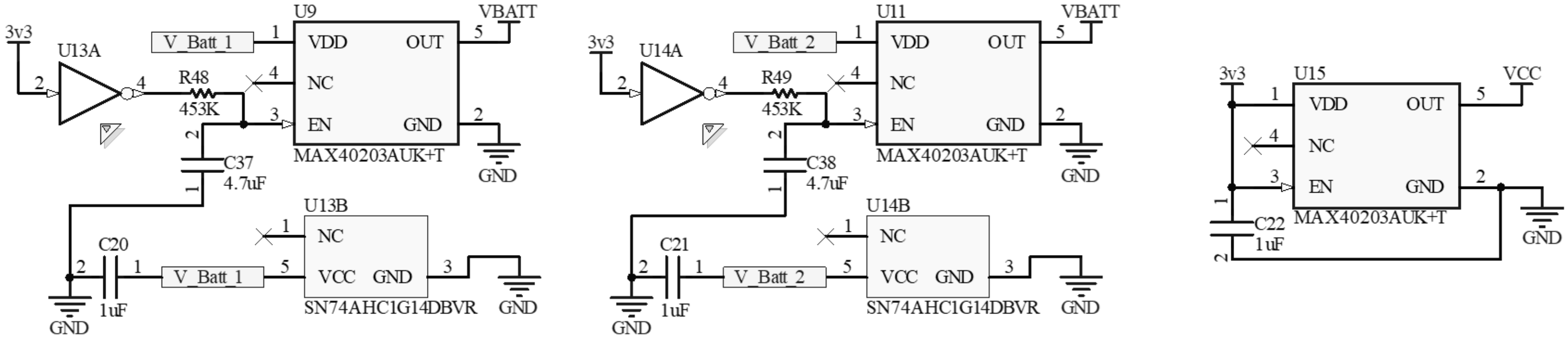


Figure D.5 - Main Board: Ideal Diodes for Load Sharing and USB System Supply

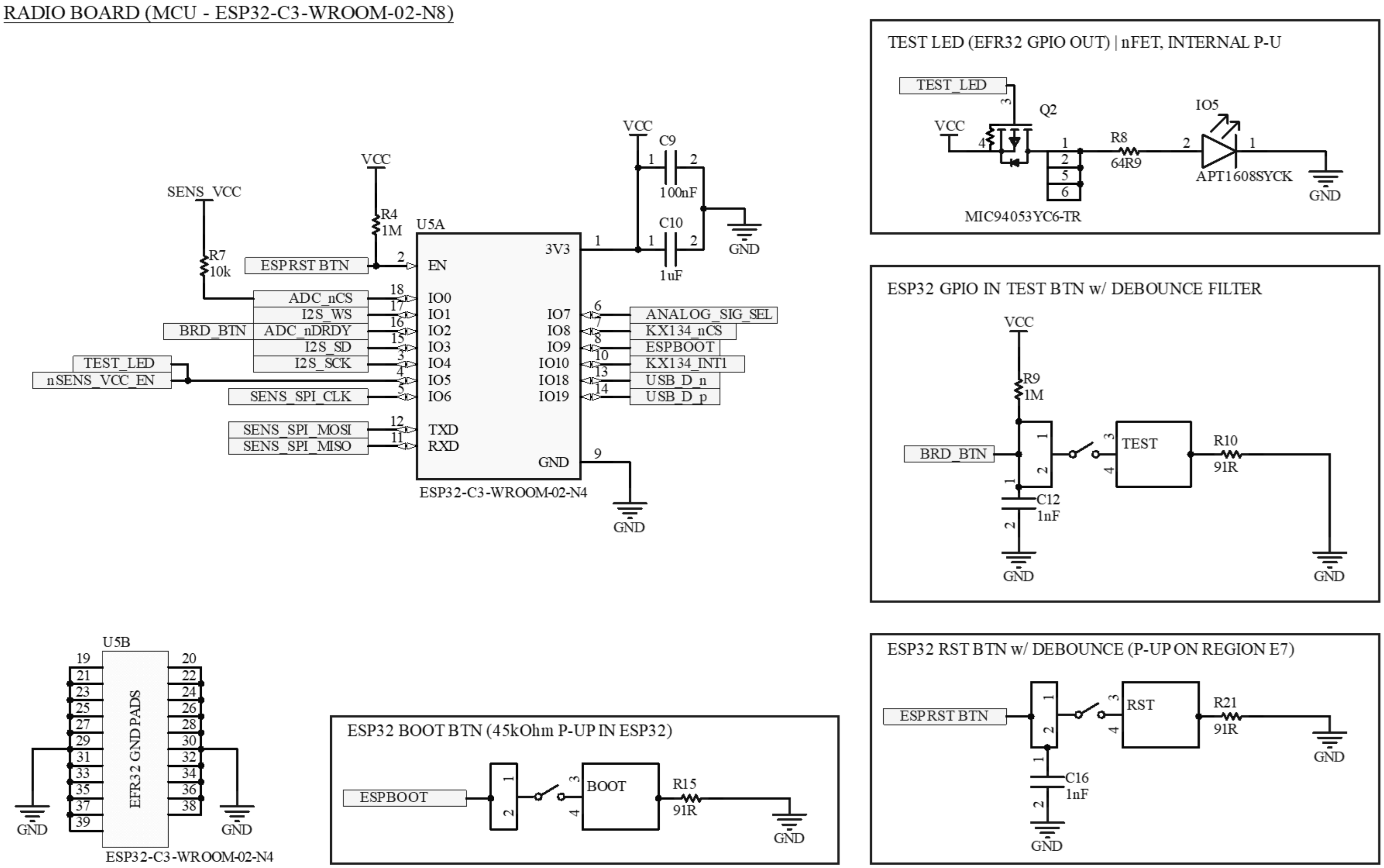


Figure D.6 - Main Board: MCU (Radio Board, Buttons, Status LED

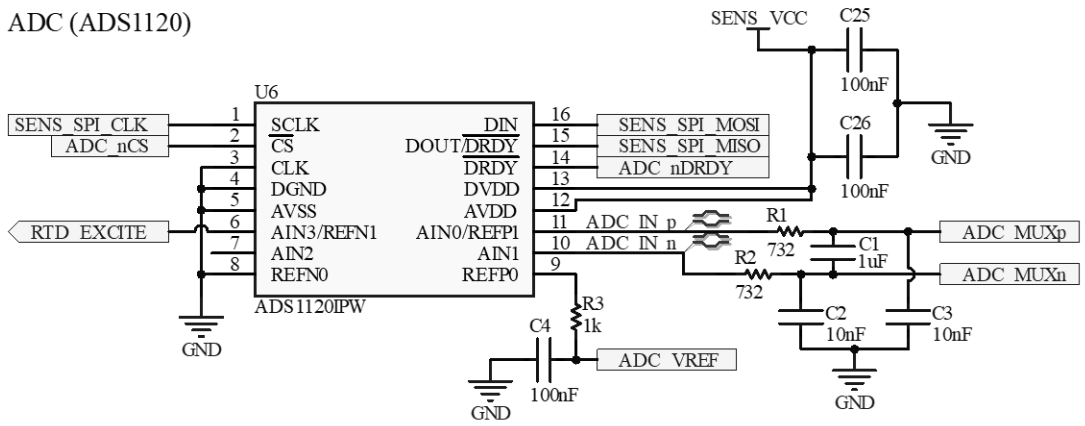


Figure D.7 - Main Board: ADC

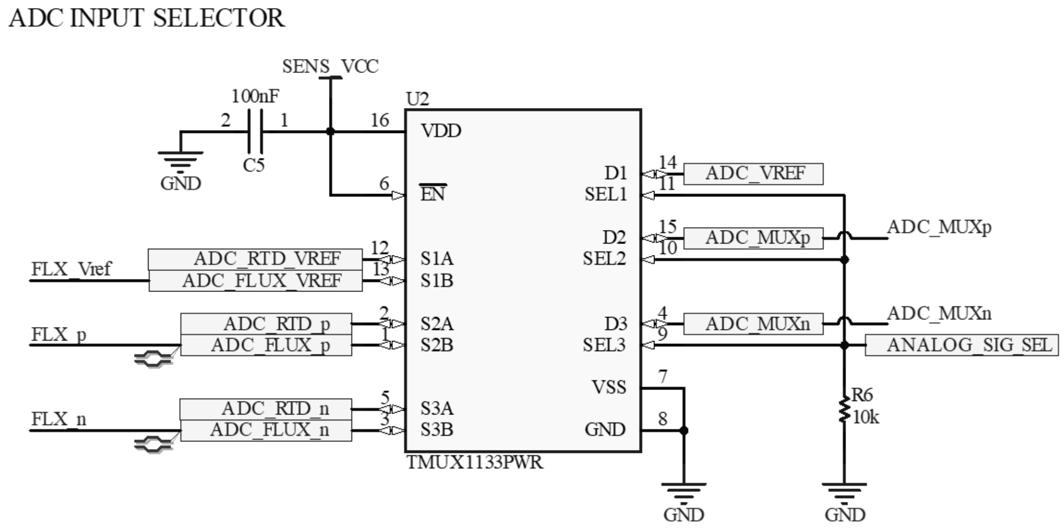


Figure D.8 - Main Board: ADC Input Selection

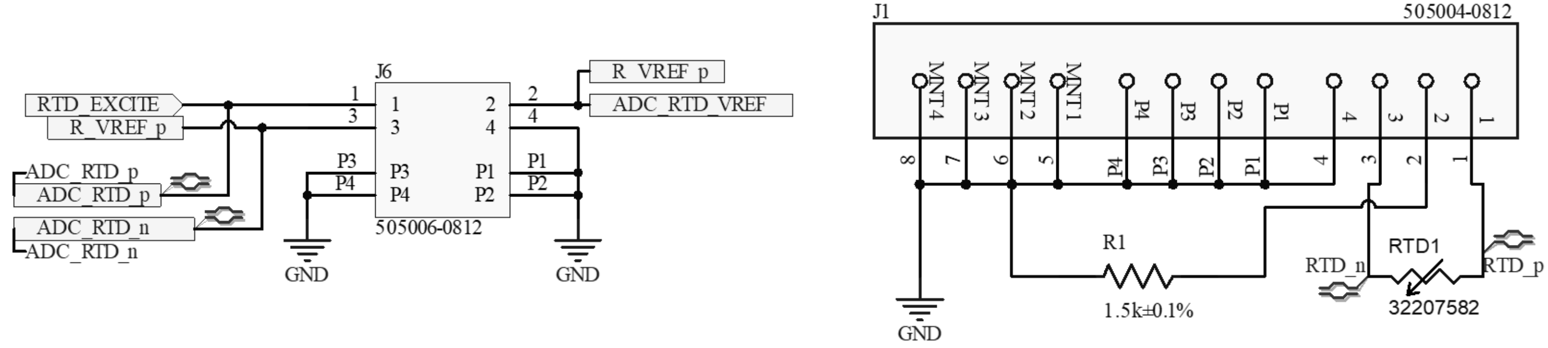


Figure D.9 - Main Board RTD Connector and RTD Breakout

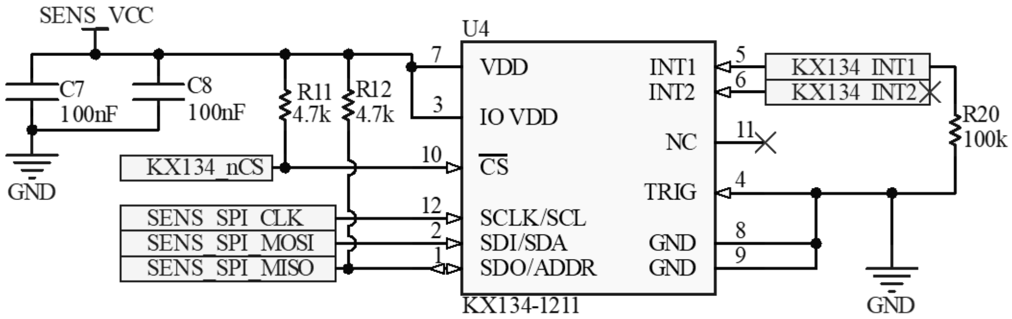


Figure D.10 - Main Board: Accelerometer

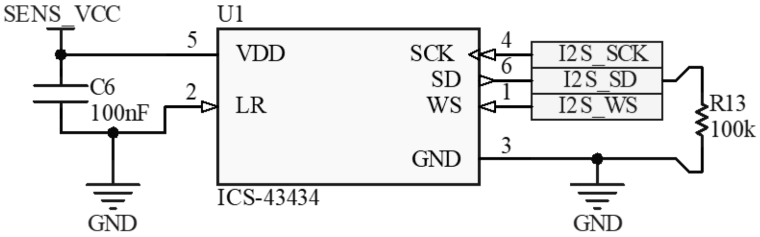
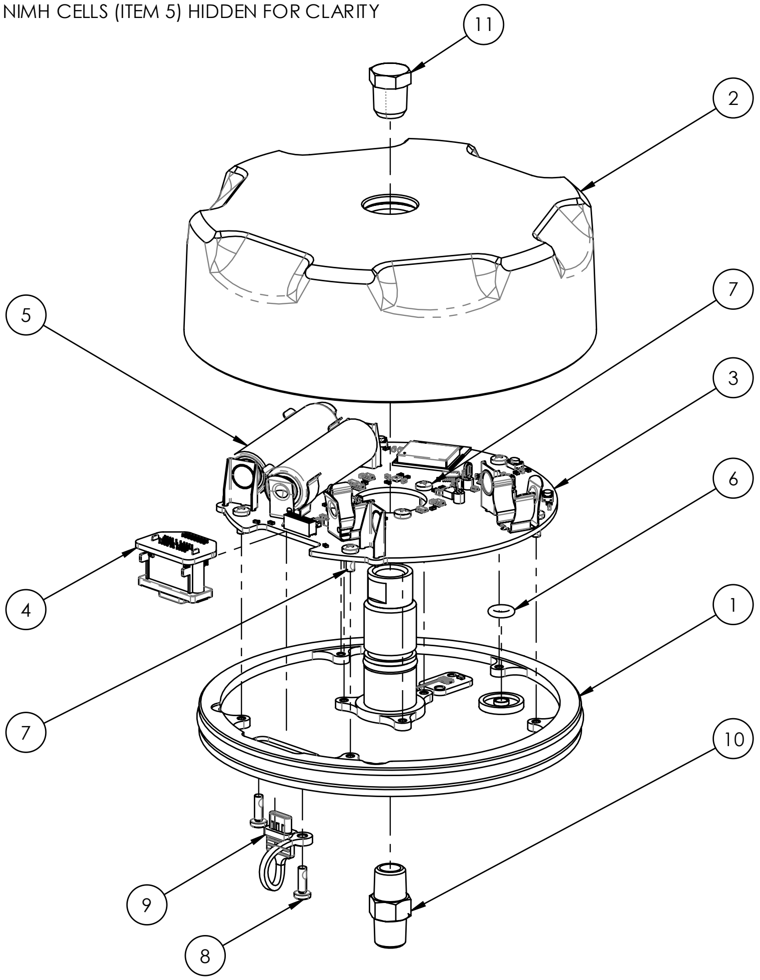


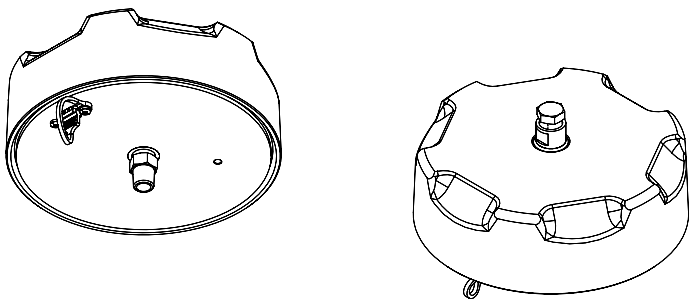
Figure D.11 - Main Board: Microphone

NOTES:
1. 2X NIMH CELLS (ITEM 5) HIDDEN FOR CLARITY

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED



ISOMETRIC EXPLODED VIEW
SCALE: NONE



TOP AND BOTTOM ISOMETRIC VIEWS
SCALE: NONE

ITEM	DRW/PART #	DESCRIPTION	QTY
11	MCM-C P/N: 4757T71	1/8 NPT BRASS PLUG	1
10	MCM-C P/N: 5485K21	1/8 NPT BRASS FITTING	1
9	AMPHENOL P/N: MUSBR2M5C004BP	USB-C JACK DUST COVER	1
8	MCM-C P/N: 95911A168	M2.5X0.45 X 8 mm NYLOK PAN HEAD	2
7	MCM-C P/N: 95911A164	M2.5X0.45 X 6mm NYLOK PAN HEAD	8
6	MCM-C P/N: 9452K16	DASH 008 O-RING, BUNA-N	1
5	PANASONIC P/N: BK-3MCCA	AA NIMH CELL	4
4	DDL-PCB-002	MUSBR ADAPTER	1
3	DDL-PCB-001	MACHMOS MAIN BOARD	1
2	DDL-PRT-002	COVER	1
1	DDL-PRT-001	BASE	1

BILL OF MATERIALS

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN **INCHES**
TOLERANCES:
FRACTIONAL± 1/32"
ANGULAR: MACH± .1°
TWO PLACE DECIMAL± .010"
THREE PLACE DECIMAL± .005"

BREAK ALL SHARP EDGES .01"-.02"

INTERPRET THIS DRAWING IN ACCORDANCE WITH: ASME Y14.5-2018

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	S.TREMBLAY	11/12/2025
CHECKED		
MFG APPR.		
Q.A.		

MATERIAL

MATL. SPEC.

SURFACE TEXTURE $\sqrt{63}$

HEAT TREAT

FINISH

DUMOND DESIGN LAB

TITLE

GENERAL ASSY

MACHMOS NODE

SIZE **B**

DWG. NO. DDL-ASM-001

REV -

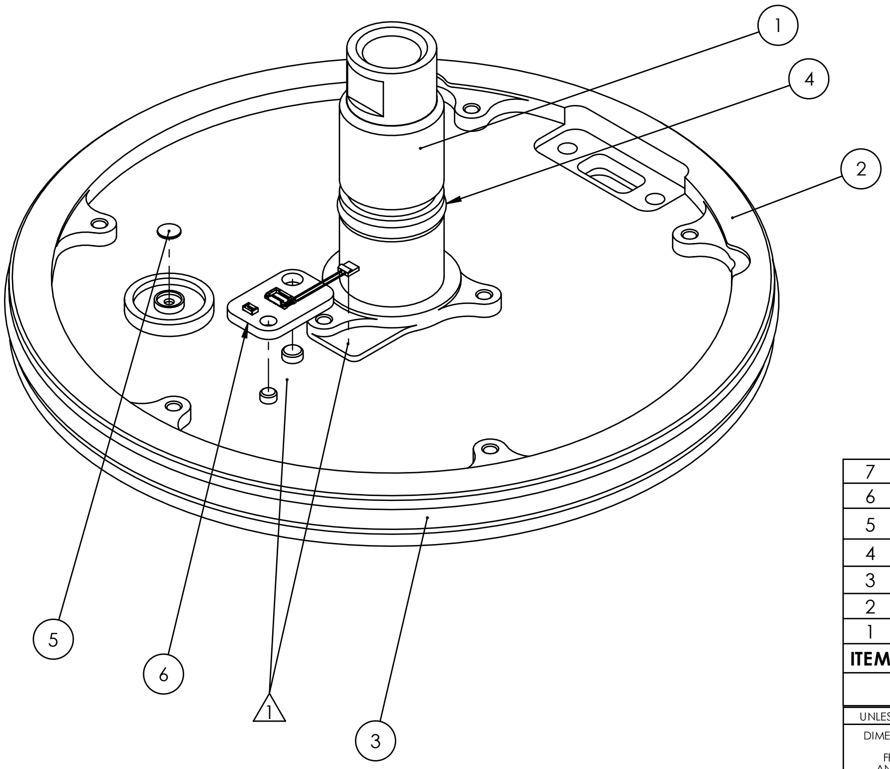
SCALE: 1:2 WEIGHT: 1.13[lb] SHEET 1 OF 1

REVISIONS

REV.	DESCRIPTION	DATE	APPROVED

NOTES:

△1 BOND ITEMS 6 AND 2 USING ITEM 7 UNDER PCB AND RTD DEVICE, WHERE SHOWN.



7	MG CHEMICAL P/N: 8329TFS	THERMAL EPOXY	1
6	DDL-PCB-003	RTD BREAKOUT	1
5	W.L. GORE & ASSOC., P/N: AVP2341.53.5	Ø3.5mm ACOUSTIC VENT, ADHESIVE	1
4	MCM-C P/N: 9452K58	DASH 014 O-RING, BUNA-N	1
3	MCM-C P/N: 9452K194	DASH 244 O-RING, BUNA-N	1
2	DDL-PRT-001_02	PLATE	1
1	DDL-PRT-001_01	PASSTHROUGH	1
ITEM	**DRW/PART #**	**DESCRIPTION**	**QTY**

BILL OF MATERIALS

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN **INCHES**
TOLERANCES:
FRACTIONAL± 1/32"
ANGULAR: MACH± .1°
TWO PLACE DECIMAL± .010"
THREE PLACE DECIMAL± .005"

BREAK ALL SHARP EDGES .01"-.02"

INTERPRET THIS DRAWING IN ACCORDANCE WITH: ASME Y14.5-2018

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	S.TREMBLAY	11/12/2025
CHECKED		
MFG APPR.		
Q.A.		

MATERIAL

MATL. SPEC.

SURFACE TEXTURE

HEAT TREAT

FINISH

DUMOND DESIGN LAB

TITLE

BASE
GENERAL ASSY
MACHMOS NODE

SIZE **B**

DWG. NO. DDL-PRT-001

REV -

SCALE: NONE

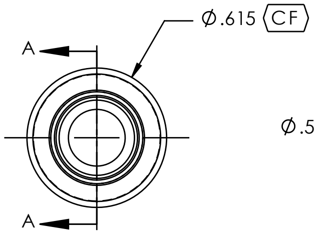
WEIGHT: 0.41[lb]

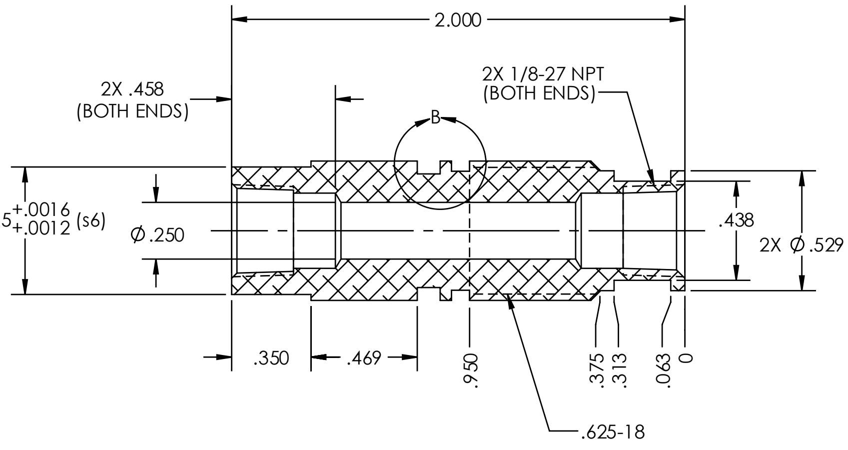
SHEET 1 OF 3

NOTES:

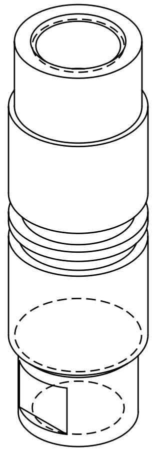
⚠1 THREAD RELIEF OPTIONAL

REVISION TABLE ON SHEET 1

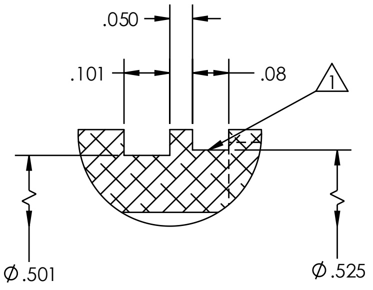




SECTION A-A



ISOMETRIC VIEW
NOT TO SCALE



DETAIL B
SCALE 4 : 1

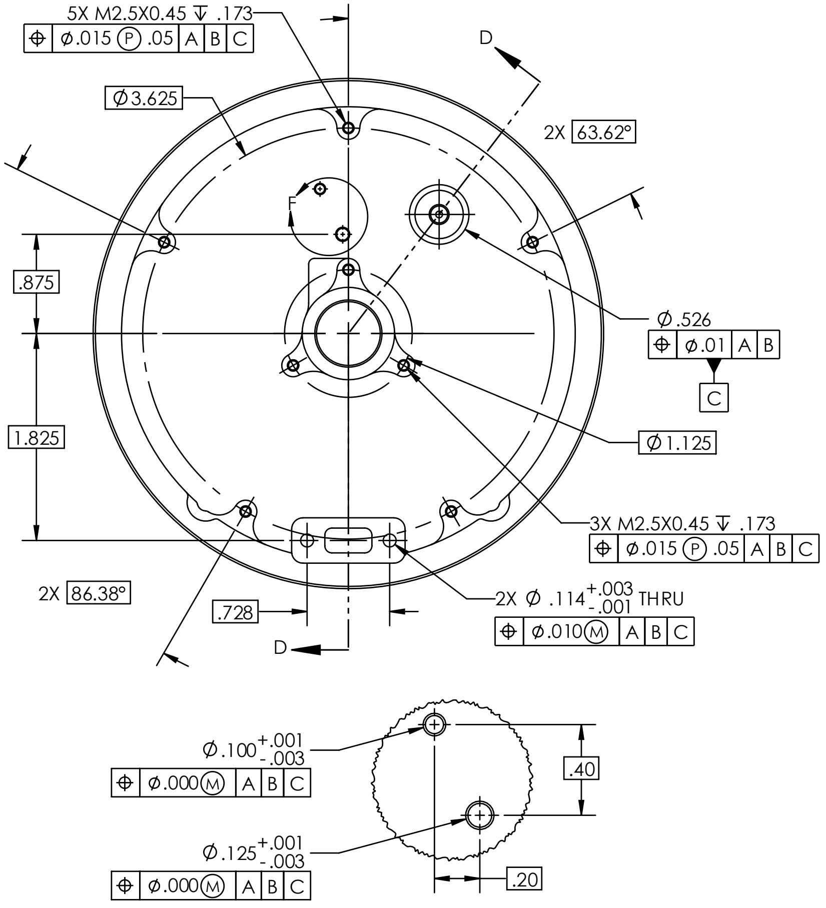
UNLESS OTHERWISE SPECIFIED:		NAME	DATE	**DUMOND DESIGN LAB**
DIMENSIONS ARE IN **INCHES** TOLERANCES: FRACTIONAL ± 1/32" ANGULAR: MACH ± .1° TWO PLACE DECIMAL ± .010" THREE PLACE DECIMAL ± .005" BREAK ALL SHARP EDGES .01"-.02"	DRAWN	S.TREMBLAY	11/12/2025	TITLE **BASE** GENERAL ASSY MACHMOS NODE
	CHECKED			
	MFG APPR.			
	Q.A.			
INTERPRET THIS DRAWING IN ACCORDANCE WITH: ASME Y14.5-2018	MATERIAL			
	MATL. SPEC.	SURFACE TEXTURE 63		SIZE **B** | DWG. NO. DDL-PRT-001_01 | REV -
	HEAT TREAT			
DO NOT SCALE DRAWING	FINISH			SCALE: 2:1 | WEIGHT: N/A | SHEET 2 OF 3

NOTES:

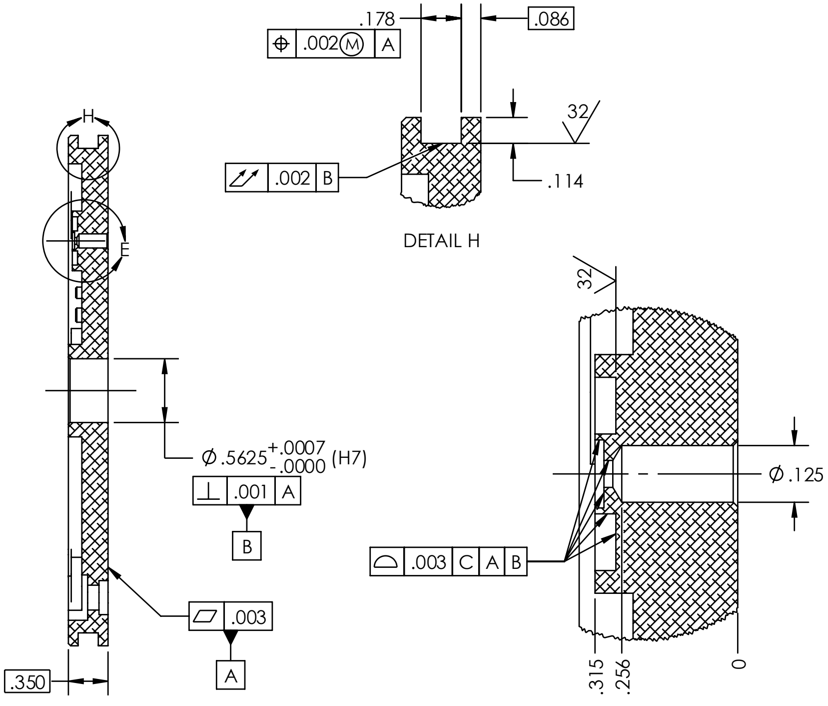
1. CAD MODEL PROVIDED FOR FULL DEFINITION OF THIS PART.
2. ALL DIMENSIONS QUERIED FROM CAD DATA ARE BASIC.
3. ALL SURFACES TO MEET | ⌓ | .01 | A | B | C | UNLESS OTHERWISE SPECIFIED.

DDL-PRT-001_02
PLATE.STEP

REVISION TABLE ON SHEET 1



DETAIL F



DETAIL H

SECTION D-D
SCALE 1 : 1

DETAIL E
SCALE 4 : 1

UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN **INCHES** TOLERANCES: FRACTIONAL± 1/32" ANGULAR: MACH± .1° TWO PLACE DECIMAL± .010" THREE PLACE DECIMAL± .005" BREAK ALL SHARP EDGES .01"-.02"	DRAWN	S.TREMBLAY	11/12/2025
	CHECKED		
	MFG APPR.		
	Q.A.		
INTERPRET THIS DRAWING IN ACCORDANCE WITH: ASME Y14.5-2018	MATERIAL		
	MATL. SPEC.	SURFACE TEXTURE 63√	
	HEAT TREAT		
DO NOT SCALE DRAWING	FINISH		

DUMOND DESIGN LAB

TITLE

BASE
GENERAL ASSY
MACHMOS NODE

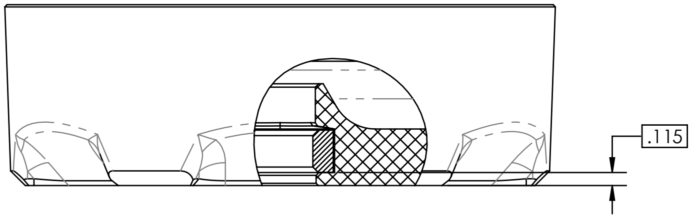
SIZE **B** DWG. NO. DDL-PRT-001_02 REV -

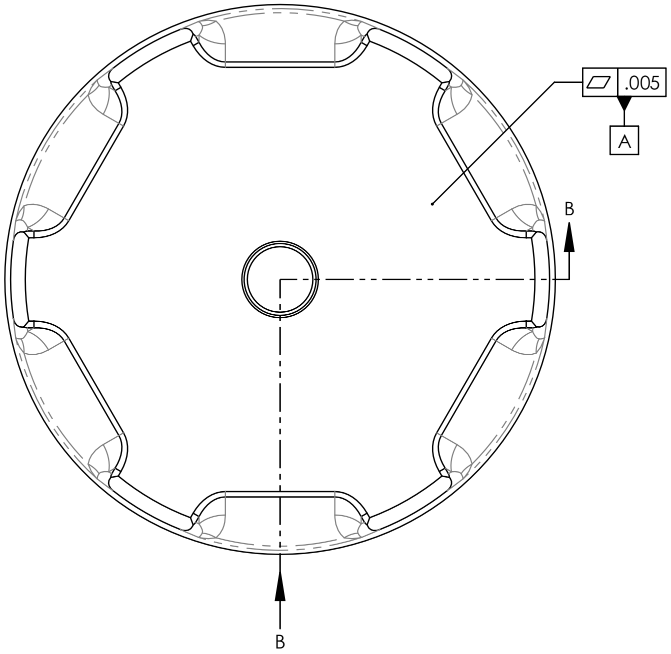
SCALE: NONE WEIGHT: N/A SHEET 3 OF 3

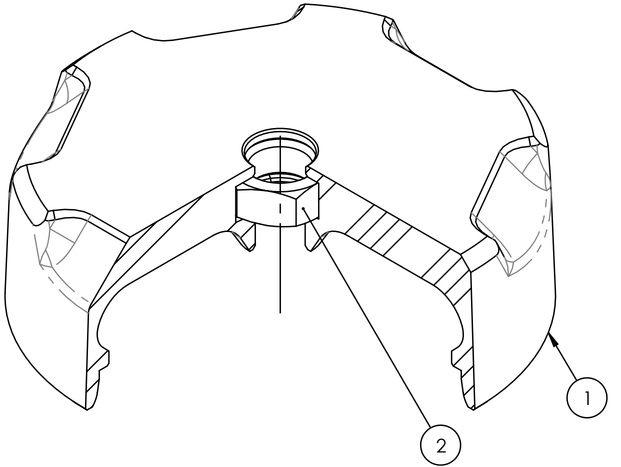
NOTES:
1. CAD MODEL SUPPLIED FOR FULL DEFINITION OF THIS PART.
2. ALL DIMENSIONS QUERIED FROM CAD DATA ARE BASIC.
3. ITEM 2 INSERTED INTO ITEM 1 DURING MANUFACTURING PROCESS.
4. ITEM 1 SURFACES TO BE WITHIN | ⌓ | .005 | A | B |

DDL-PRT-002
COVER.STEP

REVISIONS				
ZONE	REV.	DESCRIPTION	DATE	APPROVED







ISOMETRIC SECTION B-B
SCALE: NONE

ITEM	DRW/PART #	DESCRIPTION	QTY
2	MCM-C P/N: 91078A235	5/8"-18 HEX NUT, THIN, ZN PLATED	1
1	DDL-PRT-002	COVER	1

BILL OF MATERIALS

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN **INCHES**
TOLERANCES:
FRACTIONAL ± 1/32"
ANGULAR: MACH ± .1°
TWO PLACE DECIMAL ± .010"
THREE PLACE DECIMAL ± .005"

BREAK ALL SHARP EDGES .01"-.02"

INTERPRET THIS DRAWING IN ACCORDANCE WITH: ASME Y14.5-2018

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	S.TREMBLAY	11/12/2025
CHECKED		
MFG APPR.		
Q.A.		

MATERIAL
ASA

MATL. SPEC.

SURFACE TEXTURE 63√

HEAT TREAT

FINISH

DUMOND DESIGN LAB

TITLE
COVER
GENERAL ASSY
MACHMOS NODE

SIZE	DWG. NO.	REV
B	DDL-PRT-002	-

SCALE: 1:1 | WEIGHT: 0.56[lb] | SHEET 1 OF 1

Appendix E Electrical Bill of Materials

The detailed bill of materials for electronic components is provided below in Table E.1, Table E.2, and Table E.3. The costs were queried from Digi-Key, Mouser, and LCSC. Taxes are not included. All prices are in CAD.

Table E.1 - Electrical Bill of Materials (DDL-PCB-001: MAIN BOARD)

Manufacturer P/N	Manufacturer	Designator	Category	QTY	Price	Ext.
92	Keystone	BT1, BT2, BT3, BT4	Connector	4	0.54	2.16
595	Keystone	BT1, BT2, BT3, BT4	Connector	4	0.31	1.24
20021311-00016T4LF	Amphenol Communications Solutions	J1	Connector	1	2.44	2.44
505006-0812	Molex	J6	Connector	1	0.52	0.52
1255AY-2R7N=P3	Murata	L1	Inductor, Filter	1	0.57	0.57
744232102	Wurth Elektronik	L2	Inductor, Filter	1	2.08	2.08
XFL4020-102MEC	Coilcraft	L3	Inductor, Filter	1	3.17	3.17
MIC94053YC6-TR	Microchip	Q1, Q2	Integrated Circuits	2	1.19	2.38
ICS-43434	InvenSense	U1	Integrated Circuits	1	4.98	4.98
TMUX1133PWR	Texas Instruments	U2	Integrated Circuits	1	5.37	5.37
TPS61022RWUT	Texas Instruments	U3	Integrated Circuits	1	3.65	3.65
KX134-1211	ROHM	U4	Integrated Circuits	1	18.65	18.65
ESP32-C3-WROOM-02-N4	Espressif Systems	U5	Integrated Circuits	1	5.53	5.53
ADS1120IPW	Texas Instruments	U6	Integrated Circuits	1	12.85	12.85
TPS564242DRLR	Texas Instruments	U7	Integrated Circuits	1	0.85	0.85
BQ25172DSGR	Texas Instruments	U8, U10	Integrated Circuits	2	1.46	2.92
MAX40203AUK+T	Maxim	U9, U11, U15	Integrated Circuits	3	2.66	7.98
SN74AHC1G14DBVR	Texas Instruments	U13, U14	Integrated Circuits	2	0.18	0.37
APT1608LVBC/D	Kingbright	3v3	LED, Button, Test Point	1	0.90	0.90
PTS815SJK250SMTRLFS	C&K Components	BOOT, RST, TEST	LED, Button, Test Point	3	0.27	0.82

Manufacturer P/N	Manufacturer	Designator	Category	QTY	Price	Ext.
HSMA-C190	Avago	CHG_1-2, CHG_3-4	LED, Button, Test Point	2	0.93	1.86
APT1608LZGCK	Kingbright	D6	LED, Button, Test Point	1	0.78	0.78
APT1608SYCK	Kingbright	IO5	LED, Button, Test Point	1	0.31	0.31
5010	Keystone	TP1, TP2, TP4, TP7	LED, Button, Test Point	4	0.49	1.96
5011	Keystone	TP3, TP6	LED, Button, Test Point	2	0.49	0.98
CC0603KRX7R9BB105	Yageo Group	C1, C10, C18, C19, C20, C21, C22	Passive	7	0.17	1.17
GRM188R72A103KA01D	Murata	C2, C3	Passive	2	0.15	0.30
CGA3E2X7R1H104K080AA	TDK	C4, C5, C6, C7, C8, C9, C13, C15, C25, C26	Passive	10	0.07	0.66
GRM31CR61A476KE15L	Murata	C11	Passive	1	0.67	0.67
06035C102KAT2A	Kyocera AVX	C12, C16	Passive	2	0.12	0.24
GRM21BD70J226ME44L	Murata	C14	Passive	1	0.49	0.49
GRM21BR61E106MA73L	Murata	C23, C24	Passive	2	0.25	0.50
GRM188Z71A475ME15D	Murata	C37, C38	Passive	2	0.31	0.62
824011	Wurth Elektronik	D1	Passive	1	1.33	1.33
AF0603FR-07732RL	Yageo Group	R1, R2	Passive	2	0.15	0.30
RC0805FR-071KL	Yageo Group	R3	Passive	1	0.18	0.18
ERJ-3GEYJ105V	Panasonic	R4, R9	Passive	2	0.16	0.32
CRCW060310K0FKEA	Vishay	R6, R7	Passive	2	0.15	0.30
ERJ-3EKF64R9V	Panasonic	R8	Passive	1	0.15	0.15
ERJ-3EKF91R0V	Panasonic	R10, R15, R21	Passive	3	0.15	0.46
ERJ-3EKF4701V	Panasonic	R11, R12	Passive	2	0.16	0.32
RC0603FR-13100KL	Yageo Group	R13, R19, R20, R22	Passive	4	0.17	0.68
CRCW060350R0FKEA	Vishay	R16	Passive	1	0.16	0.16
CRCW060329K4FKEA	Vishay	R17	Passive	1	0.16	0.16

Manufacturer P/N	Manufacturer	Designator	Category	QTY	Price	Ext.
ERJ-3EKF4533V	Panasonic	R18	Passive	1	0.16	0.16
CRCW0603133KFKEA	Vishay	R24	Passive	1	0.16	0.16
ERA3AEB363V	Panasonic	R25, R26, R31	Passive	3	0.16	0.48
RC0603FR-07866RL	Yageo Group	R27, R32	Passive	2	0.16	0.32
RC0603FR-07158RL	Yageo Group	R28, R33	Passive	2	0.16	0.32
ERJ3EKF2702V	Panasonic	R29, R35	Passive	2	0.16	0.32
NTCG163JF103FT1	TDK	R30, R34	Passive	2	0.27	0.54
ERA3AEB2491V	Panasonic	R42	Passive	1	0.16	0.16
ERJ-3EKF33R0V	Panasonic	R46, R47	Passive	2	0.16	0.32
CRCW0603453KFKEA	Vishay	R48, R49	Passive	2	0.16	0.32
Main Board	JLCPCB	N/A	PCB	1	3.36	3.36
					Total:	$ 100.79

Table E.2 - Electrical Bill of Materials (DDL-PCB-002: MUSBR Adapter Breakout)

Manufacturer P/N	Manufacturer	Designator	Category	QTY	Price	Ext.
MUSBR-M5C1-M0	Amphenol ICC / FCI	J1	Connector	1	27.29	27.29
20021111-00016T4LF	Amphenol ICC / FCI	J2	Connector	1	1.80	1.80
CRCW06035K10FKEA	Vishay	R1, R2	Passive	2	0.16	0.32
MUSBR Adapter Breakout	DDL	N/A	PCB	1	3.30	3.30
					Total:	$ 32.71

Table E.3 - Electrical Bill of Materials (DDL:PCB-003: RTD Breakout)

Manufacturer P/N	Manufacturer	Designator	Category	QTY	Price	Ext.
505004-0812	Molex	J1	Connector	1	0.52	0.52
32207582	Heraeus	RTD1	Integrated Circuits	1	5.93	5.93
RP73PF2A1K5BTDF	TE Connectivity	R1	Passive	1	0.46	0.46
RTD Breakout	JLCPCB	N/A	PCB	1	1.54	1.54
					Total:	$ 8.45

Appendix F Vibro-Acoustic Test Results

F.1 Frequency and Amplitude

1 kHz tests are not shown in this appendix as they are already discussed in section 5.2. Accelerometer results are omitted from the 3 kHz, 4 V and 8 V tests as these are already included in section 5.2.

F.1.1 20 Hz, 1 V Input

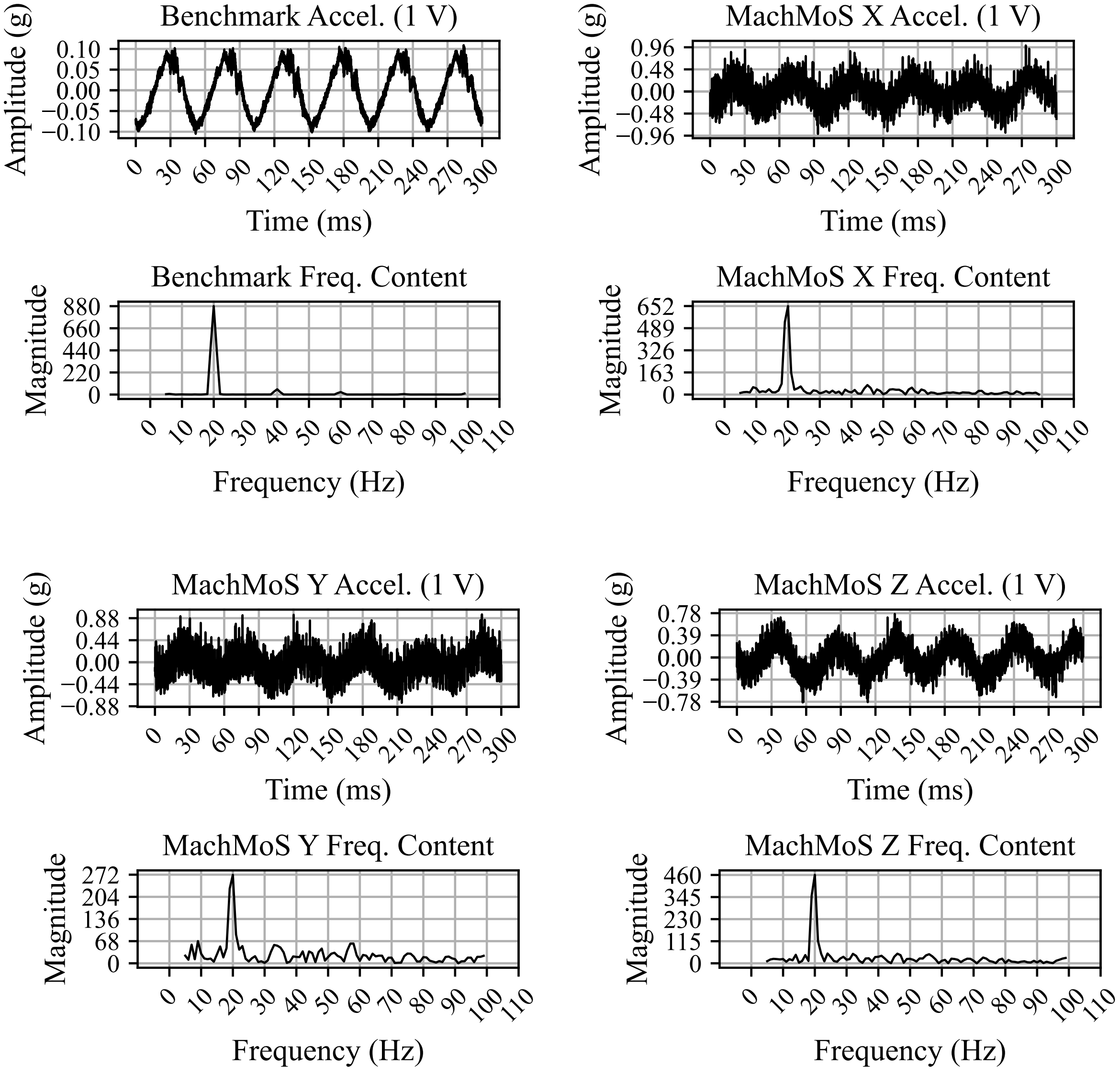


Figure F.1 - Accelerometer Test Results: 20 Hz, 1 V Input

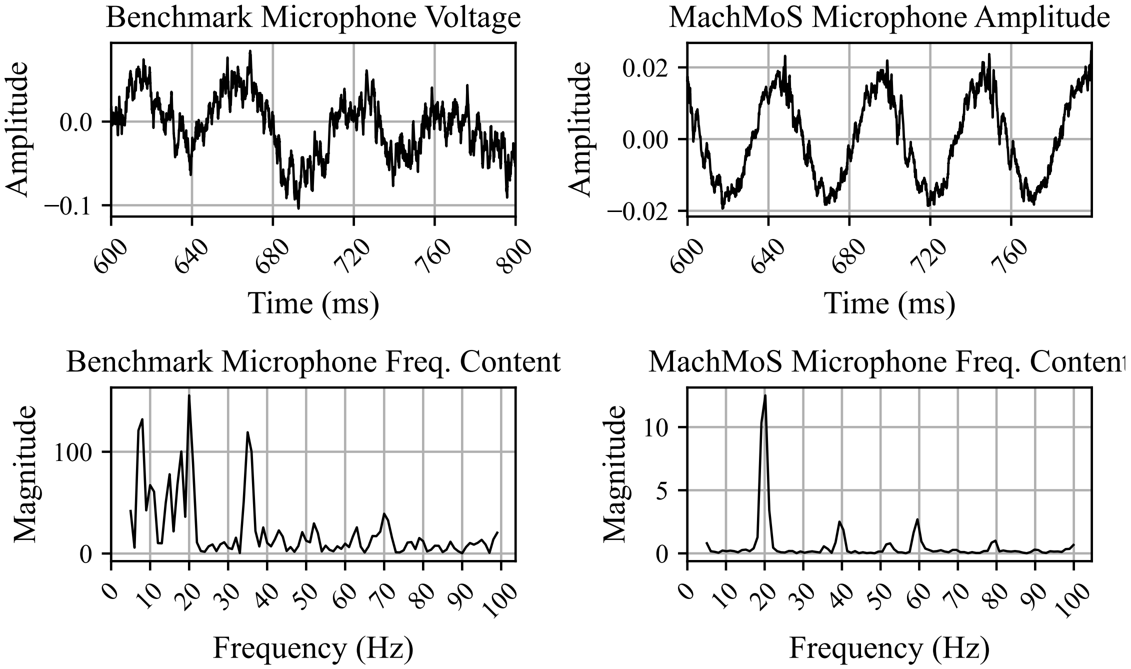
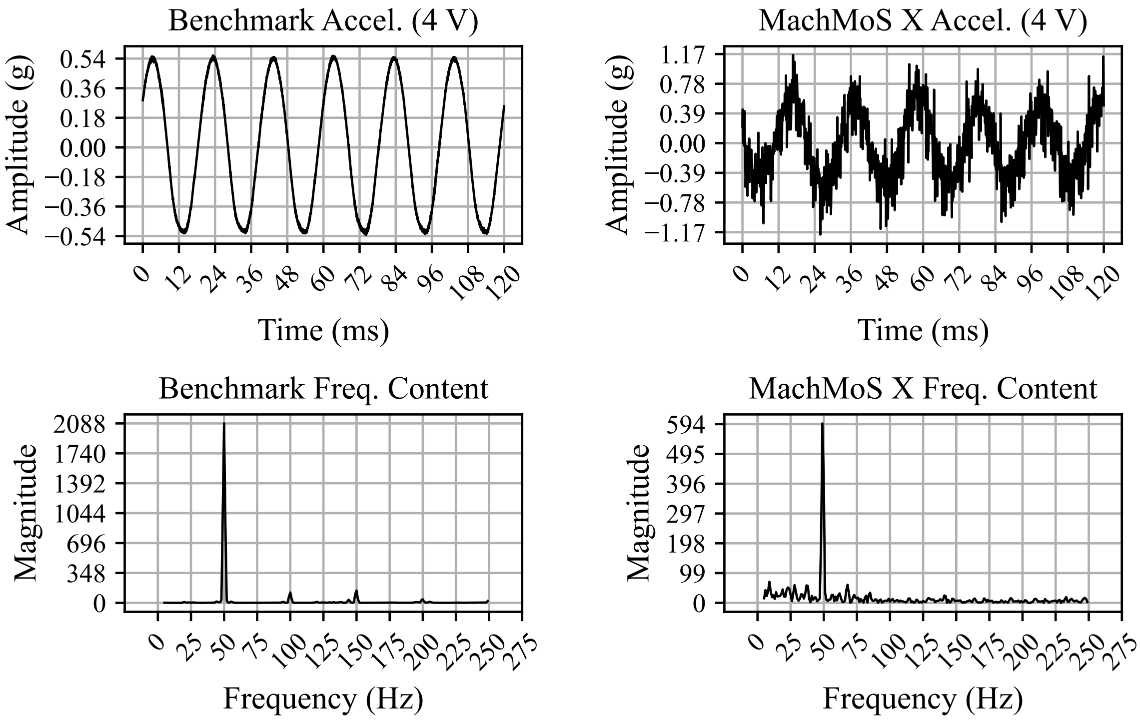


Figure F.2 - Accelerometer Test Results: 20 Hz, 1 V Input

F.1.2 50 Hz, 4 V Input



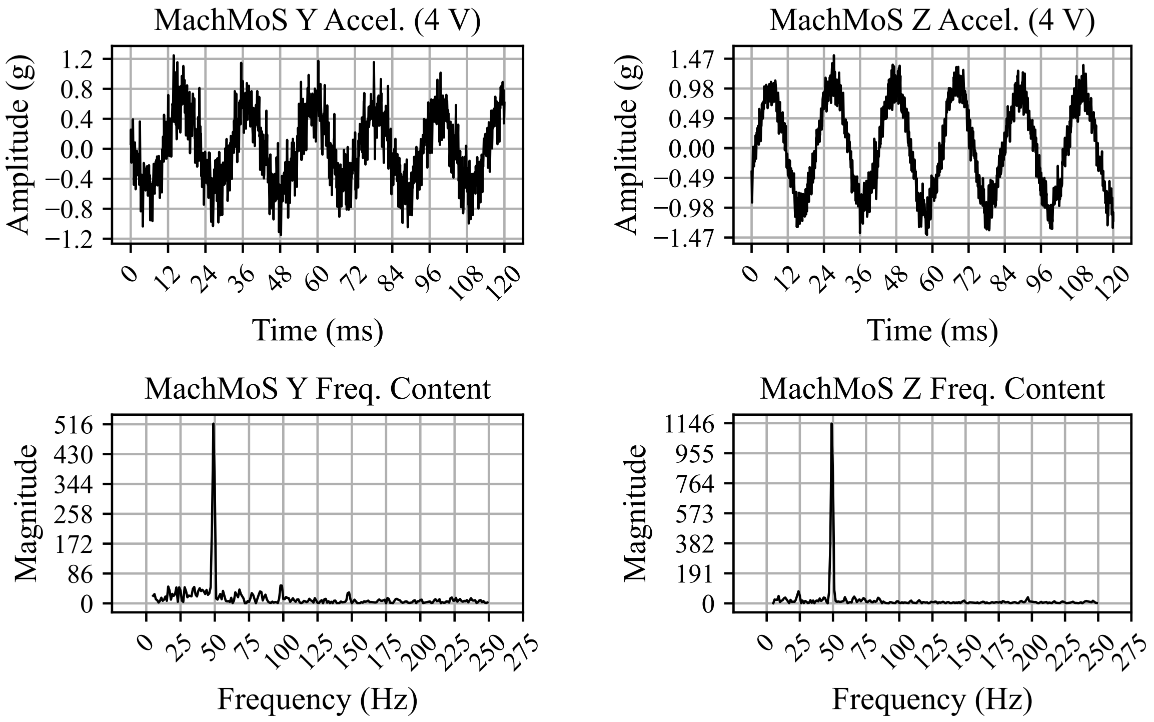


Figure F.3 - Accelerometer Test Results: 50 Hz, 4 V Input

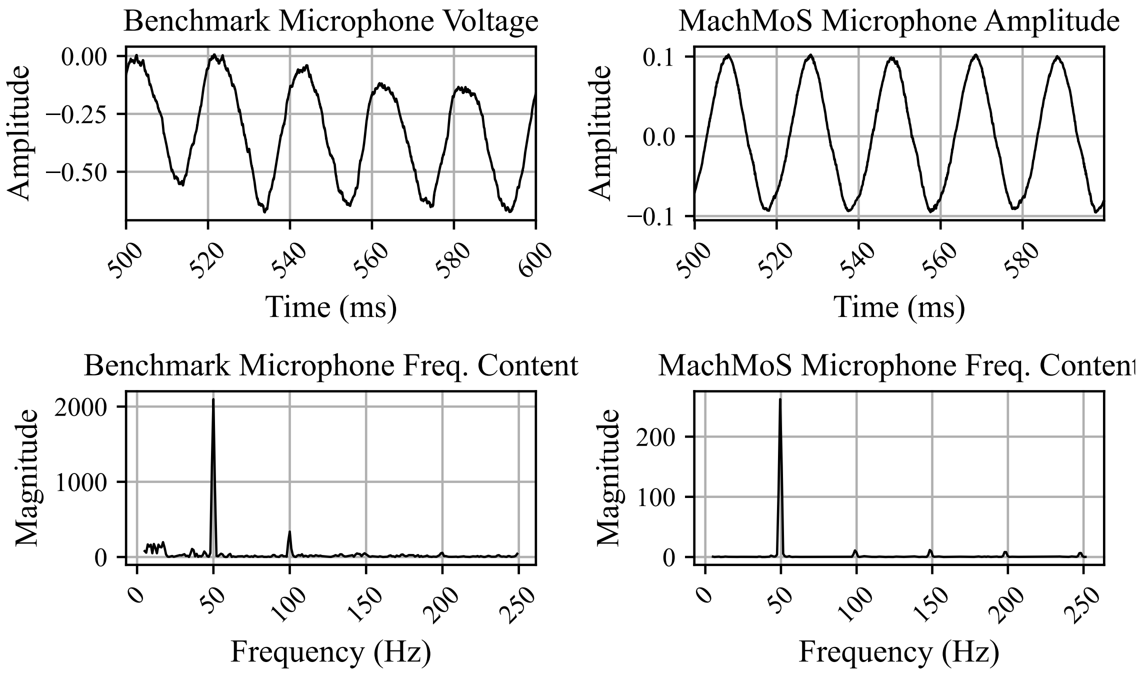


Figure F.4 - Microphone Test Results: 50 Hz, 4 V Input

F.1.3 100 Hz, 4 V Input

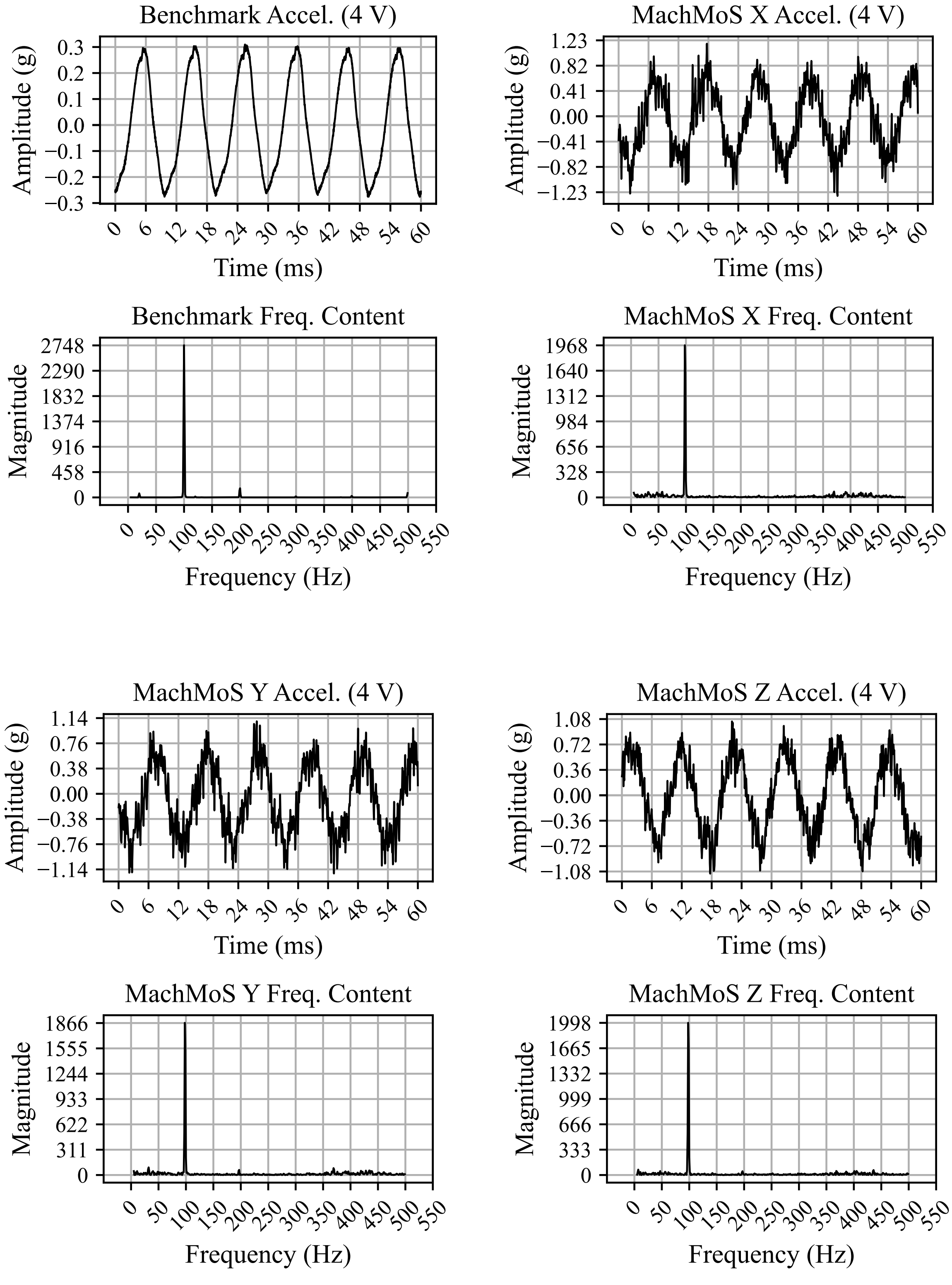


Figure F.5 - Accelerometer Test Results: 100 Hz, 4 V Input

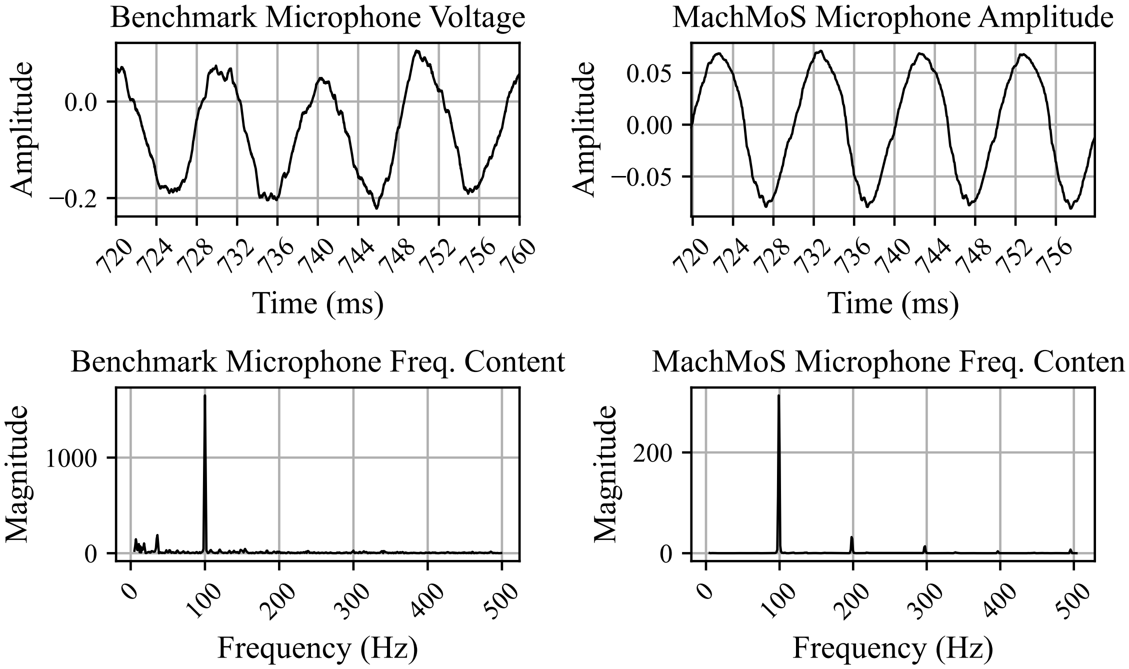
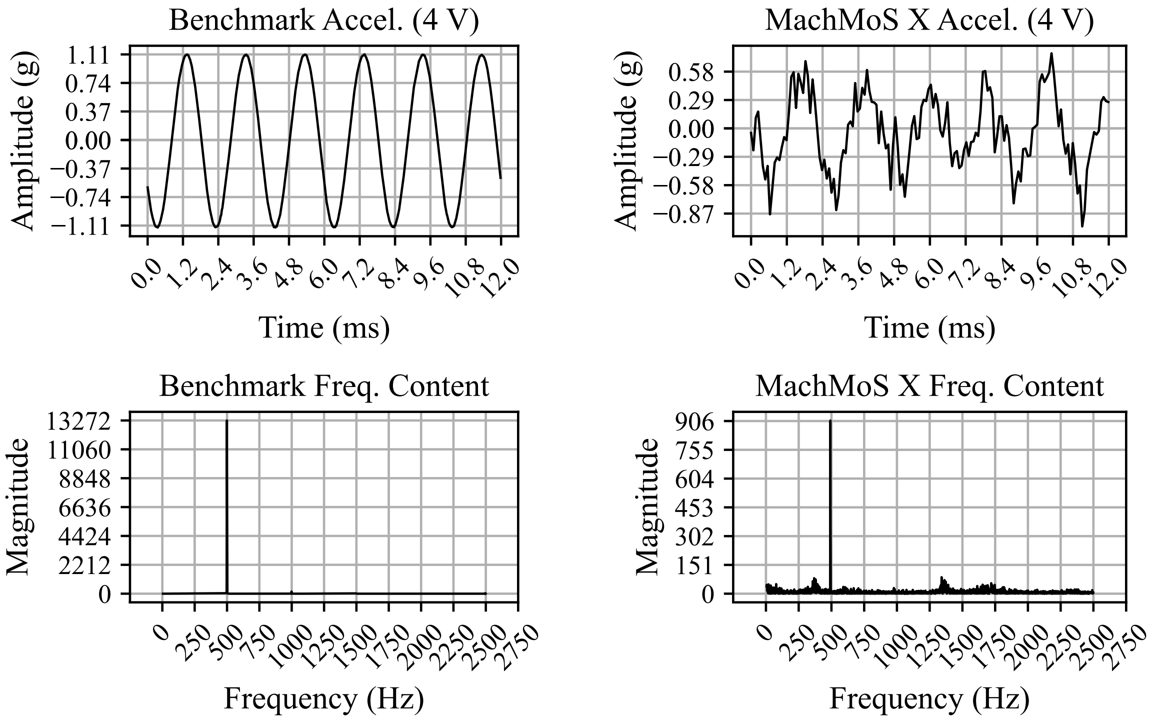


Figure F.6 - Microphone Test Results: 100 Hz, 4 V Input

F.1.4 500 Hz, 4 V Input



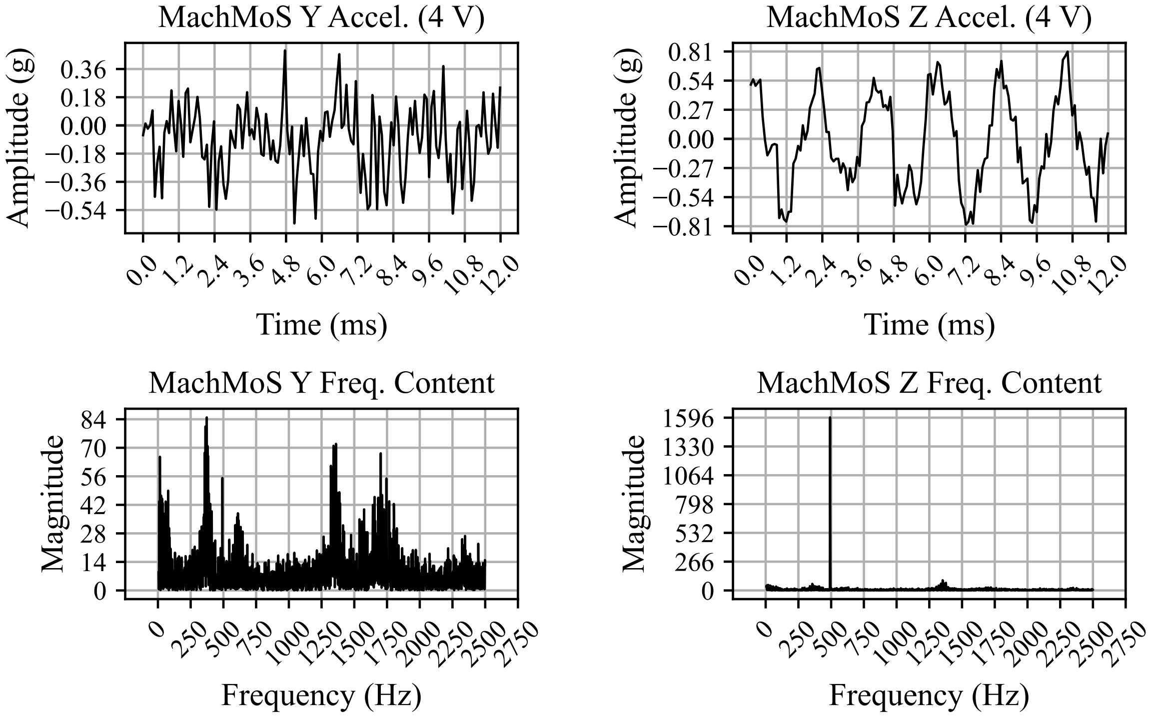


Figure F.7 - Accelerometer Test Results: 500 Hz, 4 V Input

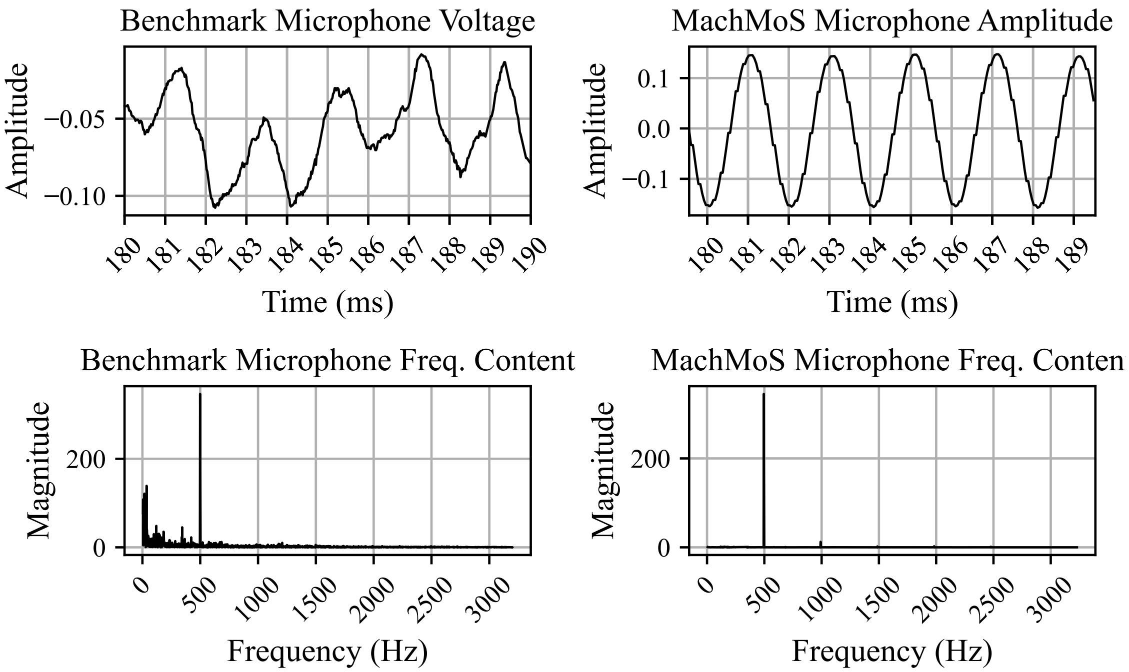


Figure F.8 - Microphone Test Results: 500 Hz, 4 V Input

F.1.5 2 kHz, 4 V Input

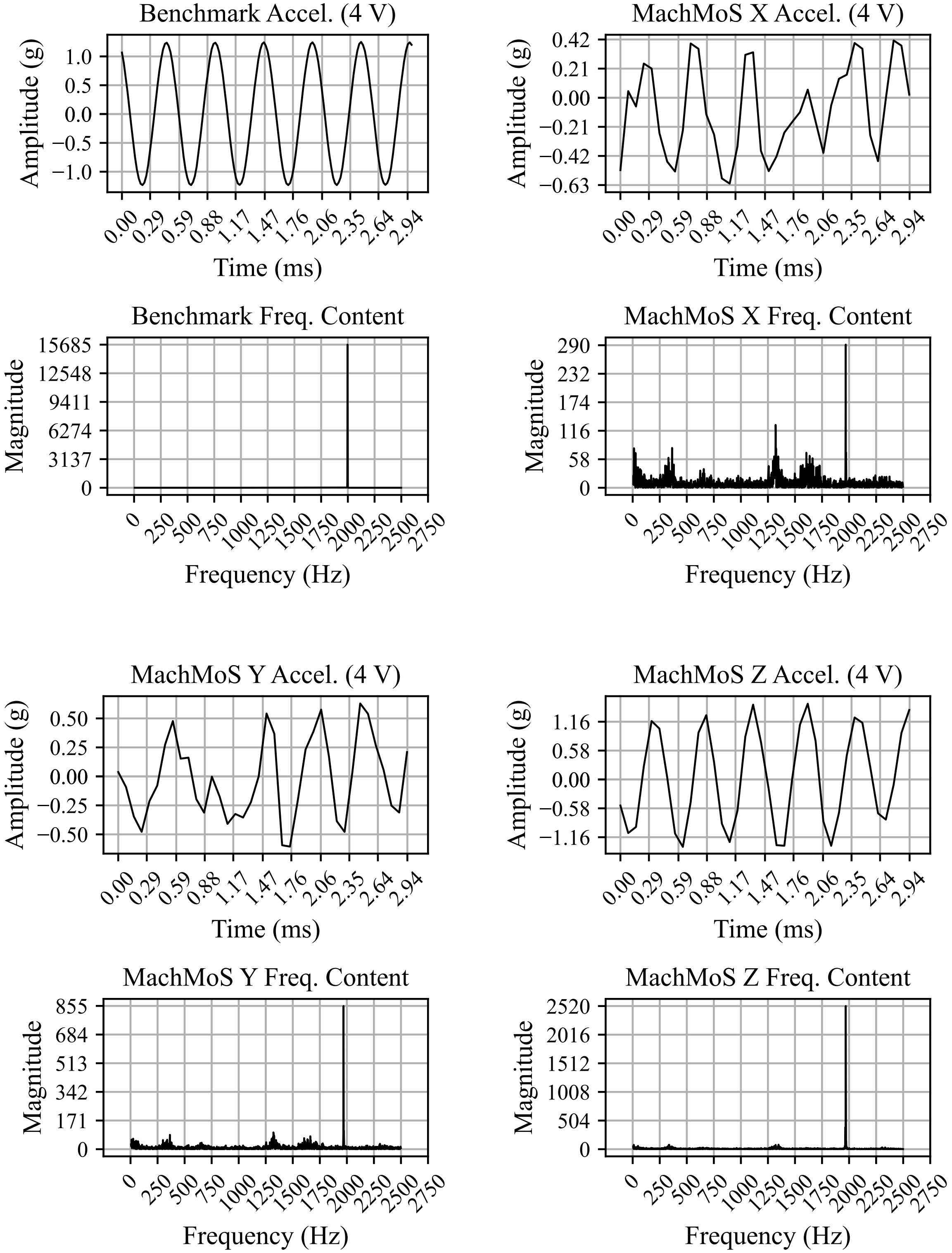


Figure F.9 - Accelerometer Test Results: 2 kHz, 4 V Input

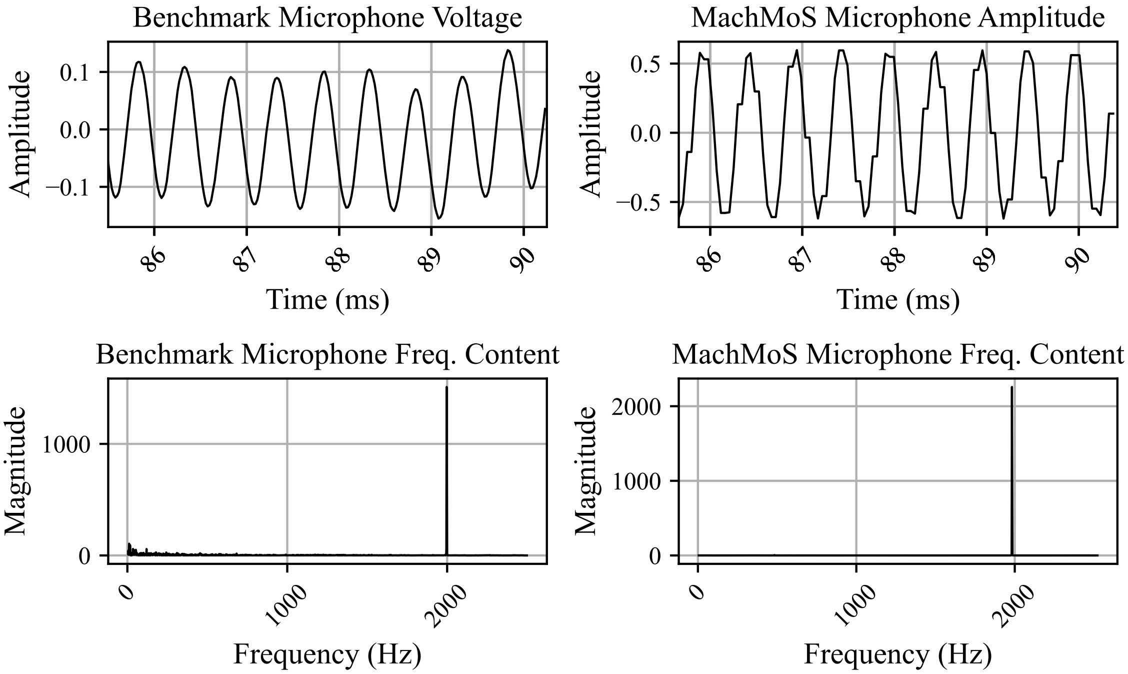


Figure F.10 - Microphone Test Results: 2 kHz, 4 V Input

F.1.6 3 kHz, 4 V and 8 V Input (Microphone)

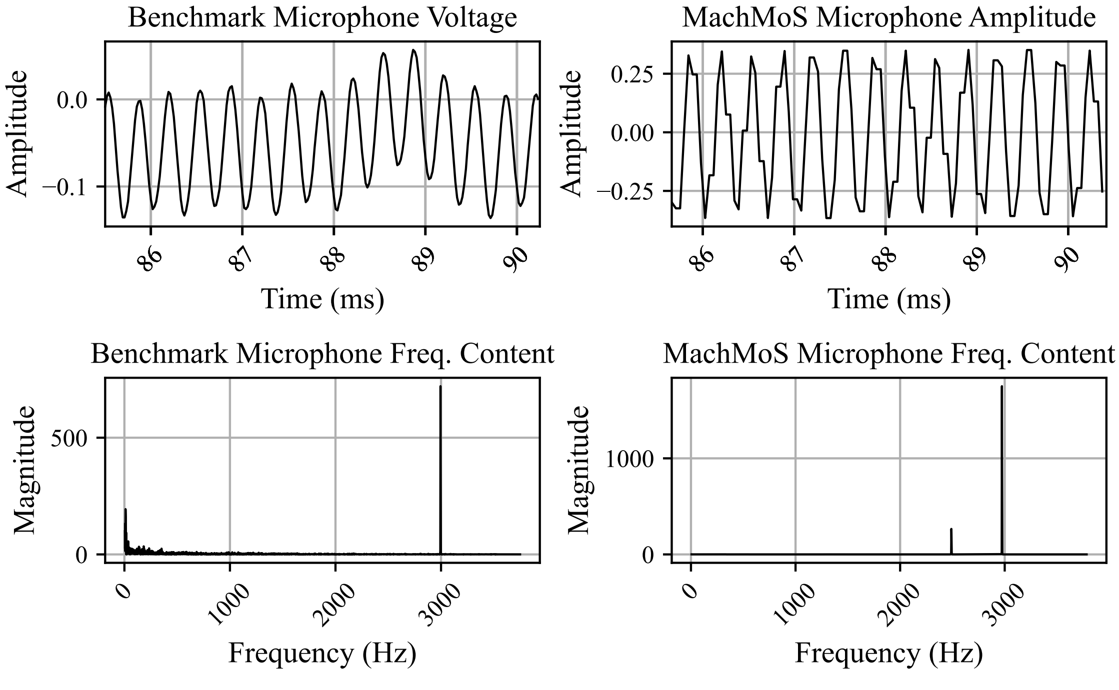


Figure F.11 - Microphone Test Results: 3 kHz, 4 V Input

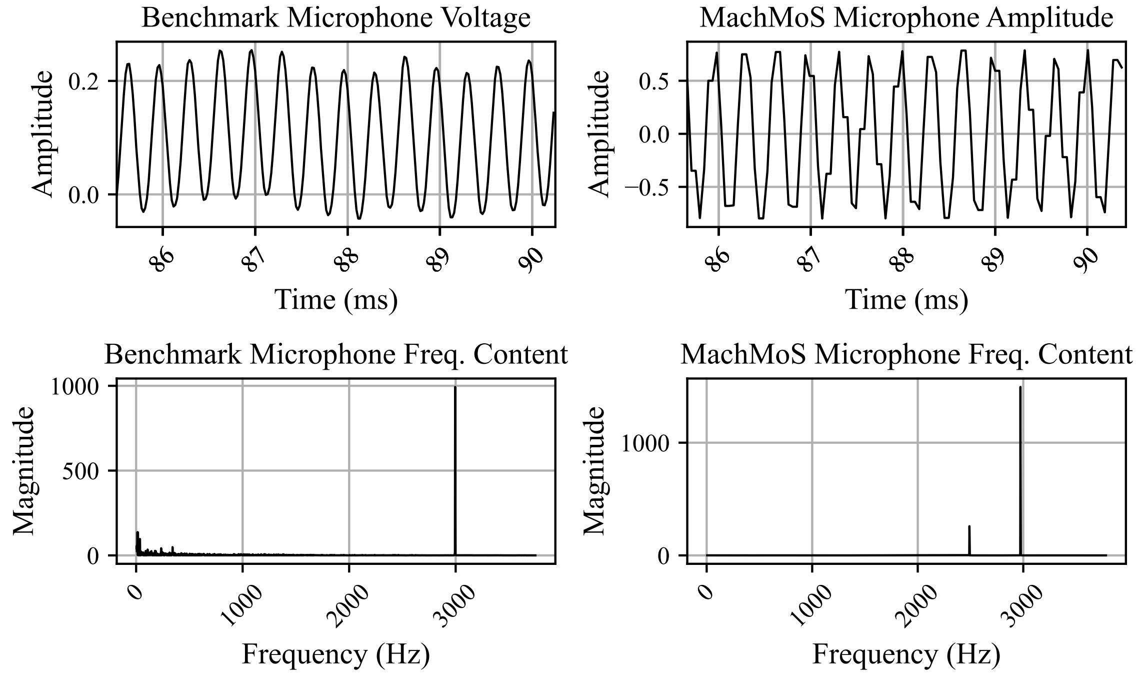
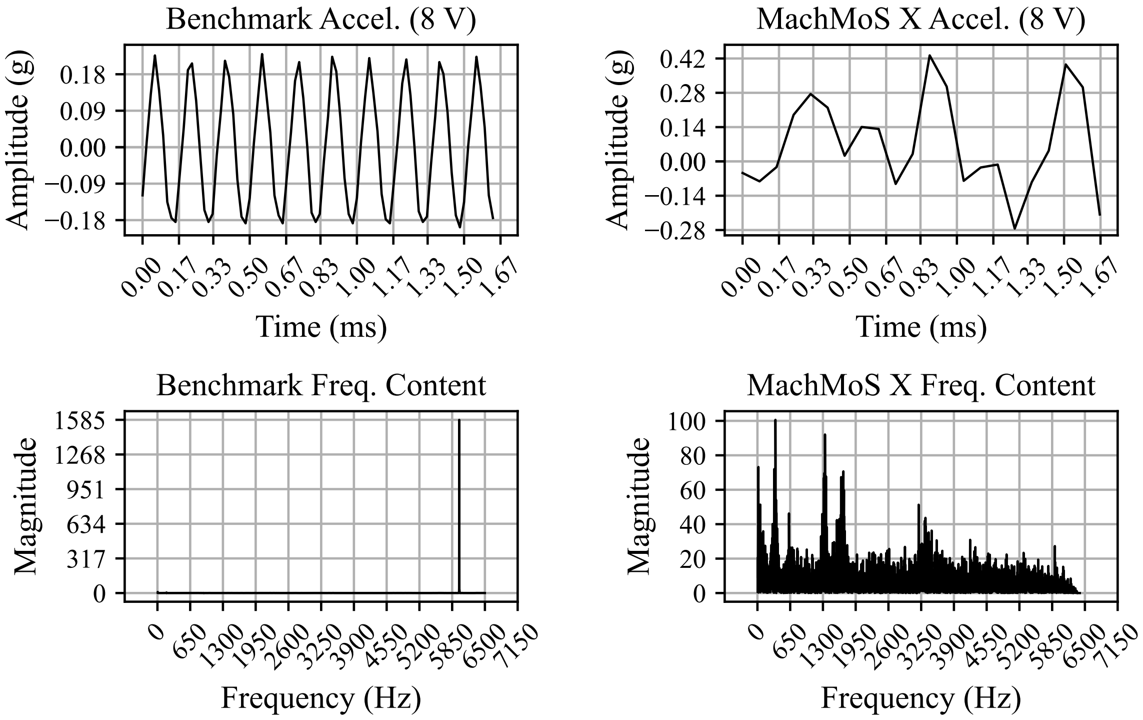


Figure F.12 - Microphone Test Results: 3 kHz, 8 V Input

F.1.7 6 kHz, 8 V Input



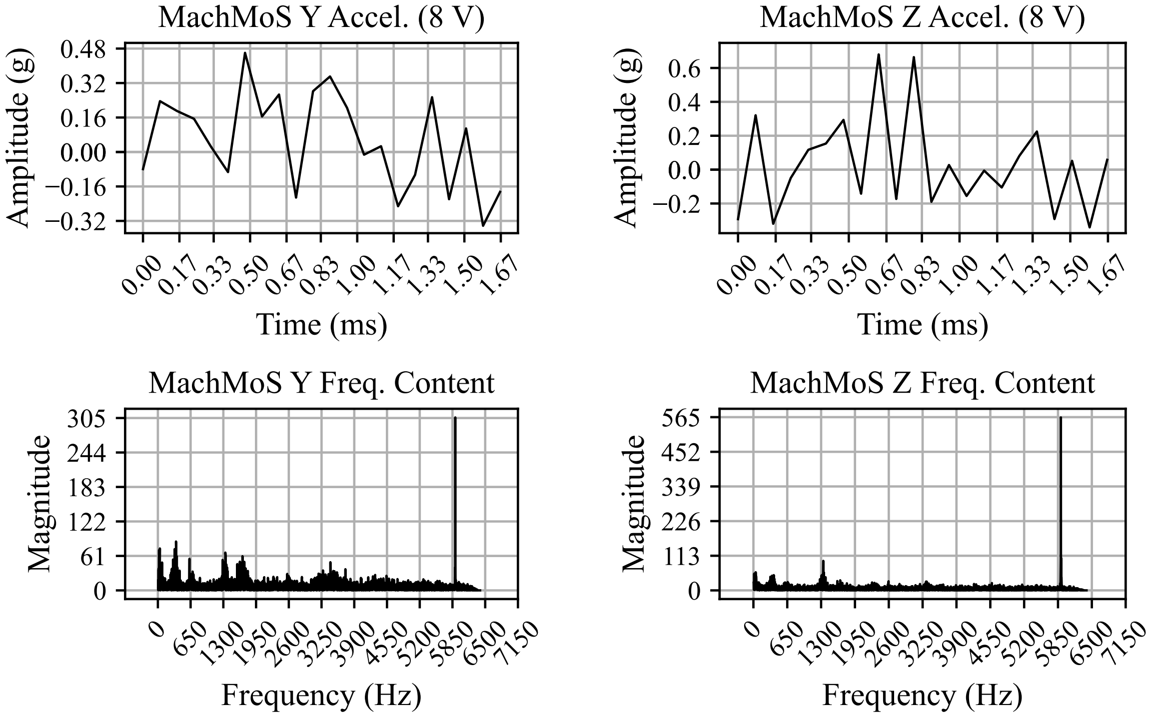


Figure F.13 - Accelerometer Test Results: 6 kHz, 8 V Input

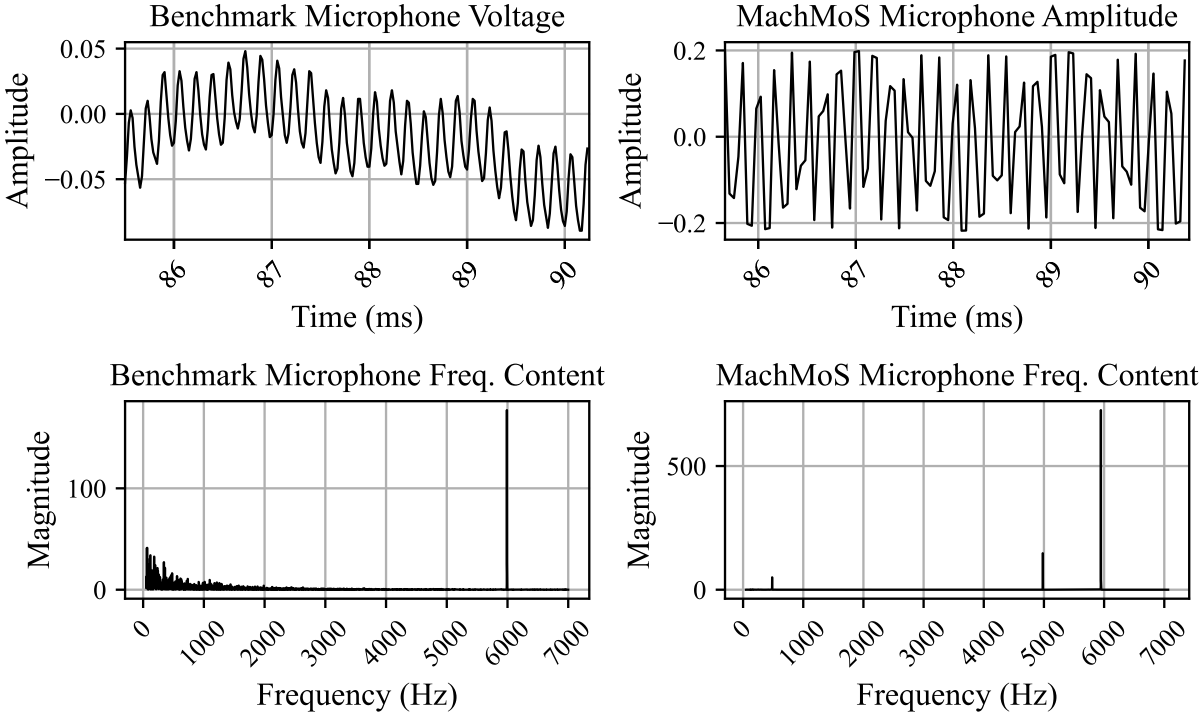


Figure F.14 - Microphone Test Results: 6 kHz, 8 V Input

F.2 Aliasing Testing

These tests are conducted with the KX134-1211 accelerometer LP filter configured to ODR/9. Two tests were conducted for the accelerometer: 7 kHz (10 V) and 9.6 kHz (40 V). The microphone is tested at 12.6 kHz (10 V).

F.2.1 7 kHz, 10 V Input (Accelerometer)

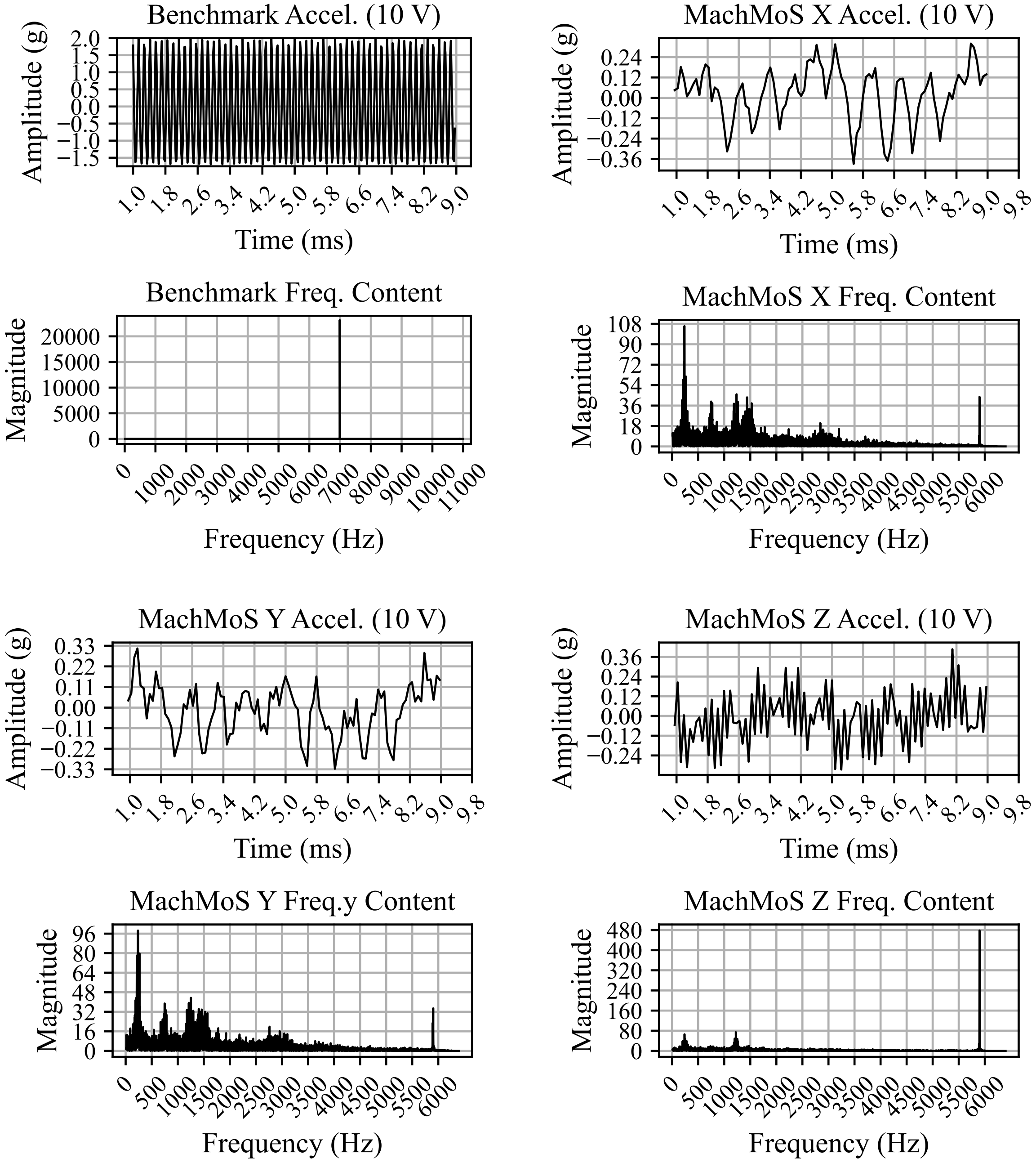


Figure F.15 - Accelerometer Aliasing Test Results: 7 kHz, 10 V Input

F.2.2 9.6 kHz, 40 V Input (Accelerometer)

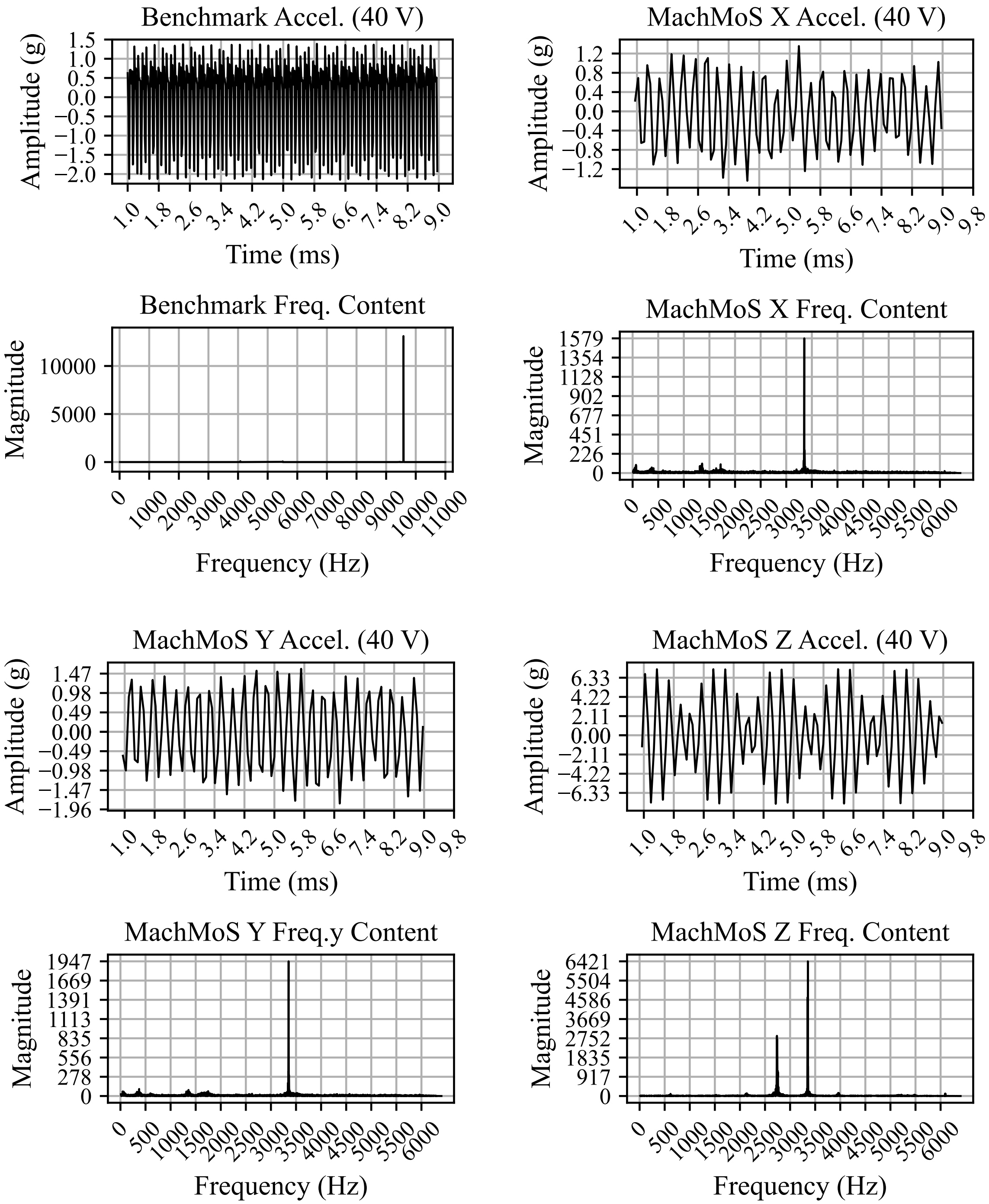


Figure F.16 - Accelerometer Aliasing Test Results: 9.6 kHz, 40 V Input

F.2.3 12.6 kHz, 10 V Input (Microphone)

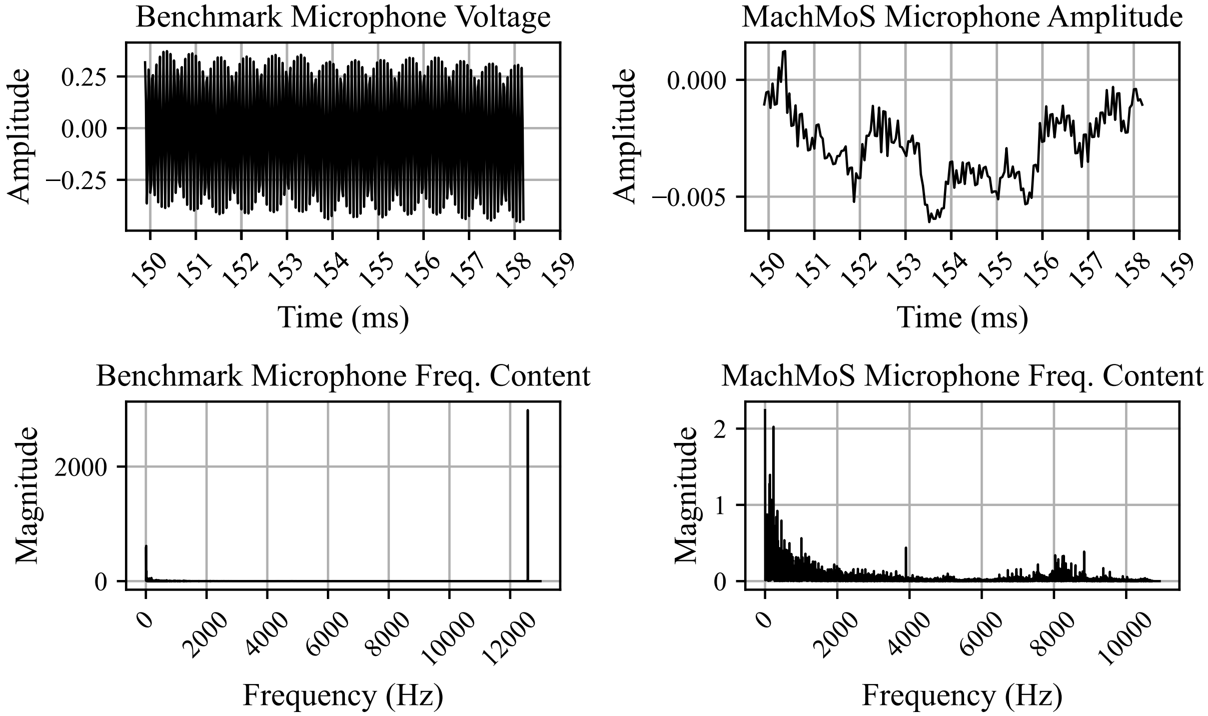


Figure F.17 - Microphone Aliasing Test Results: 12.6 kHz, 10 V Input

Appendix G Code

This appendix provides some of the important code created for this project.

G.1 MachMoS Node Embedded Code

The embedded code for the MachMoS node was not written by the author. It may be consulted on the Dumond Design Lab GitHub repository, once it is made open access.

G.2 MachMoS Node Sensor Validation

This code evaluates the data from the MachMoS node by comparing it against a benchmark.

G.2.1 Libraries and Functions

```
# MachMoS Node libraries
import serial
from serial.tools import list_ports as lstprts
import time
from datetime import datetime
import os
import wave

# DAQ Libraries
import nidaqmx as daq
from nidaqmx.constants import Edge
from nidaqmx.constants import AcquisitionType

# Visualisation libraries and other useful libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from scipy.signal import butter,filtfilt

# Matplotlib settings
params = {'legend.fontsize': '14',
          'figure.figsize': (8, 4),
          'figure.dpi': 300,
         'axes.labelsize': '12',
         'axes.titlesize':'12',
         'xtick.labelsize':'10',
         'ytick.labelsize':'10',
         'legend.fontsize': '10',
         'font.family': 'Times New Roman'}
plt.rcParams.update(params)

## Here are some useful functions for spectrum analysis
def butter_lowpass_filter(data, cutoff, fs, order):
    normal_cutoff = cutoff / (0.5*fs)
    # Get the filter coefficients
    b, a = butter(order, normal_cutoff, btype='low', analog=False)
    y = filtfilt(b, a, data)
    return y
```

```

def make_ffts(raw_data, period):
    #windowed_data = np.pad(raw_data * np.hanning(len(raw_data)), (0, len(raw_data)),
mode='constant')
    windowed_data = raw_data * np.hanning(len(raw_data))
    fft = np.fft.rfft(windowed_data)
    fft = np.abs(np.real(fft))
    # Multiply frequency bins as we'll only use the positive side of the frequencies
    frequencies = np.fft.rfftfreq(len(raw_data), d=period)
    return [frequencies, fft]

def interp_peak_freq(fft):

    # Find the index of the maximum magnitude (the peak), while ignoring frequencies
neighbouring DC
    peak_idx = np.argmax(fft[1][10:])+10 # We must shift the index by the qty of freqs
we offset away from DC

    # Quadratic interpolation to find a more accurate peak frequency
    if 0 < peak_idx < len(fft[1]) - 1:
        alpha, beta, gamma = fft[1][peak_idx-1], fft[1][peak_idx], fft[1][peak_idx+1]
        delta = 0.5 * (alpha - gamma) / (alpha - 2 * beta + gamma) # Shift in bin
index for interpolated peak
        return fft[0][peak_idx] + delta * (fft[0][1] - fft[0][0]) # Correct frequency
using interpolated bin shift
    else:
        return fft[0][peak_idx]

def getSNR(fft, frequencies, signal_frequency, percentage):
    start_rng = 0
    end_rng = 0
    for i in range(len(frequencies)):
        if frequencies[i] > (signal_frequency - signal_frequency*percentage) and
start_rng == 0:
            start_rng = i
        if frequencies[i] > (signal_frequency + signal_frequency*percentage) and
end_rng == 0:
            end_rng = i
            break

    sig_pow = np.sum(np.power(fft[start_rng:end_rng],2))
    noise_pow = np.sum(np.power(fft[:start_rng],2)) +
np.sum(np.power(fft[end_rng:],2))
    print(f"The signal power and noise power are: {sig_pow} and {noise_pow}")
    return 10*np.log10(sig_pow/noise_pow)
```

G.2.2 Data Capture Option

G.2.2.1 MachMoS Node COM setup.

This code block sets up serial communication with the MachMoS Node. Running this code is given as an option. Loading data is also possible, as shown in section G.2.3.

```python
# list all available ports
ports = lstprts.comports()
print("Available serial ports:")
for i in range(len(ports)):
     print(f"{i + 1}: {ports[i].device}")

# prompt the user to select a port
port_selection = len(ports)
print(f"Select a port ({port_selection}):", end=" ")
try:
     # port_selection = int(input())
     port_selection = 1
     port = ports[port_selection - 1].device
     # connect to the selected port
     ser = serial.Serial(port=port,
                         baudrate=9600,
                         bytesize=8,
                         parity=serial.PARITY_NONE,
                         stopbits=serial.STOPBITS_ONE)
     ser.close()
     print(f"{port} configured.")
     print()
except:
     print("\nNo device detected.")

machmos_accel_rate = 12800.0 # Hz
machmos_mic_rate = 21854.0 # Hz
```

G.2.2.2 Benchmark Accelerometer

The Benchmark Accelerometer is then set up. The benchmark data is processed into digital data by a National Instruments DAQ.

```python
ni_capture_task = daq.Task()
ni_rate = 52000.0    # Hz
ni_time = 1.0      # seconds
ni_samples = (int) (ni_rate/ni_time)

ni_capture_task.ai_channels.add_ai_voltage_chan("Dev1/ai5",
name_to_assign_to_channel="Benchmark Accel", min_val=-10, max_val=10)
ni_capture_task.ai_channels.add_ai_voltage_chan("Dev1/ai0",
name_to_assign_to_channel="Benchmark Microphone", min_val=-1, max_val=1)
print(ni_capture_task.channel_names)
ni_capture_task.timing.cfg_samp_clk_timing(rate=ni_rate, source="",
active_edge=Edge.RISING, sample_mode=AcquisitionType.FINITE,
samps_per_chan=ni_samples)
```

G.2.2.3 Capture Data

After setting up, the data is captured.

```python
ser.close() # Ensures previously opened serial is closed before we try reopening it.
now = datetime.now()# generate session as formatted date and time
session = now.strftime("%Y%m%d-%H%M%S")
print(f"Session: {session}")

# Start the benchmark capture task. Data coming back is in 1/10th of a G (100mV per
G), so multiply by 10.
benchmark_accel_data, benchmark_mic_data = [np.zeros(ni_samples),
```

```
np.zeros(ni_samples)] #
np.array(ni_capture_task.read(number_of_samples_per_channel=ni_samples, timeout=10))
benchmark_accel_data *= 10
benchmark_mic_data /= .045 # Conversion to Pa

transfer_complete = False
temp_in_waiting = 0
while not transfer_complete:
    outputs = []
    accelData = []
    micData = []
    readingAccel = False
    readingMic = False

    ser.open() # Connect to the MachMoS Node over serial.

    # send "START" command to the MachMoS node
    ser.write(b"START\n")

    while True:
        time.sleep(1)
        print(ser.in_waiting)
        if ser.in_waiting == temp_in_waiting and ser.in_waiting != 0:
            print("Serial is done its task. Reading buffer.")
            break
        temp_in_waiting = ser.in_waiting

    while True:
        raw_bytes = ser.readline()

        sample = raw_bytes.decode("utf-8", errors="ignore").strip()
        outputs.append(sample)

        if sample == "START ACCEL":
            print("Accelerometer data started")
            readingAccel = True
            continue
        elif sample == "END ACCEL":
            print("Accelerometer data ended")
            readingAccel = False
            continue
        elif sample == "START MIC":
            print("Microphone data started")
            readingMic = True
            continue
        elif sample == "END MIC":
            print("Microphone data ended")
            readingMic = False
            continue
        elif sample == "END":
            break

        if readingAccel:
            accelData.append(sample)
        elif readingMic:
            micData.append(sample)

    #resultFile.close()
    ser.close() # Release the port.

    # print the number of lines
    print("Lines read:", len(outputs))
    print("Accelerometer data:", len(accelData))
```

```
    print("Microphone data:", len(micData))

    if (len(accelData) == machmos_accel_rate and len(micData) == machmos_mic_rate):

        transfer_complete = True

        # make directory for the session
        os.mkdir(session)
        session = f"{session}/data"

        # write the samples to a file
        with open(f"{session}_log.txt", "w") as f:
            for output in outputs:
                f.write(output)
                f.write("\n")
            print(f"Log saved to {session}.txt")

        with open(f"{session}_accel_raw.csv", "w") as f:
            for sample in accelData:
                f.write(sample)
                f.write("\n")
            print(f"Accelerometer data saved to {session}_accel_raw.csv")

        with open(f"{session}_mic_raw.csv", "w") as f:
            for sample in micData:
                f.write(sample)
                f.write("\n")
            print(f"Microphone data saved to {session}_mic_raw.csv")

        # parse the accelerometer data
        machmos_accel_samples = []
        machmos_mic_samples = []
        for line in accelData:
            machmos_accel_samples.append([float(x) for x in line.strip().split(",")])

        for line in micData:
            machmos_mic_samples.append([float(x) for x in line.strip().split(",")])

        # s[0] is id, s[1] is x, s[2] is y, s[3] is z
        machmos_accel_samples = np.array(machmos_accel_samples)
        machmos_mic_samples = np.array(machmos_mic_samples)
    else:
        print("Data incomplete. Retrying.")

# Combine Mic bytes together
# Microphone data comes in as 24 bit data in byte chunks [Index, Byte0, Byte1, Byte2],
where Byte 0 is the least significant one.
machmos_mic_samples_stripped = machmos_mic_samples.transpose()[1:].transpose()
machmos_mic_samples_combined = [None]*len(machmos_mic_samples_stripped)

i = 0
for sample in machmos_mic_samples_stripped:
    machmos_mic_samples_combined[i] = (int(sample[2]) << 16) | (int(sample[1]) & 0xFF)
<< 8 | (int(sample[0]) & 0xFF) # multiply byte 1 by 2^8 and add LSByte
    i += 1
```

G.2.2.4 Create Dataframes from the Data & Save

```
conversion_factor = 1.0/4098.0

# machmos_accel_df = pd.DataFrame(data=machmos_accel_samples[1:,1:], # values
#               index=machmos_accel_samples.transpose()[0][1:],
#               columns=['X', 'Y', 'Z'])    # 1st column as index
```

```

machmos_mic_df = pd.DataFrame(data=np.array(machmos_mic_samples_combined),
            columns=['Mic_Voltage'])

# for axis in ['X', 'Y', 'Z']: machmos_accel_df[axis] =
machmos_accel_df[axis].multiply(conversion_factor)
machmos_mic_df['Mic_Voltage'] = machmos_mic_df['Mic_Voltage'].multiply(1.0/8388607.0)
# 1/(2^23-1) convert to ratio of full scale

# benchmark_accel_df = pd.DataFrame(data=benchmark_accel_data, columns=['Amplitude
(g)'])
benchmark_mic_df = pd.DataFrame(data=benchmark_mic_data, columns=['Mic Pressure
(Pa)'])

# Save data
# machmos_accel_df.to_csv(f"{session}_accel_dataframe.csv")
# benchmark_accel_df.to_csv(f"{session}_benchmark_accel_dataframe.csv")

machmos_mic_df.to_csv(f"{session}_mic_dataframe.csv")
benchmark_mic_df.to_csv(f"{session}_benchmark_mic_dataframe.csv")
```

G.2.3 Load Data Option

Alternatively, data can be loaded from an existing CSV. 1kHz example shown.

```
session = '20240925-163545 1kHz 4V pk-pk\\data' # 1kHz
# session = "20241004-101529 7kHz, 10V pk-pk, ODR_9 Aliasing test\\data" # Alias Test,
7kHz
# session = "20240925-171332 9.6kHz, 40V pk-pk Aliasing test\\data" # Alias Test,
9.6kHz

machmos_accel_df = pd.read_csv(f"{session}_accel_dataframe.csv")
benchmark_accel_df = pd.read_csv(f"{session}_benchmark_accel_dataframe.csv")

machmos_mic_df = pd.read_csv(f"{session}_mic_dataframe.csv")
benchmark_mic_df = pd.read_csv(f"{session}_benchmark_mic_dataframe.csv")

ni_rate = 52000.0    # Hz
ni_time = 1.0      # seconds
ni_samples = (int) (ni_rate/ni_time)

machmos_accel_rate = 12600.0 # Hz, new sampling rate
machmos_mic_rate = 21854.0
```

G.2.4 Display Data

The data is then displayed.

G.2.4.1 Accelerometer Data

```
minimum = 10
maximum = 600
min_bench = int(minimum*ni_rate/machmos_accel_rate)
max_bench = int(maximum*ni_rate/machmos_accel_rate)
maxHz = 1500
minHz = 10
```

```

t = range(int(maximum)-int(minimum)+1) #machmos_accel_df.index[minimum:maximum]

x_avg = np.average(machmos_accel_df['X'])
y_avg = np.average(machmos_accel_df['Y'])
z_avg = np.average(machmos_accel_df['Z'])

df = machmos_accel_df

df['Time_fix'] = df.index * 1.0/machmos_accel_rate
benchmark_accel_times = np.array(range(len(benchmark_accel_df['Amplitude
(g)'])))/ni_rate

T = 1.0/machmos_accel_rate
fs = machmos_accel_rate
nyq = 0.5 * fs  # Nyquist Frequency
cutoff = nyq*.99     # desired cutoff frequency of the filter, Hz
order = 2

df['X_filt'] = butter_lowpass_filter(df['X'], cutoff, fs, order)
df['Y_filt'] = butter_lowpass_filter(df['Y'], cutoff, fs, order)
df['Z_filt'] = butter_lowpass_filter(df['Z'], cutoff, fs, order)


benchmark_fft = make_ffts(benchmark_accel_df['Amplitude (g)'], 1.0/ni_rate)
x_fft = make_ffts(df['X_filt'], 1.0/df['X_filt'].size)
y_fft = make_ffts(df['Y_filt'], 1.0/df['Y_filt'].size)
z_fft = make_ffts(df['Z_filt'], 1.0/df['Z_filt'].size)

fig, axes = plt.subplots(8, 1, sharex=False, figsize=(10,80))

axes[0].set_title('Benchmark Acceleration')
axes[1].set_title('Benchmark Frequency Content')
axes[2].set_title('MachMoS X Acceleration Sample')
axes[3].set_title('MachMoS X Frequency Content')
axes[4].set_title('MachMoS Y Acceleration Sample')
axes[5].set_title('MachMoS Y Frequency Content')
axes[6].set_title('MachMoS Z Acceleration Sample')
axes[7].set_title('MachMoS Z Frequency Content')

for ax in axes:
    ax.tick_params(axis='x', labelrotation=45)
    ax.grid(True)

for i in [0,2,4,6]:
    axes[i].set_xlabel("Time (s)")
    axes[i].set_ylabel("Amplitude (g)")

for i in [2,4,6]:
    axes[i].set_xticks(np.arange(min(df['Time_fix'].loc[minimum:maximum]),
max(df['Time_fix'].loc[minimum:maximum])+.001,
(max(df['Time_fix'].loc[minimum:maximum])-
min(df['Time_fix'].loc[minimum:maximum]))/20.0))


axes[0].plot(benchmark_accel_times[min_bench:max_bench], benchmark_accel_df['Amplitude
(g)'].loc[min_bench:max_bench-1], "k-", linewidth=1)
axes[0].set_xticks(np.arange(min(benchmark_accel_times[min_bench:max_bench]),
max(benchmark_accel_times[min_bench:max_bench])+.001,
(max(benchmark_accel_times[min_bench:max_bench])-
min(benchmark_accel_times[min_bench:max_bench]))/20.0))
axes[2].plot(df['Time_fix'].loc[minimum:maximum], df['X_filt'].loc[minimum:maximum] -
```

```
x_avg, "k-", linewidth=1)
axes[4].plot(df['Time_fix'].loc[minimum:maximum], df['Y_filt'].loc[minimum:maximum] -
y_avg, "k-", linewidth=1)
axes[6].plot(df['Time_fix'].loc[minimum:maximum], df['Z_filt'].loc[minimum:maximum] -
z_avg, "k-", linewidth=1)

maxHz = 1500
minHz = 10

for i in [1,3,5,7]:
    axes[i].set_xlabel("Frequency (Hz)")
    axes[i].set_ylabel("Magnitude")

for i in [3,5,7]:
    axes[i].set_xticks(np.arange(0,len(x_fft[0][:maxHz])+2, 250))

axes[1].set_xticks(np.arange(0,len(benchmark_fft[0][minHz:maxHz])+2, 250))

axes[1].plot(benchmark_fft[0][minHz:maxHz], benchmark_fft[1][minHz:maxHz], "k-",
linewidth=0.5)
axes[3].plot(x_fft[0][minHz:maxHz], x_fft[1][minHz:maxHz], "k-", linewidth=1)
axes[5].plot(y_fft[0][minHz:maxHz], y_fft[1][minHz:maxHz], "k-", linewidth=1)
axes[7].plot(z_fft[0][minHz:maxHz], z_fft[1][minHz:maxHz], "k-", linewidth=1)

print(f"Most prominent frequency for Benchmark accel:
{interp_peak_freq(benchmark_fft)}")
print(f"Most prominent frequency for MachMoS node accel (x):
{interp_peak_freq(x_fft)}")
print(f"Most prominent frequency for MachMoS node accel (y):
{interp_peak_freq(y_fft)}")
print(f"Most prominent frequency for MachMoS node accel (z):
{interp_peak_freq(z_fft)}")

fig.savefig(session + '_FFT_accel' + '.png')
print(f"Figure saved to {session}_FFT_accel.png")

fig, axes = plt.subplots(2, 1, sharex=False, figsize=(6,4))

def reset_axes_labels():
    for ax in axes:
        ax.tick_params(axis='x', labelrotation=45)
        ax.grid(True)
    axes[0].set_xlabel("Time (ms)")
    axes[0].set_ylabel("Amplitude (g)")
    axes[1].set_xlabel("Frequency (Hz)")
    axes[1].set_ylabel("Magnitude")

fig.tight_layout()

# BENCHMARK
axes[0].set_title('Benchmark Acceleration')
axes[1].set_title('Benchmark Frequency Content')

# Benchmark xticks
axes[0].set_xticks(np.round(np.arange(min(benchmark_accel_times[min_bench:max_bench]),
max(benchmark_accel_times[min_bench:max_bench])+.001,
(max(benchmark_accel_times[min_bench:max_bench])-
min(benchmark_accel_times[min_bench:max_bench]))/20.0)*1000,2))
axes[0].set_yticks([-1,-0.5,0,0.5,1])
axes[1].set_xticks(np.arange(0,len(benchmark_fft[0][minHz:maxHz])+100, 250))

# Plot
axes[0].plot(1000*benchmark_accel_times[min_bench:max_bench],
```

```
benchmark_accel_df['Amplitude (g)'].loc[min_bench:max_bench-1], "k-", linewidth=1.25)
axes[1].plot(benchmark_fft[0][minHz:maxHz], benchmark_fft[1][minHz:maxHz], "k-",
linewidth=1.25)

# Save benchmark
reset_axes_labels()
fig.tight_layout()
fig.savefig("C:/Users/Simon/Dropbox/__Université - Cours/_1st Masters/_Masters
Thesis/Papers, Conferences/CSME/2025/Accelerometer Aliasing_Benchmark.svg")
axes[0].cla()
axes[1].cla()

#MACHMOS
# MachMoS xticks
axes[0].set_xticks(np.round(np.arange(min(df['Time_fix'].loc[minimum:maximum]),
max(df['Time_fix'].loc[minimum:maximum])+.001,
(max(df['Time_fix'].loc[minimum:maximum])-
min(df['Time_fix'].loc[minimum:maximum]))/20.0)*1000,2))
axes[0].set_yticks([-1,-0.5,0,0.5,1])
axes[1].set_xticks(np.arange(0,len(x_fft[0][:maxHz])+100, 250))

# X
axes[0].set_title('MachMoS X Acceleration Sample')
axes[1].set_title('MachMoS X Frequency Content')
axes[0].plot(1000*df['Time_fix'].loc[minimum:maximum],
df['X_filt'].loc[minimum:maximum] - x_avg, "k-", linewidth=1.25)
axes[1].plot(x_fft[0][minHz:maxHz], x_fft[1][minHz:maxHz], "k-", linewidth=1.25)

reset_axes_labels()
fig.tight_layout()
fig.savefig(…)
axes[0].cla()
axes[1].cla()

# Y
axes[0].set_title('MachMoS Y Acceleration Sample')
axes[0].set_yticks([-1,-0.5,0,0.5,1])
axes[1].set_title('MachMoS Y Frequency Content')
axes[0].plot(1000*df['Time_fix'].loc[minimum:maximum],
df['Y_filt'].loc[minimum:maximum] - y_avg, "k-", linewidth=1.25)
axes[1].plot(y_fft[0][minHz:maxHz], y_fft[1][minHz:maxHz], "k-", linewidth=1.25)

reset_axes_labels()
fig.tight_layout()
fig.savefig(…)
axes[0].cla()
axes[1].cla()

# Z
axes[0].set_title('MachMoS Z Acceleration Sample')
axes[0].set_yticks(np.multiply([-1,-0.5,0,0.5,1],1))
# axes[0].set_yticks([-6,-4,-2,0,2,4,6]) % Anti-Aliasing 40V pk-pk test, very high
amplitude case.
axes[1].set_title('MachMoS Z Frequency Content')
axes[0].plot(1000*df['Time_fix'].loc[minimum:maximum],
df['Z_filt'].loc[minimum:maximum] - z_avg, "k-", linewidth=1.25)
axes[1].plot(z_fft[0][minHz:maxHz], z_fft[1][minHz:maxHz], "k-", linewidth=1.25)
reset_axes_labels()
fig.tight_layout()
fig.savefig(…)

print(f"Most prominent frequency for Benchmark accel:
{interp_peak_freq(benchmark_fft)}")
```

```
print(f"Most prominent frequency for MachMoS node accel (x):
{interp_peak_freq(x_fft)}")
print(f"Most prominent frequency for MachMoS node accel (y):
{interp_peak_freq(y_fft)}")
print(f"Most prominent frequency for MachMoS node accel (z):
{interp_peak_freq(z_fft)}")
def get_stats_min_max_pkpk(data):
    avg = np.average(data)
    max = np.max(data)-avg
    min = np.min(data)-avg
    pk2pk = max-min
    return min, max, pk2pk

min, max, pk2pk = get_stats_min_max_pkpk(benchmark_accel_df['Amplitude (g)'])
print(f"Benchmark Min: {min}")
print(f"Benchmark Max: {max}")
print(f"pk-pk: {pk2pk}")

tranche = 128
benchmark_pk2pk = []
benchmark_min = []
benchmark_max = []

pk2pk_xyz = [[],[],[]]
min_xyz = [[],[],[]]
max_xyz = [[],[],[]]

for i in np.arange(0,len(benchmark_accel_df['Amplitude (g)']) - tranche, tranche):
    min, max, pk2pk = get_stats_min_max_pkpk(benchmark_accel_df['Amplitude
(g)'][i:i+tranche])
    benchmark_pk2pk.append(pk2pk)
    benchmark_min.append(min)
    benchmark_max.append(max)

for i in np.arange(0,len(df['X_filt']) - tranche, tranche):
    min_x, max_x, pk2pk_x = get_stats_min_max_pkpk(df['X_filt'][i:i+tranche])
    min_y, max_y, pk2pk_y = get_stats_min_max_pkpk(df['Y_filt'][i:i+tranche])
    min_z, max_z, pk2pk_z = get_stats_min_max_pkpk(df['Z_filt'][i:i+tranche])

    pk2pk_xyz[0].append(pk2pk_x)
    pk2pk_xyz[1].append(pk2pk_y)
    pk2pk_xyz[2].append(pk2pk_z)
    min_xyz[0].append(min_x)
    min_xyz[1].append(min_y)
    min_xyz[2].append(min_z)
    max_xyz[0].append(max_x)
    max_xyz[1].append(max_y)
    max_xyz[2].append(max_z)

print("\nPeak 2 peak:")
print(np.average(benchmark_pk2pk))
for i in range(3):
    print(np.average(pk2pk_xyz[i]))

print("Minima:")
min = np.min(benchmark_min)
max = np.max(benchmark_min)
print(f"{min} to {max} ({max-min})")
for i in range(3):
    min = np.min(min_xyz[i])
    max = np.max(min_xyz[i])
    print(f"{min} to {max} ({max-min})")

```

```python
print("Maxima:")
min = np.min(benchmark_max)
max = np.max(benchmark_max)
print(f"{min} to {max} ({max-min})")
for i in range(3):
    min = np.min(max_xyz[i])
    max = np.max(max_xyz[i])
    print(f"{min} to {max} ({max-min})")
```

G.2.4.2 Microphone Data

```python
# Filter requirements.
df = machmos_mic_df
df['Time_fix'] = df.index * 1.0/machmos_mic_rate
benchmark_mic_times = np.array(range(len(benchmark_mic_df['Mic Pressure
(Pa)'])))/ni_rate
T = df['Time_fix'][len(df['Time_fix'])-1]                                  # Sample
Period
fs = 1/(df['Time_fix'][len(df['Time_fix'])-1]/(len(df['Time_fix'])))       # sample
rate, Hz
nyq = 0.5 * fs          # Nyquist Frequency
cutoff = nyq*.99        # desired cutoff frequency of the filter, Hz
order = 2

df['Mic_Voltage'] = butter_lowpass_filter(df['Mic_Voltage'], cutoff, fs, order)

mic_fft = make_ffts(df['Mic_Voltage'], 1.0/machmos_mic_rate)
benchmark_mic_fft = make_ffts(benchmark_mic_df['Mic Pressure (Pa)'], 1.0/ni_rate)

fig, axes = plt.subplots(2, 2, sharex=False, figsize=(10,3.5))
plt.rcParams['font.family'] = 'serif'
plt.rcParams['font.serif'] = 'Times New Roman'

axes[0][0].set_title('Benchmark Microphone Voltage')
axes[1][0].set_title('Benchmark Microphone Frequency Content')
axes[0][1].set_title('MachMoS Microphone Amplitude')
axes[1][1].set_title('MachMoS Microphone Frequency Content')

benchmark_min = 0
benchmark_max = int(len(benchmark_mic_df['Mic Pressure (Pa)'])/10)
node_min = 0
node_max = int(len(machmos_mic_df['Time_fix'])/10)

for row in axes:
    for ax in row:
        ax.tick_params(axis='x', labelrotation=45)
        ax.grid(True)

for i in [0,1]:
    axes[0][i].set_xlabel("Time (ms)")
    axes[0][i].set_ylabel("Amplitude")

for i in [0,1]:
    axes[1][i].set_xlabel("Frequency (Hz)")
    axes[1][i].set_ylabel("Magnitude")

axes[0][0].plot(1000*benchmark_mic_times[benchmark_min:benchmark_max],
benchmark_mic_df['Mic Pressure (Pa)'][benchmark_min:benchmark_max], "k-", linewidth=1)
axes[0][0].set_xticks(np.round(np.arange(np.min(benchmark_mic_times[benchmark_min:benc
hmark_max]), np.max(benchmark_mic_times[benchmark_min:benchmark_max])+.001,
(np.max(benchmark_mic_times[benchmark_min:benchmark_max])-
np.min(benchmark_mic_times[benchmark_min:benchmark_max]))/20.0)*1000,0))
```

```
axes[0][1].plot(1000*machmos_mic_df['Time_fix'][node_min:node_max],
machmos_mic_df['Mic_Voltage'][node_min:node_max], "k-", linewidth=1)
axes[0][1].set_xticks(np.round(np.arange(np.min(machmos_mic_df['Time_fix'][node_min:no
de_max]), np.max(machmos_mic_df['Time_fix'][node_min:node_max])+.001,
(np.max(machmos_mic_df['Time_fix'][node_min:node_max])-
np.min(machmos_mic_df['Time_fix'][node_min:node_max]))/20)*1000,0))

maxHz = 1500

# Plot FFTs
axes[1][0].plot(benchmark_mic_fft[0][:maxHz], benchmark_mic_fft[1][:maxHz], "k-",
linewidth=1)
axes[1][0].set_xticks(np.arange(0,len(benchmark_mic_fft[0][:maxHz])+2, 100))
axes[1][1].plot(mic_fft[0][:maxHz], mic_fft[1][:maxHz], "k-", linewidth=1)
axes[1][1].set_xticks(np.arange(0,len(mic_fft[0][:maxHz])+2, 100))

print(f"Most prominent frequency for Benchmark mic:
{interp_peak_freq(benchmark_mic_fft)}")
print(f"Most prominent frequency for MachMoS node mic: {interp_peak_freq(mic_fft)}")

fig.tight_layout()
fig.savefig(…)
```

G.3 SolidWorks Thermal Simulations

Below is the VBA code used to automate iterating through the thermal simulations described in subsection 4.7.1 and Appendix A.4:

```
Option Explicit

Sub main()

    Dim swApp As SldWorks.SldWorks
    Dim part As SldWorks.ModelDoc2
    Dim CWObject As CosmosWorksLib.CwAddincallback
    Dim COSMOSWORKS As CosmosWorksLib.COSMOSWORKS
    Dim ActDoc As CosmosWorksLib.CWModelDoc
    Dim StudyMngr As CosmosWorksLib.CWStudyManager
    Dim Study As CosmosWorksLib.CWStudy
    Dim LBCMgr As CosmosWorksLib.CWLoadsAndRestraintsManager
    Dim CWConv As CosmosWorksLib.CWConvection
    Dim CWMachineTemp As CosmosWorksLib.CWTemperature
    Dim selManager As SldWorks.SelectionMgr
    Dim cwResults As CosmosWorksLib.cwResults
    Dim load As Object
    Dim selObj As Object
    Dim selection As String
    Dim varSel As Variant
    Dim bRet As Boolean
    Dim errcode As Long
    Dim nStep As Long
    Dim thermals As Variant
    Dim RTDthermals As Variant
```

```
    Dim midSurfaceRingTemp As Variant
    Dim rtdTemperatures As Variant
    Dim resultsFilepath As String

    ' Useful iterators
    Dim i As Long
    Dim j As Long
    Dim k As Long

    ' Get SolidWorks instance
    Set swApp = Application.SldWorks
    Set part = swApp.ActiveDoc
    ' Get Selection Manager
    Set selManager = part.SelectionManager
    ' Get Simulation add-in object
    Set CWObject = swApp.GetAddInObject("SldWorks.Simulation")
    ' Get CosmosWorks interface
    Set COSMOSWORKS = CWObject.COSMOSWORKS
    ' Get active Simulation document (for the current model)
    Set ActDoc = COSMOSWORKS.ActiveDoc()
    ' Get Study Manager
    Set StudyMngr = ActDoc.StudyManager()
    ' Get the second study in the tree
    Set Study = StudyMngr.GetStudy(1)
    ' Get Loads and Restraints Manager
    Set LBCMgr = Study.LoadsAndRestraintsManager
    ' Find the convection load and machine temperature params
    For i = 0 To LBCMgr.count
        Set load = LBCMgr.GetLoadsAndRestraints(i, errcode)
        If load.name = "Convection" Then
            Set CWConv = load
        ElseIf load.name = "MachineTemperature" Then
            Set CWMachineTemp = load
        End If
    Next i

    ' Load up the surfaces of interest from their base64 codes
    thermals = getSurface("mid", part)
    RTDthermals = getSurface("RTD", part)

    ' Ambient and machine temperatures for multiple simulations
    Dim ambientTemps As Variant
    Dim ambientTemp As Double
    Dim machineTemps As Variant
    Dim machineTemp As Double

    ' Useful variables to aggregate the data
    ' and evaluate the simulation results
    Dim surfaceTempGuess As Double
```

```
    Dim temperatureError As Double
    Dim iterations As Long
    Dim sum As Double
    Dim count As Long
    Dim avgRing As Double
    Dim avgRTD As Double
    Dim theta As Double

    ' Description of the heat transfer case to calculate the heat transfer coefficient
    Dim caseDescription As String
    Dim caseDescriptions As Variant
    caseDescriptions = Array("Horizontal Hot Plate Bottom Side", "Horizontal Hot Plate
Top Side", "Vertical")

    ' Max error between mid temperature and plate temperature guess
    Dim convergenceCriteria As Double
    convergenceCriteria = 0.5 ' degrees K

    ' Ambient temperatures for the simulation
    ambientTemps = Array(233, 253, 273, 293, 313, 323)

    ' Plate angle
    theta = 0

    For k = LBound(caseDescriptions) To UBound(caseDescriptions)
        caseDescription = caseDescriptions(k)
        ' Save CSV here:
        resultsFilepath = "[…]_thermal sim results_" & caseDescription

    For i = LBound(ambientTemps) To UBound(ambientTemps)
        ambientTemp = ambientTemps(i)
        machineTemps = generateMachineTemps(ambientTemp, 10)
        iterations = 0

        Debug.Print "Running new ambient temperature: " & Format(ambientTemp, "0.00")
        For j = LBound(machineTemps) To UBound(machineTemps)
            machineTemp = machineTemps(j)
            If iterations = 0 Then
                surfaceTempGuess = machineTemp
                Debug.Print "Starting new machine Temperature: " & Format(machineTemp,
"0.00")
            End If

            Debug.Print "Simulating with temperature guess: " &
Format(surfaceTempGuess, "0.00")

            ' Set the simulation parameters
            CWConv.ConvectionBeginEdit
            CWConv.Unit = 0
```

```
            CWConv.ConvectionCoefficient = getConvCoeff(surfaceTempGuess, ambientTemp,
theta, caseDescription)
            CWConv.BulkAmbientTemperature = ambientTemp
            CWConv.ConvectionEndEdit

            CWMachineTemp.TemperatureBeginEdit
            CWMachineTemp.Unit = 0
            CWMachineTemp.TemperatureValue = machineTemp
            CWMachineTemp.TemperatureEndEdit

            ' Solve the study
            Study.MeshAndRun

            ' Extract results and check convergence criteria
            Set cwResults = Study.Results
            nStep = cwResults.GetMaximumAvailableSteps

            midSurfaceRingTemp = cwResults.GetThermalForEntities(0, nStep, Nothing,
thermals, 0, errcode)
            If errcode <> 0 Then
                Debug.Print "No temperature results for the mid ring. Exiting."
                Exit Sub
            End If

            rtdTemperatures = cwResults.GetThermalForEntities(0, nStep, Nothing,
RTDthermals, 0, errcode)
            If errcode <> 0 Then
                Debug.Print "No temperature results for the RTD surface. Exiting."
                Exit Sub
            End If

            avgRing = getArrayAverage(midSurfaceRingTemp)
            avgRTD = getArrayAverage(rtdTemperatures)

            If avgRing = 0 Or avgRTD = 0 Then
                Debug.Print "Calculation error on average temperatures. Exiting."
                Exit Sub
            End If

            temperatureError = surfaceTempGuess - avgRing

            Debug.Print "Done! Here are the results:"
            Debug.Print "Mid Surface Temperature: " & Format(avgRing, "0.00") & "°K"
            Debug.Print "Temperature error, guess vs. mid surface: " &
Format(temperatureError, "0.00") & "°K"

            If Abs(temperatureError) > convergenceCriteria Then
                Debug.Print "Error too large, retrying with a new plate temperature."
                surfaceTempGuess = avgRing ' Use the mid temperture as the new guess
                ' Re-run with new guess.
```

```
                iterations = iterations + 1
                j = j - 1
            Else
                Debug.Print "Reached convergence criteria. RTD Temperature: " &
Format(avgRTD, "0.00") & "°K"
                Call WriteResultsToCSV(ambientTemp, machineTemp, surfaceTempGuess,
avgRing, avgRTD, resultsFilepath)
                iterations = 0
                Debug.Print "--------------------------------"
            End If

        Next j
    Next i

    Next k

End Sub

Private Function getConvCoeff(surfaceTemp As Double, ambTemp As Double, theta As
Double, description As String) As Double
    ' Assumes temperature inputs are in Kelvin
    Dim T_film As Double
    T_film = (surfaceTemp + ambTemp) / 2 ' Film temperature

    Dim charLen As Double
    charLen = 0.02837 ' meters

    Dim g As Double
    theta = theta * 3.1416 / 180 ' convert to rad
    g = 9.81 * Cos(theta) ' m/s^2

    ' Columns: Temperature (K), rho (kg/m^3), Cp (kJ/kg-K), mu (N-s/m^2), nu (m^2/s),
k (W/m-K), alpha (m^2/s), Pr
    Dim airProps As Variant
    airProps = getAirThermophysicalProps()

    Dim interpolatedProps(1 To 8) As Double
    Dim T1 As Double, T2 As Double, Y1 As Double, Y2 As Double

    Dim i As Long
    Dim j As Long
    ' Find bounding rows
    For i = 1 To 9
        If T_film >= airProps(i, 1) And T_film < airProps(i + 1, 1) Then
            T1 = airProps(i, 1): T2 = airProps(i + 1, 1)

            ' Interpolate each property
            For j = 1 To 8
                Y1 = airProps(i, j)
                Y2 = airProps(i + 1, j)
```

```
                interpolatedProps(j) = Y1 + (Y2 - Y1) * (T_film - T1) / (T2 - T1)
            Next j
        End If
    Next i

    Dim Ra As Double
    Ra = g * 1 / T_film * (surfaceTemp - ambTemp) * charLen ^ 3 / interpolatedProps(5)
/ interpolatedProps(7)

    Select Case description
    Case "Horizontal Hot Plate Bottom Side"
        getConvCoeff = 0.52 * Ra ^ 0.2 * interpolatedProps(6) / charLen
    Case "Horizontal Hot Plate Top Side"
        getConvCoeff = 0.54 * Ra ^ 0.25 * interpolatedProps(6) / charLen
    Case "Vertical"
        getConvCoeff = (0.68 + (0.67 * Ra ^ 0.25) / (1 + (0.492 /
interpolatedProps(8)) ^ (9 / 16)) ^ (4 / 9)) * interpolatedProps(6) / charLen
    End Select

End Function

Private Function getAirThermophysicalProps() As Variant
    Dim airProps(1 To 10, 1 To 8) As Double
    ' Columns:
    ' (1) Temperature (K), (2) rho (kg/m^3), (3) Cp (kJ/kg-K),
    ' (4) mu (N-s/m^2), (5) nu (m^2/s),
    ' (6) k (W/m-K), (7) alpha (m^2/s), (8) Pr

airProps(1, 1) = 100: airProps(1, 2) = 3.5562: airProps(1, 3) = 1.032:
airProps(1, 4) = 0.00000711: airProps(1, 5) = 0.000002:
airProps(1, 6) = 0.00934: airProps(1, 7) = 0.00000254: airProps(1, 8) = 0.786
airProps(2, 1) = 150: airProps(2, 2) = 2.3364: airProps(2, 3) = 1.012:
airProps(2, 4) = 0.00001034: airProps(2, 5) = 0.000004426:
airProps(2, 6) = 0.0138: airProps(2, 7) = 0.00000584: airProps(2, 8) = 0.758
airProps(3, 1) = 200: airProps(3, 2) = 1.7458: airProps(3, 3) = 1.007:
airProps(3, 4) = 0.00001325: airProps(3, 5) = 0.00000759:
airProps(3, 6) = 0.0181: airProps(3, 7) = 0.0000103: airProps(3, 8) = 0.737
airProps(4, 1) = 250: airProps(4, 2) = 1.3947: airProps(4, 3) = 1.006:
airProps(4, 4) = 0.00001596: airProps(4, 5) = 0.00001144:
airProps(4, 6) = 0.0223: airProps(4, 7) = 0.0000159: airProps(4, 8) = 0.72
airProps(5, 1) = 300: airProps(5, 2) = 1.1614: airProps(5, 3) = 1.007:
airProps(5, 4) = 0.00001846: airProps(5, 5) = 0.00001589:
airProps(5, 6) = 0.0263: airProps(5, 7) = 0.0000225: airProps(5, 8) = 0.707
airProps(6, 1) = 350: airProps(6, 2) = 0.995: airProps(6, 3) = 1.009:
airProps(6, 4) = 0.00002082: airProps(6, 5) = 0.00002092:
airProps(6, 6) = 0.03: airProps(6, 7) = 0.0000299: airProps(6, 8) = 0.7
airProps(7, 1) = 400: airProps(7, 2) = 0.8711: airProps(7, 3) = 1.014:
airProps(7, 4) = 0.00002301: airProps(7, 5) = 0.00002641:
airProps(7, 6) = 0.0338: airProps(7, 7) = 0.0000383: airProps(7, 8) = 0.69
```

```
airProps(8, 1) = 450: airProps(8, 2) = 0.774: airProps(8, 3) = 1.021:
airProps(8, 4) = 0.00002507: airProps(8, 5) = 0.00003239:
airProps(8, 6) = 0.0373: airProps(8, 7) = 0.0000472: airProps(8, 8) = 0.686
airProps(9, 1) = 500: airProps(9, 2) = 0.6964: airProps(9, 3) = 1.03:
airProps(9, 4) = 0.00002701: airProps(9, 5) = 0.00003879:
airProps(9, 6) = 0.0407: airProps(9, 7) = 0.0000567: airProps(9, 8) = 0.684
airProps(10, 1) = 550: airProps(10, 2) = 0.6329: airProps(10, 3) = 1.04:
airProps(10, 4) = 0.00002884: airProps(10, 5) = 0.00004557:
airProps(10, 6) = 0.0439: airProps(10, 7) = 0.0000667: airProps(10, 8) = 0.683

    getAirThermophysicalProps = airProps
End Function

Private Function generateMachineTemps(Tamb As Double, stepSize As Double) As Variant
    Dim temps() As Double
    Dim T As Double
    Dim n As Long

    T = Tamb + 10
    n = 0

    Do While T <= 393 ' Stop below 120 degrees.
        ReDim Preserve temps(0 To n)
        temps(n) = T
        n = n + 1
        T = T + stepSize
    Loop

    generateMachineTemps = temps
End Function

Private Function getSurface(name As String, part As SldWorks.ModelDoc2) As Variant

    Dim Surface As Variant
    Dim errcode As Long
    Dim SurfaceString As String

    Select Case name
    Case "mid"
        SurfaceString =
"UEYAAAMAAAD//v8hRQBuAGMAbABvAHMAdQByAGUAIABJAGQAZQBhAHQAaQBvAG4AXwAxAC0A" & _
"MQBAAEkAZABlAGEAdABpAG8AbgBfADEABAAAABAAAAABAAAAAQAAABgAAAD//wEACwBtb0Zh" & _
"Y2VSZWZfYwEAAAAAAAAABQAAAAADAAAAAAAAfcOUJa1JslR9w5QlrUmyVAAA//8BABIAbW9Q" & _
"TGluZVN1cmZJZFJlcF9jAAD//wEABgBtb0ZSX2P//wEADQBtb0V4dE9iamVjdF9j//8BABEA" & _
"bW9DU3RyaW5nSGFuZGxlX2P//v9LQwA6AFwAVQBzAGUAcgBzAFwAUwBpAG0AbwBuAFwARABv" & _
"AGMAdQBtAGUAbgB0AHMAXABTAG8AbABpAGQAVwBvAHIAawBzACAAUwBpAG0AdQBsAGEAdABp" & _
"AG8AbgBzAFwARQBuAGMAbABvAHMAdQByAGUAIABJAGQAZQBhAHQAaQBvAG4AXwAxAC4AUwBM" & _
"AEQAUABSAFQACYD//v8URQBuAGMAbABvAHMAdQByAGUAIABJAGQAZQBhAHQAaQBvAG4AXwAx" & _
"AAIAALdY5mT//v8A//7/AP/+/wAACAAAAAABAAAAAAAAGwFAAD//v8LVABoAGUAbQBhAGwA" & _
```

```
"IABTAGkAbQBzAAUAAAAAAAAAAAAAAAAAAAADjBAAANObTaP//AQAUAG1vRW5kRmFjZVN1cmZJ" & _
"ZFJlcF9jAAAFgAgAIgAAAM1Y5mQBAAAAAAAAAAIAAAD//wEAEgBtb1BMaW5lUHJvaklkUmVw" & _
"X2MAAAWACADjBAAANObTaAyAAAAFgAgAIgAAAM1Y5mQBAAAAAAAAAAEAAAAAAAA+AAAAFgAgA" & _
"4wQAADTm02gMgAAABYAIACIAAADNWOZkAQAAAAAAAAAADAAAAAAAAAAAAAAAAAFBGAAAAAAAA" & _
"AAAAAA=="

    Case "RTD"
        SurfaceString =
"UEYAAAMAAAD//v8hRQBuAGMAbABvAHMAdQByAGUAIABJAGQAZQBhAHQAaQBvAG4AXwAxAC0A" & _
"MQBAAEkAZABlAGEAdABpAG8AbgBfADEABAAAABAAAAABAAAAAQAAABgAAAD//wEACwBtb0Zh" & _
"Y2VSZWZfYwEAAAAAAAAAACQAAAAADAAAAAAAAAAfcOUJa1JslR9w5QlrUmyVAAA//8BABgAbW9F" & _
"bmRGYWNlM0ludFN1cmZJZFJlcF9jAAD//wEABgBtb0ZSX2P//wEADQBtb0V4dE9iamVjdF9j" & _
"//8BABEAbW9DU3RyaW5nSGFuZGxlX2P//v9LQwA6AFwAVQBzAGUAcgBzAFwAUwBpAG0AbwBu" & _
"AFwARABvAGMAdQBtAGUAbgB0AHMAXABTAG8AbABpAGQAVwBvAHIAawBzACAAUwBpAG0AdQBs" & _
"AGEAdABpAG8AbgBzAFwARQBuAGMAbABvAHMAdQByAGUAIABJAGQAZQBhAHQAaQBvAG4AXwAx" & _
"AC4AUwBMAEQAUABSAFQACYD//v8URQBuAGMAbABvAHMAdQByAGUAIABJAGQAZQBhAHQAaQBv" & _
"AG4AXwAxAAIAALdY5mT//v8A//7/AP/+/wAACAAAAAABAAAAAAAAAGwFAAD//v8LVABoAGUA" & _
"bQBhAGwAIABTAGkAbQBzAAUAAAAAAAAAAAAAAAAAAAAMBAAAnqtVZQEAAAAAAAAAAAAAAAP//" & _
"AQAbAG1vRnJvbVNrdEVudDNJbnRTdXJmSWRSZXBfYwAABYAIAAwEAACeq1VlAQAAAP////8A" & _
"AAAA//8BABMAbW9GaWxsZXRTdXJmSWRSZXBfYwAABYAIAA4EAAChrFVlAQAAAAyAAAAFgAgA" & _
"DAQAAJ6rVWUCAAAA/////wAAAAAMgAAABYAIAAwEAACeq1VlBAAAAP////8AAAAADIAAAAWA" & _
"CABuAAAAumTmZAQAAAD/////AAAAAAyAAAAFgAgAdgMAAOW5MmUfAAAA/////wAAAAAMgAAA" & _
"BYAIAHYDAADluTJlBgAAAP////8AAAAAAAAAAAAAAAAAAFBGAAAAAAAAAAAAAAA=="
    End Select

    Surface = Base64ToArray(SurfaceString)
    getSurface = Array(part.Extension.GetObjectByPersistReference3(Surface, errcode))

End Function

Private Function WriteResultsToCSV(Tamb As Double, Tmachine As Double, TGuess As
Double, Tmid As Double, Trtd As Double, filePath As String)
    Dim fso As Object, f As Object
    Dim i As Long

    ' Create FileSystemObject
    Set fso = CreateObject("Scripting.FileSystemObject")
    ' If file does not exist, create and add header
    If Not fso.FileExists(filePath) Then
        Set f = fso.CreateTextFile(filePath, True, True)
        ' Write header row
        f.WriteLine "Tamb (K),Tmachine (K), TGuess (K), Tmid (K), Trtd (K)"
        f.Close
    End If

    Set f = fso.OpenTextFile(filePath, 8, True, -1) ' 8 = ForAppending, -1 for Unicode
    f.WriteLine Format(Tamb, "0.00") & ", " & _
        Format(Tmachine, "0.00") & ", " & _
        Format(TGuess, "0.00") & ", " & _
```

```
        Format(Tmid, "0.00") & ", " & _
        Format(Trtd, "0.00")
    f.Close

    Debug.Print "Result written to " & filePath
End Function

' Gets the average of a 1-D array of size N that is built
' as follows: {node0, value0, node1, value1, ... , nodeN, valueN}.
Private Function getArrayAverage(list As Variant) As Double
    Dim i As Long
    Dim count As Long
    Dim sum As Double

    For i = LBound(list) To UBound(list) Step 2
        sum = sum + list(i + 1)
        count = count + 1
    Next i

    If count > 0 Then
        getArrayAverage = sum / count
    Else
        getArrayAverage = 0
    End If

End Function

' This function provides a clean way to refer to surfaces, using base 64 decoding.
' The author claims no copyright to this code.
Private Function Base64ToArray(base64 As String) As Variant
'*************************************************************************************
    ' Copyright(C) 2025 Xarial Pty Limited'
    ' Reference: https://www.codestack.net/solidworks-api/document/tracking-
objects/persist-references/ '
    ' License: https://www.codestack.net/license/'
'*************************************************************************************

    Dim xmlDoc As Object
    Dim xmlNode As Object

    Set xmlDoc = CreateObject("MSXML2.DOMDocument")
    Set xmlNode = xmlDoc.createElement("b64")

    xmlNode.DataType = "bin.base64"
    xmlNode.Text = base64

    Base64ToArray = xmlNode.nodeTypedValue

End Function
```