



# Proof of Concept Design of an Acoustic Beamforming Microphone Array for the Early Detection of Defects

Final Year Project Report - MECH4841 Part B

October 2019

Samuel Willis<sup>1</sup>

---

<sup>1</sup> Student of Mechanical Engineering,  
The University of Newcastle, Callaghan, NSW 2308, AUSTRALIA  
Student Number: 3256767  
E-mail: [s.willis@uon.edu.au](mailto:s.willis@uon.edu.au)

## **Abstract**

This report entails my current work and my plans for future work in my final year project towards designing and building a working beamforming acoustic array for the purpose of condition monitoring of machinery. The use of acoustic arrays in the condition monitoring industry will be able to improve the early detection of defects and be much easier to setup and monitor.

Currently I have designed an acoustic array consisting of 12 MEMS Microphones that has been tested and detects and locates a sound source successfully, these results have been verified through simulation of the expected response to the input signal and the array's expected frequency response to a signal of that frequency.

In the future I wish to improve on the current algorithm and investigate new ones for hopes to improve the resolution and accuracy of the array, improve the geometry of the array and increase the number of microphones, and perform more thorough and scientific testing as well as some real-world testing.

## Contents

<b>1. Introduction</b>	<b>8</b>
<b>2. Literature Review</b>	<b>9</b>
2.1. Acoustic Beamforming . . . . .	9
2.2. Digital Signal Processing . . . . .	10
2.2.1. Fourier Transform . . . . .	12
2.2.2. Delay-and-Sum Beamformer . . . . .	14
2.2.3. Frequency-Domain Beamformer . . . . .	15
2.2.4. MUSIC Algorithm . . . . .	16
2.3. Condition Monitoring . . . . .	17
<b>3. Aims and Objectives</b>	<b>18</b>
3.1. Problem . . . . .	18
3.2. Use of Microphone Arrays for Condition Monitoring . . . . .	18
3.3. Project Aim . . . . .	18
<b>4. Software Research and Development</b>	<b>20</b>
4.1. Delay-and-Sum Algorithm . . . . .	20
4.1.1. LabVIEW Implementation . . . . .	22
4.2. MUSIC Algorithm . . . . .	24
4.2.1. LabVIEW Implementation . . . . .	26
4.3. Simulation Package . . . . .	29
4.3.1. Array Generator . . . . .	29
4.3.2. Delay Calculations . . . . .	30
4.3.3. Spatial Signal Generator . . . . .	31
4.3.4. File Playback . . . . .	33
4.3.5. Simulation Testing and Validation . . . . .	33
4.3.6. Algorithm Simulation Comparison . . . . .	34
<b>5. Hardware Research and Development</b>	<b>37</b>
5.1. 2D Frequency Response . . . . .	37
5.2. 3D Frequency Response . . . . .	39
5.3. Geometry Design . . . . .	41
5.3.1. First Design . . . . .	42
5.3.2. Second Design . . . . .	44
5.3.3. Design Comparison . . . . .	50
5.4. Microphones . . . . .	51
5.5. Data Acquisition . . . . .	51
5.5.1. First Array . . . . .	51
5.5.2. Second Array . . . . .	52
<b>6. Application to Machine Defects</b>	<b>55</b>

<b>7. Testing</b>	<b>57</b>
7.1. Early Testing of First Design . . . . .	57
7.2. Final Design Testing . . . . .	59
7.2.1. Design Verification . . . . .	59
7.2.2. Algorithm Comparison . . . . .	60
7.2.3. Defect Detection . . . . .	61
7.3. Site Testing . . . . .	63
<b>8. Recommendations</b>	<b>64</b>
<b>9. Conclusion</b>	<b>65</b>
<b>A. Simulation Figures</b>	<b>69</b>
<b>B. Testing Figure</b>	<b>71</b>

## List of Tables

1. Array Time Delay Comparisons . . . . .	60
---	----

## List of Figures

1. Microphone Array Designs (Prime and Doolan, 2013). . . . .	10
2. Nyquist Rate Visualised (National-Instruments, n.d.) . . . . .	11
3. Fourier Transform Visualization. . . . .	13
4. Delay-and-Sum Beamforming Diagram (Greensted, 2019). . . . .	15
5. MUSIC and Delay and Sum Comparison (Näsholm, 2012). . . . .	16
6. Uniform Linear Array of Microphone Array (Benesty et al., 2008). . . . .	21
7. LabVIEW Delay Block Diagram Data Flows Left to Right . . . . .	23
8. LabVIEW MUSIC Implementation Block Diagram Pt.1 . . . . .	27
9. LabVIEW MUSIC Implementation Block Diagram Pt.2 . . . . .	28
10. Array Generator LabVIEW Block Diagram . . . . .	29
11. Array Generator LabVIEW Front Panel User Interface . . . . .	30
12. Microphone Relative Delay Generator LabVIEW Block Diagram . . . . .	31
13. Spatial Signal Generator LabVIEW Block Diagram . . . . .	32
14. Spatial Signal Generator LabVIEW Front Panel User Interface . . . . .	33
15. File Playback LabVIEW Front Panel User Interface . . . . .	33
16. Comparison of Simulation Results to Expected Results . . . . .	34
17. DAS and MUSIC Algorithms compared at 1863Hz . . . . .	35
18. DAS and MUSIC Algorithms compared at 1863Hz Source Separation . . . . .	36
19. 2D Frequency Response . . . . .	37
20. 2D Frequency Response Steered to 130° . . . . .	37
21. 2D Frequency Response over Varying Frequencies . . . . .	38
22. 3D Frequency Responses . . . . .	40
23. 3D Frequency Response Heatmap in Decibel . . . . .	40
24. Chosen Underbrink Array . . . . .	43
25. Underbrink Array 3D Responses . . . . .	44
26. Initial Microphone Array . . . . .	44
27. Series of tested microphone array geometries . . . . .	46
28. 3D Frequency Response Results . . . . .	47
29. Final Array Azimuth Frequency Sweep Beam Patterns . . . . .	48
30. Final Array Elevation Frequency Sweep Beam Patterns . . . . .	48
31. Final Array . . . . .	50
32. Array Comparison 3D Responses . . . . .	50
33. NI cDAQ with a single NI 9232 Installed and Wired Up . . . . .	52
34. Final Microphone Array with 24 Microphones . . . . .	53
35. Final Microphone Array Wiring . . . . .	54
36. Comparison of Frequency Responses at 3kHz . . . . .	54
37. Result of Whistle testing . . . . .	57

38.	Whistle test 3D Frequency Response . . . . .	58
39.	Whistle test Simulation . . . . .	58
40.	Array Verification Test . . . . .	60
41.	Real Algorithm Comparison to 1871Hz Signal . . . . .	61
42.	Microphone Signals . . . . .	62
43.	Zoomed Microphone Signals . . . . .	62
44.	Differentiate signal testing results . . . . .	63

## Nomenclature

### Latin Alphabet

$c$	Speed of sound
$D$	Array diameter (Aperture)
$f_s$	Sample rate
$N$	Number of microphones
$P_{\theta,\phi}$	Power at a point
$r$	Signal source distance from origin of array
$r_0$	Minimum array radius
$r_{max}$	Maximum array radius
$s(t)$	Received signal
$t$	Time
$v$	Underbrink Spiral angle
$x$	x-coordinate location of microphone
$y$	y-coordinate location of microphone

### Greek Alphabet

$\lambda$	Wavelength
$\tau$	Delay
$\tau_s$	Delay in samples
$\tau_\varphi$	Phase delay
$\theta$	Azimuth angle
$\phi$	Elevation angle
$\varphi_A$	Steered azimuth angle
$\varphi_E$	Steered elevation angle

## 1. Introduction

Beamforming is a technique where an array of sensors can be steered towards a particular direction either for transmission of signals or for the locating of signal sources (Shahbazpanahi and Jing, 2018).

The cocktail party effect, a phenomenon that relies on the human ability of being able to focus on one particular sound or voice when there are multiple other sources of noise, competing sources and distractions. This was first investigated in 1953 by Colin Cherry, what he referred to as the cocktail party problem, asking the question, can we replicate the ability of the cocktail party effect with man-made systems? This is the problem beamforming microphone arrays aims to solve (Benesty et al., 2008).

For the purpose of this project beamforming is applied with an array of microphones being used to locate a sound source. Beamforming acoustic arrays are not a new concept, they have been used extensively in aerocoustics for the purpose of analysing aerofoils of aeroplanes and wind turbines for characterising and measuring generated noise as well as applications in finding the source of gas leaks, source of reverberation and general noise in studio environments, and finding squeaks and rattles in the automotive industry. One notable array is from a collaboration between Hyundai Motor Group and SM Instruments who together developed a small real-time acoustic array for the purpose of finding squeaks and rattles in Hyundai car's this array is available for purchase but costs upwards of \$65'000, though it's not suited for the purpose of early detection of defects on machinery especially as it can only beamform between the frequencies of 2kHz and 8kHz (SM-Instruments, n.d.). The purpose of this project is to design an acoustic array to locate the sources of potential early defects in machinery for condition monitoring and analysis of plant.

## 2. Literature Review

### 2.1. Acoustic Beamforming

Through the digital processing of multiple signals received at microphones at known locations an estimated direction of arrival and location of a signal of interest can be found, this is the basic concept of acoustic beamforming. This process can be performed in either the time-domain or the frequency-domain by using different algorithms and processing techniques. One of the core assumptions for all algorithms is whether the audio waves are treated as far-field or near-field. The far-field assumption treats the sound waves as a planar wave-front while the near-field assumption treats sound waves as a spherical wave front. Sound waves can be assumed to be far-field when they have propagated such that Equation 2.1 is true (McCowan, 2001), before this they are treated as near-field, this is due to point sources producing a spherical wave-front and as it propagates becoming planar (Yan, 2005).

$$r > \frac{2D^2}{\lambda} \quad (2.1)$$

add a calc somewhere in report

where,  $r$  = Signal source distance from array origin  
 $D$  = Array Diameter  
 $\lambda$  = Wavelength (see Equation 2.2)

For the purpose of acoustic beamforming an array of microphones are used where the exact coordinate of each and every microphones is known and usually constant, the received signal at each sensor can then be synchronised, digitized and processed through a choice of algorithms and output the spatial location of a source of interest or audio from that point. This process involves using the known propagation speed of sound waves in air and the time difference of arrival for the signals at each one of the sensors.

The geometry of the microphone array and the quantity of sensors becomes important for the accuracy and resolution of the estimated signal location. The choice of geometry arrangement whether it be in a grid, straight line or any other design can cause problems at particular frequencies due to spatial aliasing this will be further explained in Section 2.2, this requires careful consideration when designing the geometry of the microphone array, thus the frequencies of interest must be defined beforehand to determine the required microphone array design and size. These frequencies are particularly important

for choosing the diameter of the microphone array (also known as the aperture) and the microphone spacing for grid and straight line arrays.

Considerable amounts research and design have been achieved in the area of microphone array geometries, due to this several different array designs have been devised in attempt to increase the resolution and accuracy whilst avoiding spatial aliasing and minimising side-lobe levels as is discussed further in Section 5.1. Some of the best performing and most popular array designs are shown in Figure 1. These are regarded as some of the best performing due to them all having a high resolution with a high signal to noise ratio over a broad range of frequencies.

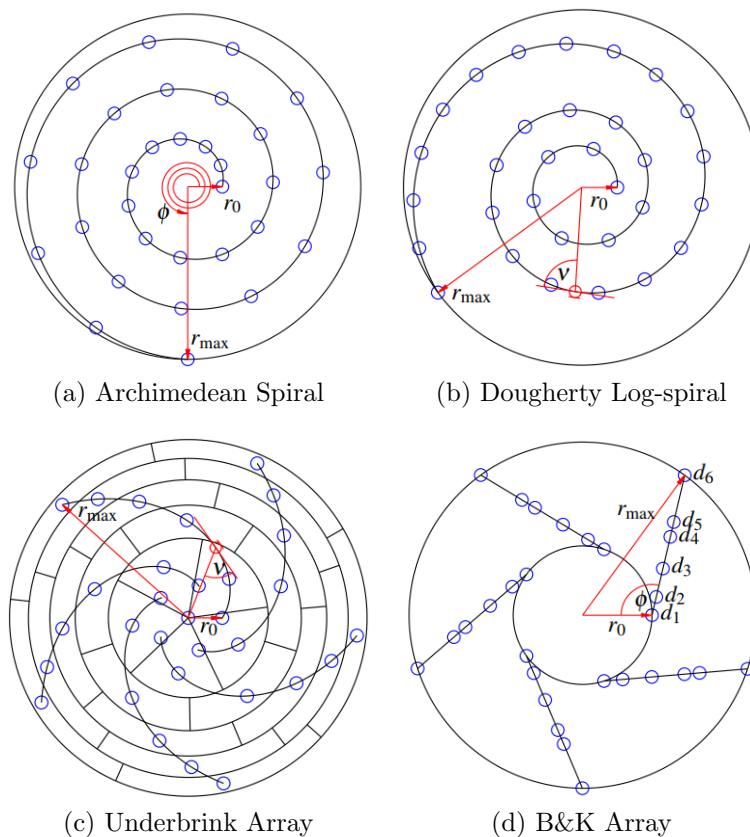


Figure 1: Microphone Array Designs (Prime and Doolan, 2013).

## 2.2. Digital Signal Processing

**what is an analog signal** The processing of digital signals is the most important factor when it comes to beamforming, as an array of only two microphones have the potential to perform beamforming to a certain degree of accuracy, whilst improving the signal

condition, signal processing, and choice of algorithm, the resulting accuracy can be improved.

Sampling of an analog signal is one of the core processes of digital signal processing, this is the process of discretization allowing the signal to be defined by a finite set of values. When sampling the signal it must be ensured that it is being sampled at a high enough rate to avoid aliasing, this is distortion that occurs when the reconstructed signal differs from the original due to not sampling at a high enough rate. This occurs when a signal is sampled at less than half of its frequency, this required sampling rate is known as the Nyquist Rate. When sampled below the Nyquist Rate the audio waves are not able to be recreated accurately (Benesty et al., 2008). This concept is best explained with a visualisation, Figure 2 shows one sine wave sampled at three different rates, from this we can see how if the sampling rate is equal to the frequency of the signal it results in a constant amplitude whilst sampling at other frequencies below the Nyquist Rate will result in completely different signals with varying frequencies and amplitudes.

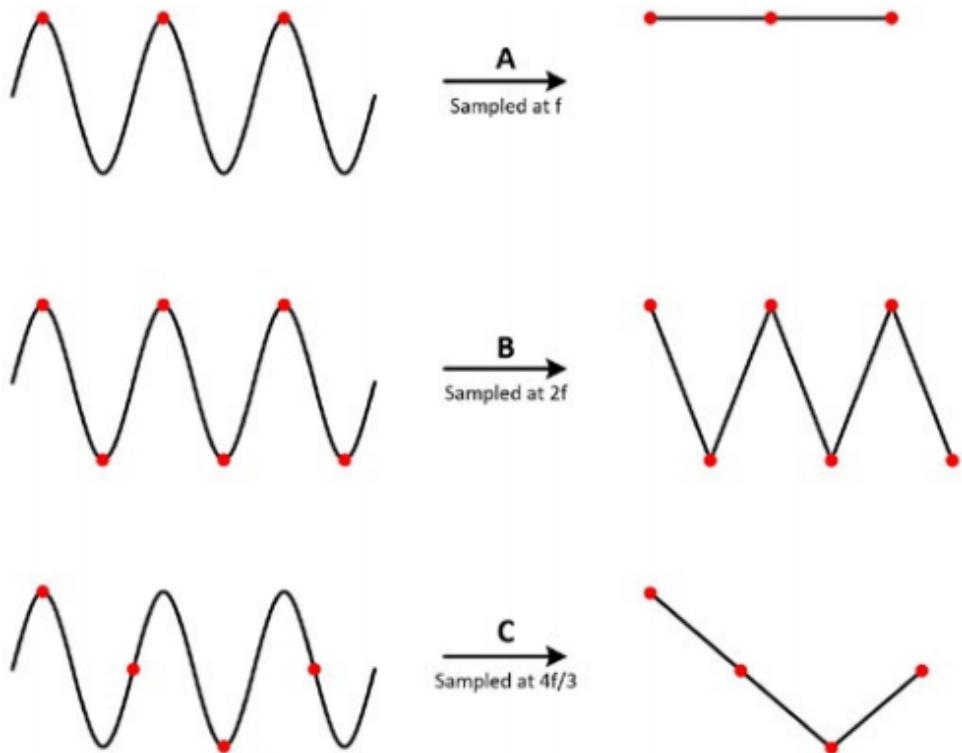


Figure 2: Nyquist Rate Visualised (National-Instruments, n.d.)

The concepts of a Nyquist rate and aliasing is directly related to Spatial Aliasing, this must be taken into account when designing microphone arrays. Spatial Aliasing occurs

when the location of the signal source cannot be determined due to the spacing of sensors in the microphone array relative to the wavelength of the signal of interest. This is similar to sampling and normal aliasing but rather than the points being sampled on the sine wave as being the problem its the spatial location of the microphones hence "spatial" aliasing. For example a 1000Hz signal will have a wavelength of 34.3cm shown by Equation 2.2, therefore if you have sensors spaced this distance apart it's not possible to tell exactly what direction the 1000Hz signal is originating from, this can be explained as if you were to picture a signal at its peak at the first microphone it will also be at the peak for the second microphone, and therefore appear exactly the same, this is because they are exactly one wavelength apart, very similar to what is shown in Figure 2. For sensor arrays in a uniform line or grid the sensors must have a spacing of less than half the wavelength of the maximum frequency of interest to avoid spatial aliasing. An appropriately sized aperture is another important property of array geometry, the aperture is the diameter of the array and a larger aperture is required to beamform low frequency signals (Amin et al., 2008). **why**

$$\lambda = \frac{c}{f} \quad (2.2)$$

where,  $c$  = Speed of sound (see Equation 4.3)  
 $f$  = Frequency

### 2.2.1. Fourier Transform

The Fourier transform is a vital part of digital signal processing, especially for analysis of vibration signals and condition monitoring, as well as a requirement for beamforming in the frequency-domain. A Fourier Transform is used to change a signal from the time-domain to the frequency-domain. A signal in the time domain consists of a series of amplitudes with corresponding time-stamps, Fourier transforming will result in the signal being broken up into the individual frequencies that make up that signal, where each frequency is essentially a sinusoidal wave that have an individual amplitude and phase (Schaedler, 2017). Figure 3 shows how a Fourier transform will break down a square wave signal, the red pane shows the original signal in the time-domain, the blue pane on the right shows the signal Fourier transformed spectrum, and in the middle we can see the actual individual waveforms the signal can be broken down into, this visualization allows us to see the frequency composition and the differing phases and amplitudes for

each frequency.

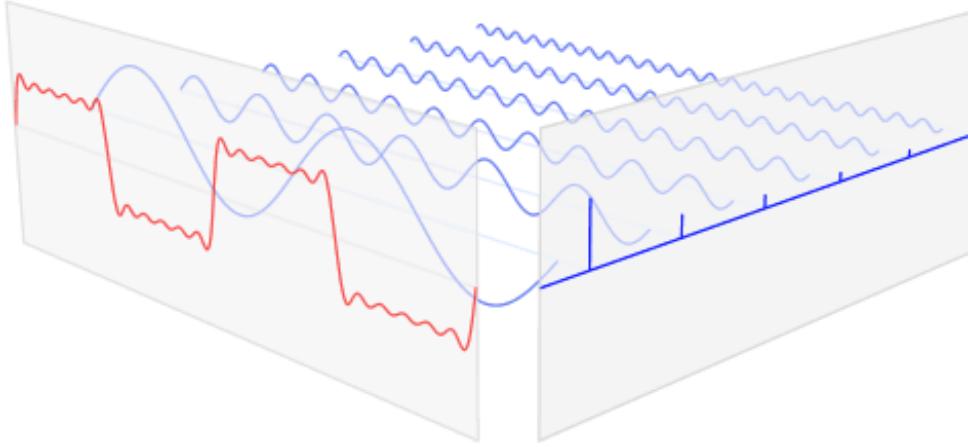


Figure 3: Fourier Transform Visualization.

The discrete Fourier transform is defined by Equation 2.3, the core operation of the Fourier transform is the dot product. Performing a dot product of two equal length signals would be the equivalent multiplying them element by element and summing the result. The dot product result can be thought of as similarity, as the dot product of a sinusoidal wave and cosine wave of the same frequency will be 0, whilst two sinusoidal waves of the same frequency will be a maximum and two sinusoidal waves 180 degrees out of phase will be a negative maximum (Schaedler, 2017).

$$DFT[k] = \sum_{n=0}^{N-1} x[n] \cdot (\cos(\varphi) - \sin(\varphi)i) \quad (2.3)$$

$$\text{Where } \varphi = k \frac{n}{N} 2\pi$$

When performing a fourier transform on a finite length signal the resolution of the fourier transform depends on the length and sampling rate of the digitized signal, due to this we have frequency bins which is the resolution of the fourier transform and therefore frequency spectrum, if the spectrum begins at 0 and the frequency bin size is 4Hz this would mean all of the frequencies from 0 to 4Hz would essentially be summed into the single bin at 4Hz, and frequencies from 4 to 8Hz would be in the 8Hz bin and so on. From Eqn. 2.3 we define  $N$  as the number of samples in a signal and  $k$  as a frequency bin, as the output of a discrete Fourier Transform is a series of frequency bins. Each one of the  $k$  output frequency bins corresponds to a particular frequency that can be calculated with

Equation 2.4, the result when  $k = 1$  is the size of the frequency bins.

$$f_k = k \times \frac{\text{sample rate}}{N} \quad (2.4)$$

As can be determined by Equation 2.4 the resolution of the discrete Fourier Transform relies on the amount of time of the signal is processed, 1 second signal will have 1Hz frequency bins whilst 0.25 second signal will have 4Hz frequency bins and hence the frequencies within the 4Hz range will be effectively summed together into each frequency bin.

### 2.2.2. Delay-and-Sum Beamformer

The Delay-and-Sum algorithm is the classical method of beamforming in the time-domain and one of the most robust and well used. As the location of each sensor in the array is known and we know that sound waves propagate at the speed of sound, we can determine that a distant sound source of will arrive at each sensor at a slightly different time this is the time difference of arrival that is calculated with the array geometry and is the core of all methods beamforming.

The name of this algorithm comes from the processes undertaken, consisting of two main steps; delaying and summing. First the signal from each sensor is individually shifted by a time delay that corresponds to a particular spatial location relative to the sensor, the signals are then summed to receive the resulting waveform from that location (Greensted, 2019) this process is digitally steering the microphone array. Once the resulting signal is divided through by the number of sensors we have attenuated and amplified the signal at the angle of interest and dulled signals from all other locations as they will not interfere constructively. This concept is visually demonstrated in Figure 4, showing how the angle of interest is attenuated in comparison to all other angles being dulled.

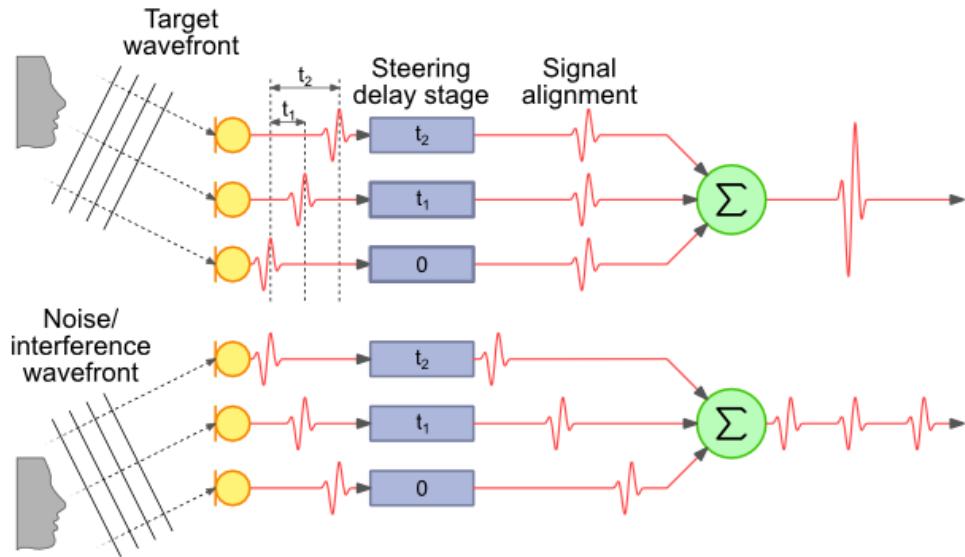


Figure 4: Delay-and-Sum Beamforming Diagram (Greensted, 2019).

### 2.2.3. Frequency-Domain Beamformer

Delay-and-Sum beamforming is performed in the time-domain as the signal is a function of amplitude and time, there are other algorithms that are used in the time-domain based off of delay-and-sum theory, though there are several other algorithms that process in the frequency-domain. Beamforming in the frequency-domain involves analysing the signals by their amplitudes of individual frequency bins without respect to time but to the signal as a whole and therefore must be over a set period of time.

A conventional frequency-domain beamformer requires the incoming signal to be in the Fourier-domain which is achieved by performing a Fourier transform as described in Section 2.2. Compared to the time-domain delay-and-sum beamformer where the signal is delayed by shifting/delaying the times of the incoming signals the frequency-domain differs, the signal is delayed by adding a phase delay to the individual frequency bins that are a result of the Fourier transform, this theoretically achieves the same result as the delay-and-sum beamforming but has the advantage of being far less computationally intensive and potentially have far greater accuracy in results. (Krishnaveni et al., 2013).

#### 2.2.4. MUSIC Algorithm

The MULTiple SIgnal Classification algorithm (MUSIC) is a subspace frequency-domain method of beamforming relying upon an eigen-decomposition of the spatial co-variance matrix of the received signal data. The spatial co-variance matrix is decomposed into it's eigenvalues and vectors that are then sorted by eigenvalue and separated into noise and signal subspaces. With the use of array steering vectors based on phase rather than time, the direction of arrival can be estimated as the steering vector is theoretically orthogonal to the noise subspace obtained from eigenvalue decomposition and therefore when multiplied together results in 0 (Paulraj et al., 1993) (MathWorks, 2020), when taking the reciprocal of this it will tend towards infinity when at the angle of the sound source similar to what is seen in Figure 5. This results in a high resolution algorithm for estimating the direction of arrival of either narrowband or wideband signals and far less computationally heavy compared to the traditional Delay and Sum beamforming algorithms. Below in Figure 5 a comparison of the two algorithms is shown for the purpose of demonstrating the additional resolution and accuracy gained.

What is subspace? covariance?

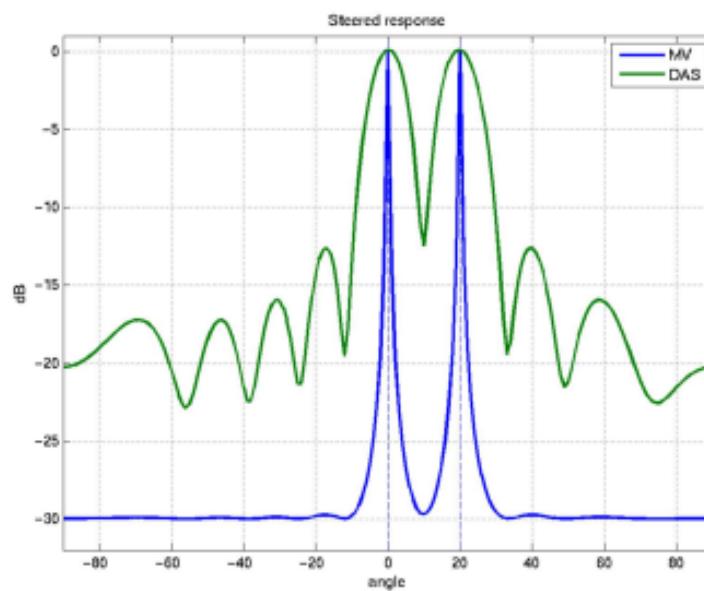


Figure 5: MUSIC and Delay and Sum Comparison (Näsholm, 2012).

## 2.3. Condition Monitoring

”Condition monitoring is the continuous monitoring of rotating machines during process conditions. This ensure the optimal use of machines. Condition Monitoring supports predictive maintenance, which is more efficient than preventive maintenance. Condition Monitoring provides insight in performance, wear and mechanical risks of machines. This helps to prevent unnecessary and unplanned downtime.” (Bakker, n.d.).

Preventive maintenance of machinery is the regular replacement of components based on a rigid schedule and the typical operating life expectations resulting in parts being replaced or lubricated more often than necessary to meet these requirements resulting in inefficient use of valuable time. With the introduction of electronic instrumentation allowing vibration analysis and potentially acoustic analysis these preventive maintenance programs can be easily converted to predictive maintenance programs. Predictive maintenance uses data from measurements and other analysis performed on a regular basis, the concept is to predict when it is necessary to take the machine out of service for maintenance and/or repair rather than waiting for the machine to breakdown and require fixing, this leads to an effective and efficient maintenance program (Laws and Muszynska, 1987).

Condition Monitoring is part of the maintenance program though relates more to a particular machine. Condition Monitoring aims to define the current condition of a machine by comparing it to previously known conditions. Analysing the trends of this allows for the prediction of machine condition over time. Through the use of an acoustic array that could potentially be permanently setup the condition of the machine can be easily monitored and any changes or particular noises will be able to be analysed individually or monitoring could be done on a routine basis to analyse trends in the machinery sound-scape.

### **3. Aims and Objectives**

#### **3.1. Problem**

Currently the condition monitoring of rotating machinery is primarily performed using vibration, proximity and temperature sensors. This is generally performed on a schedule with routine data analysis for insight into any trends or patterns indicating machinery may need additional maintenance or repair. If this machine in the mean-time starts to make odd noises from an unknown location it can be a lengthy and expensive process to determine where it is coming from and why. With an acoustic array set up in-front of machinery it is possible to monitor a large area for any changes in sound or appearance of problem noises and therefore will be able to identify where the problem has arisen from, even further it will be possible to analyse the noise further and potentially diagnose the problem. This could potentially save a great amount of time and money especially if other techniques require machinery to be taken out of service for analysis.

#### **3.2. Use of Microphone Arrays for Condition Monitoring**

Currently there are few acoustic arrays that have been built for the purpose of condition monitoring, they have been shown to be a powerful tool with serious potential in an article on the analysis of electric motors with an acoustic camera (Orman and Pinto, 2013). There is one acoustic technology company "Squarehead Technology" that offer condition monitoring acoustic arrays as one of their services in the USA and Norway but besides this there is a large gap in the market of acoustic arrays purpose built for condition monitoring especially for outright purchase. With the ability to choose how the algorithms are processing the incoming data it will give the ability to pinpoint the locations of individual machine defects through the known fault frequencies and harmonics.

#### **3.3. Project Aim**

By the completion of my final year project I aim to have a completely functioning acoustic array. This array should be able to determine when a noise of interest is present, locate it spatially and return a clean signal for further analysis. The current milestones are as follows:

- Learn about audio signal processing and developed a greater understanding.

- Develop an understanding of condition monitoring, vibration analysis and its application.
- Learn LabVIEW and Develop a greater understanding of data acquisition and software development.
- Implement a working algorithms in both the time and frequency domain and show it working in simulation.
- Build an acoustic array using NI Hardware and MEMS Microphone.
- Show a working acoustic array able to locate and detect sounds.
- Show the acoustic array picking particular sounds out of noisy environments.
- Extract the signal from point of interest for further analysis.
- Test the acoustic array on site on operating machinery.

With a working acoustic array it will allow for the optimisation of the maintenance and monitoring for a large range of machinery improving the financial costs, time and safety as well as analysis for early detection of defects and locating of unknown sources of noises.

## 4. Software Research and Development

The software chosen for the development environment and data acquisition is National Instruments LabVIEW due to the ease of use with currently available hardware and the great potential for expansion of sensors. LabVIEW also makes the programming of field programmable gate array relatively simple which can potentially make impressive speed improvements to the algorithm.

LabVIEW is short for Laboratory Virtual Instrument Engineering Workbench, a programming environment in which you create programs graphically connecting function blocks with wires for data flow. LabVIEW features a graphical user-interface making it simple for simulations, presentation of ideas and general programming (Travis and Kring, 2006). With the inbuilt digital signal processing and data acquisition libraries this makes LabVIEW incredibly useful for the purpose of developing an acoustic array.

### 4.1. Delay-and-Sum Algorithm

To begin a full investigation into acoustic beamforming arrays a complete understanding of the delay-and-sum beamformer is required. For explanation and understanding it is easiest to think of a uniform linear array of  $N$  microphones. Figure 6 demonstrates this, where  $s(t)$  is the unknown source in the far-field,  $\theta$  is the incoming incident angle,  $d$  is the spacing between sensors,  $v_n(t)$  is additive noise and interference,  $x_n(t)$  is the signal at each sensor, and  $y_n(t)$  the output of each of these sensors.

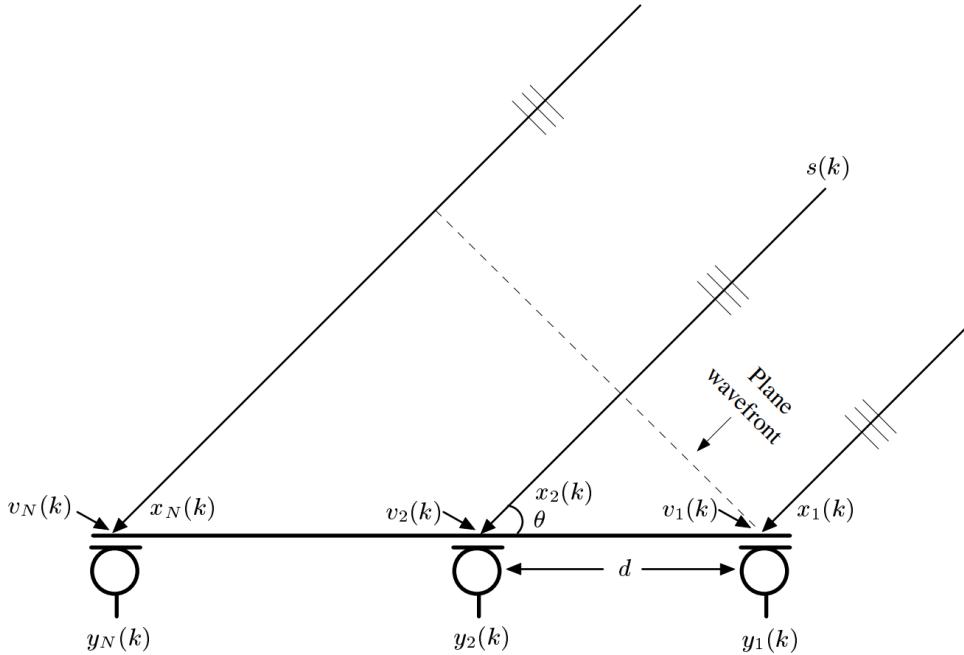


Figure 6: Uniform Linear Array of Microphone Array (Benesty et al., 2008).

As can be seen in the diagram, the path to each microphone has slightly different lengths, this is what causes the differing time delays and allows for beamforming. If we let  $s(t)$  be the received signal at the origin then an array of signals received at each sensor at time instant  $t$  is written as and  $s_n(t)$  will be the signal received at sensor  $n$ :

$$s(t) = [s(t - \tau_1), s(t - \tau_2), \dots, s(t - \tau_N)]^T \quad (4.1)$$

$$s_n(t) = s(t - \tau_n) \quad (4.2)$$

In Equation 4.1  $\tau$  is the time delay also known as the time difference of arrival. For arrays of any geometry this delay can be quite easily calculated using trigonometry, the speed of sound, and location of the sensors. It should always be taken into account that the speed of sound in air will change with the temperature,  $T$ , this relationship is shown in Equation 4.3.

$$c = 331.3 \times \sqrt{1 + \frac{T}{273.15}} \quad (4.3)$$

For the uniform linear array we can find the delay for each microphone using the respective  $x$  location relative to the origin at the centre of the microphone array and the Equation

4.4.

$$\tau = \frac{x \cdot \cos(\theta)}{c} \quad (4.4)$$

This can be modified for 2D planar arrays so that it takes into account the  $x$  and  $y$  location relative to the origin at the centre of the microphone array, by using Equation 4.5, this will allow the use of any array of microphones, including the arrays pictured in Figure 1.

$$\tau = \frac{x \cdot \cos(\theta) + y \cdot \sin(-\phi)}{c} \quad (4.5)$$

Both Equation 4.4 and 4.5 make the assumption of, the azimuth ( $\theta$ ) being the angle sweeping across horizontally in-front of the array ranging from  $0^\circ \rightarrow 180^\circ$  where  $90^\circ$  is at the origin and the elevation ( $\phi$ ) sweeping from the bottom at  $-90^\circ$  to the top at  $90^\circ$  vertically. This is used to digitally steer the beam over a virtual area in-front of the array to determine where a signal is coming from. The point in-front of the array where the signal has the highest power is the estimate of source location.

Using this delay-and-sum algorithm will also allow for the extracting of signals from a particular spatial location by applying the algorithm to a single spatial point rather. For the point of interest the channels are delayed accordingly, summed and then normalised as described above to return a single waveform for the location of interest.

#### 4.1.1. LabVIEW Implementation

For the real implementation of the delay-and-sum algorithm the delay equation needs a minor modification due to the digital sampling of signals. When a signal is sampled at 30kHz known as the sample rate ( $f_s$ ), each sample is approximately 0.033ms, now we have to talk in discrete time steps of this sample length rather than time, therefore the delay formula is re-written in Equation 4.6 to take this into account.

$$\tau_s = \frac{x \cdot \cos(\theta) + y \cdot \sin(-\phi)}{c} \cdot f_s \quad (4.6)$$

A problem arises using Equation 4.6 when the calculated delay for a particular channel is not an integer, due to the digital sampling the smallest delay is the sample length as it is not possible to delay by sub-sample time steps. When using a lower sampling rate

this problem can cause large inaccuracies, the easiest way to prevent this is to use the highest possible sample rate to minimise the error, otherwise, the incoming signal can be up-sampled by a preset factor using a Linear Up-sampler or Finite Impulse Response Filter, this up-sampling factor is taken into account with the delay by multiplying the original sampling rate by this same factor.

The process in LabVIEW for applying the delay to each signal was to find the largest and the smallest delay required for that particular point, for each channel the absolute of the smallest delay at that point plus the individual channel delay worth of zeroes are added to the beginning of the signal. Following this, the absolute value of the largest delay of samples are removed from this signal to ensure there are no zeroes left in the signal to keep integrity. The block diagram for this process is shown in Figure 7, the two rectangular structures are For Loops in LabVIEW, the outer iterates through a larger 2D array of delays and returns a 1D array of delays for a particular azimuth and elevation, the inner loop iterates through each channel applying the correct delay.

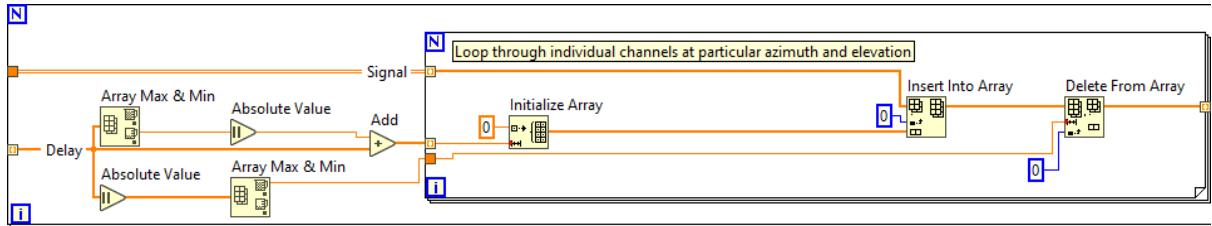


Figure 7: LabVIEW Delay Block Diagram Data Flows Left to Right

To produce a heat-map of the effective view of the acoustic array the array is digitally steered to each point on a virtual grid in-front of the array and evaluates the equivalent power of the signal at this point. For the delay-and-sum beamformer this process involves going through each azimuth and elevation giving a grid 180 by 180 points representing the area in-front of the array and delaying and summing at each point. The signals on acquisition are converted to an array of floating point values where each index is an individual sample of amplitude, then for each point on the array the signal is to be delayed by the calculated amount using Equation 4.6, the equivalent power at a point ( $P$ ) as a single number is calculated by Equation 4.7.

$$P_{(\theta,\phi)} = \sum_{t=0}^{\infty} \left( \frac{\sum_{n=1}^N x_n(t)}{N} \right)^2 \quad (4.7)$$

where,  $x_n(T) = x_n(t - \tau_{sn})$

Equation 4.7 determines the power at a single point by summing all the channels at that point resulting in a single signal wave that has been either constructively and destructively interfered then divided by the number of channels normalising the resulting wave, an example of this can be seen in Appendix A. Now the single interfered wave is squared element-by-element, this process makes all values positive but also attenuates the peaks of the signal compared to the troughs. Finally each amplitude value for every sample of the wave is summed resulting in a single value for that point. This process makes sure positions with larger amounts of constructively interfered signals that are closer to the signal location are attenuated and return a higher value allowing for a heat-map to be created with the estimated signal source location. So that the heatmap could be easily interpreted the output 2D array of power of the grid is scaled between 1 and -1 therefore if the value for a location is 1 it is the most likely estimation of the location of the signal of interest.

## 4.2. MUSIC Algorithm

This section has been adapted from Direction of Arrival Estimation (Adve, 2013) and DOA Estimation based on MUSIC algorithm (Tang, 2014). When approximating the direction of arrival of signals using the multiple signal classification algorithm the anticipated data for the  $M$  arriving signals are modeled as below in Eqn. 4.8.

$$\mathbf{x} = \mathbf{S}\alpha + \mathbf{n} \quad (4.8)$$

$\mathbf{S}$  is a  $N \times M$  matrix consisting of the array steering vectors for the  $M$  impinging signals.  $\alpha$  is a vector of the narrowband signals that consists of each signals amplitude and phase.  $\mathbf{n}$  is a vector consisting of the noise received at each of the microphones assumed to be a zero-mean Gaussian with a co-variance of  $\sigma^2 I$ .

$$\mathbf{S} = [\mathbf{s}(\theta_1, \phi_1) \quad \mathbf{s}(\theta_2, \phi_2) \quad \cdots \quad \mathbf{s}(\theta_M, \phi_M)] \quad (4.9)$$

$$\alpha = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_M]^T \quad (4.10)$$

Where:

$$\alpha = ae^{jb} \quad (4.11)$$

$$s(\theta, \phi) = e^{-j2\pi\tau(\theta, \phi)f} \quad (4.12)$$

An assumption is made that the impinging signals are uncorrelated, and therefore the estimated correlation of matrix of  $\mathbf{x}$  is as shown below, where  $E[ ]$  means the expected value and  $H$  is the notation used to indicate a hermitian or conjugate transpose.

$$\begin{aligned}
 R &= E[\mathbf{x}\mathbf{x}^H] \\
 &= E[\mathbf{S}\alpha\alpha^H\mathbf{S}^H] + E[\mathbf{n}\mathbf{n}^H] \\
 &= \mathbf{S}\mathbf{A}\mathbf{S}^H + \sigma^2\mathbf{I} \\
 &= \mathbf{R}_s + \sigma^2\mathbf{I}
 \end{aligned} \tag{4.13}$$

where

$$\mathbf{R}_s = \mathbf{S}\mathbf{A}\mathbf{S}^H \tag{4.14}$$

$$\mathbf{A} = \begin{bmatrix} E[|\alpha_1|^2] & 0 & \cdots & 0 \\ 0 & E[|\alpha_2|^2] & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E[|\alpha_M|^2] \end{bmatrix} \tag{4.15}$$

$\mathbf{R}_s$  is the signal co-variance matrix,  $N \times N$  that has rank  $M$ . When performing eigenvalue decomposition of the correlation matrix  $\mathbf{R}$  it results in  $N$  eigenvalues and eigenvectors, there are  $N - M$  eigenvalues and corresponding vectors that span the noise subspace  $\mathbf{Q}_n$ , and  $M$  spanning the signal subspace  $\mathbf{Q}_s$ . The eigenvalues and corresponding vectors are then sorted in descending order of eigenvalue. The  $N - M$  eigenvalues that are equal to  $\sigma^2$  are the eigenvalues that span the noise subspace and the corresponding vectors create the matrix  $\mathbf{Q}_n$ . If we let  $\mathbf{q}_m$  be one of the eigenvectors corresponding to the noise subspace we can show,

$$\begin{aligned}
 \mathbf{R}_s\mathbf{q}_m &= \mathbf{S}\mathbf{A}\mathbf{S}^H\mathbf{q}_m = 0 \\
 \therefore \mathbf{q}_m^H\mathbf{S}\mathbf{A}\mathbf{S}^H\mathbf{q}_m &= 0
 \end{aligned} \tag{4.16}$$

This implies that the noise eigenvectors are orthogonal to the  $M$  steering vectors and is the basis of the MUSIC algorithm. Where orthogonal vectors are those whose dot product equals 0, if plotted in 2D these would be perpendicular (Weisstein, n.d.).

Once the noise subspace matrix  $\mathbf{Q}_n$  has been determined the MUSIC pseudo spectrum of power can be calculated as a function of  $\theta$  and  $\phi$  as shown in Eqn. 4.17.

$$P_{MUSIC}(\theta, \phi) = \frac{1}{\mathbf{s}^H(\theta, \phi) \mathbf{Q}_n \mathbf{Q}_n^H \mathbf{s}(\theta, \phi)} = \frac{1}{\|\mathbf{Q}_n^H \mathbf{s}(\theta, \phi)\|^2} \quad (4.17)$$

#### 4.2.1. LabVIEW Implementation

When implementing the MUSIC algorithm in LabVIEW I started with the narrowband MUSIC approach where it is assumed the signal only contains a single frequency  $f$  the same as is outlined in Section 4.2.

Firstly the complex phase delays for each microphone in the array was generated for every combination of elevation and azimuth. These were calculated using equations 4.18 and 4.19.

$$\tau_n(\theta, \phi) = \frac{x \cdot \cos(\theta) + y \cdot \sin(-\phi)}{c} \quad (4.18)$$

$$\mathbf{s}_n(\theta, \phi) = \exp(-j \cdot 2\pi \cdot \tau \cdot f) \quad (4.19)$$

For  $n = 1, 2, \dots, N$

Next the correlation matrix is calculated for  $\mathbf{x}$ , the true value is not known and must be estimated, the estimation is made using the received data from the microphone array, this is performed by averaging the result over several snapshots of data as can be seen in Eqn. 4.20 the rule of thumb is that  $K > 2N$  for the number of averages required (Adve, 2013).

$$\mathbf{R} = \frac{1}{K} \sum_{k=0}^K \mathbf{x}_k \mathbf{x}_k^H \quad (4.20)$$

To get the value for  $\mathbf{x}_k$  the length of the incoming signals from each array in samples is determined, the signal for each channel is then divided into  $K$  equal snapshots of equal length. A Fourier transform is performed for each snapshot for each channel and the value of the complex spectrum for the particular narrowband frequency  $f$  is returned for each channel and snapshot. As the signal is being broken up into  $K$  snapshots the returned value for the complex spectrum may not be only for the narrowband of  $f$  and rather will be the value for the closest frequency bin to the frequency  $f$  as was outline in Section 2.2.1.

Now with  $K$  vectors of  $\mathbf{x}$ , the correlation matrix can be calculated for each snapshot and

then averaged, this gives an estimation for the real correlation matrix.

Once the correlation matrix has been estimated eigenvalue decomposition is performed. This returns  $N$  eigenvalues, and eigenvectors, these are sorted in descending order and split into the signal and noise subspace. The  $N - M$  eigenvectors corresponding to the  $N - M$  smallest eigenvalues these eigenvectors create the matrix  $\mathbf{Q}_n$ , in the software implementation of MUSIC the  $N - M$  eigenvalues should be the smallest and all roughly equal. Now applying Eqn. 4.17 the direction of arrival for the  $M$  signals can be determined by the peaks in the equation.

Below Figure 8 and 9 show the LabVIEW block diagrams of one of my implementation of the MUSIC algorithm, in particular wideband MUSIC through the given range of frequencies. Figure 8 shows the process of iterating through each channel and then for each snapshot the complex spectrum value is calculated as well as the microphone delay matrix.

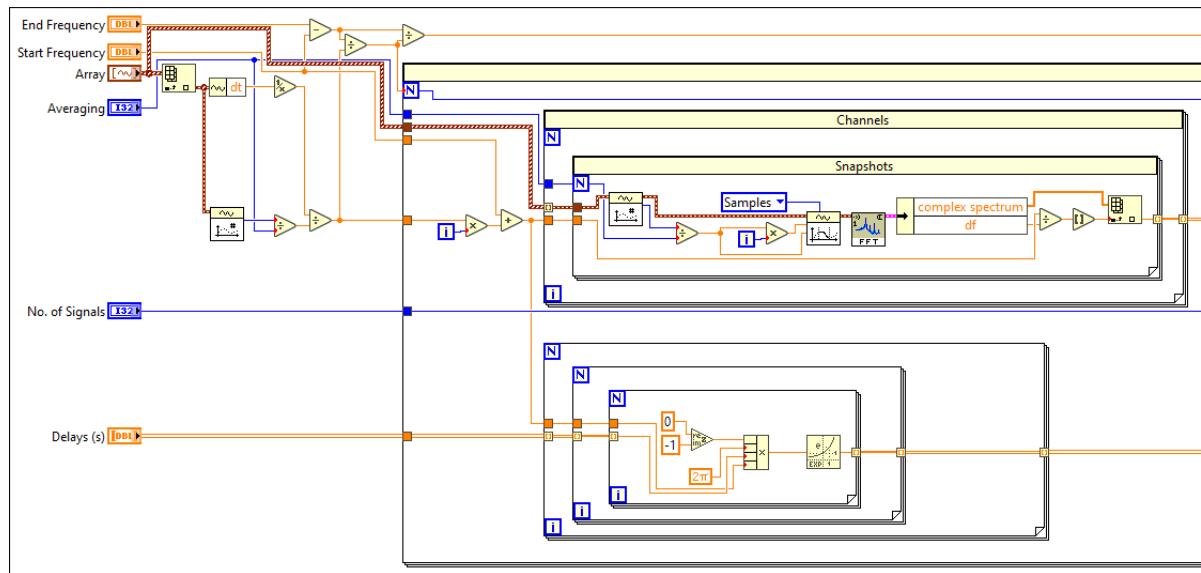


Figure 8: LabVIEW MUSIC Implementation Block Diagram Pt.1

Figure 9 shows the calculating of the correlation matrix for each snapshot, the averaging of this, the eigen-decomposition of the estimated correlation matrix over the  $K$  snapshots and the calculation of the MUSIC power spectrum following the process that has been described above.

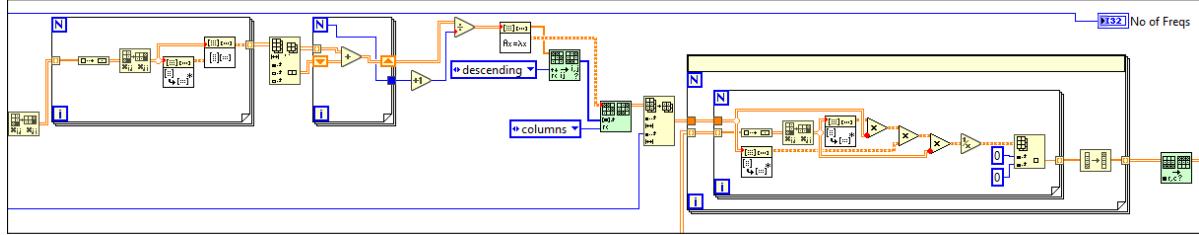


Figure 9: LabVIEW MUSIC Implementation Block Diagram Pt.2

When applying the MUSIC algorithm to a wideband signal instead of a single frequency there will most likely be multiple frequencies all belonging to the same spectrum bin, if the algorithm was to iterate through every single frequency and solve this would increase the computational effort required dramatically as it could be required to iterate through hundreds of unnecessary frequencies. To reduce the computational requirements the size of the frequency bins of the Fourier transform for each snapshot are calculated using the following equation where  $N_s$  is the number of samples in the full signal. [check](#)

$$f_{bin} = \frac{f_s \cdot K}{N_s} \quad (4.21)$$

Once the size of the frequency bins are established the wideband frequency range is then divided by the bin size to determine how many iterations/FFTs are required as then rather than iterating through each individual required frequency, iterations are done by frequency bin which reduces computational requirements potentially by hundreds of times.

For each frequency bin the full narrowband compute as described above is calculated and the resulting power for each frequency at each point is averaged to calculate the estimated direction of arrival of the  $M$  wideband signals. Another implementation of the MUSIC algorithm is using a select number of pre-determined frequencies to perform the average over for rather than a range, this is particularly useful when locating machine defects as covered in greater detail in Section 6, as well as being a more accurate way to locate defects it is much more computationally efficient than calculating the power array for all of the frequency bins in a range.

Identical to the 2D array of power returned by the delay-and-sum the MUSIC algorithm returns a value for each spatial location using equation 4.17 which is normalised between -1 and 1, again with 1 being the most likely location of the signal of interest to allow direct comparison to the delay-and-sum algorithm and easy interpretation of results. The

one downside of MUSIC compared to the delay-and-sum algorithm the MUSIC algorithm does not actually resolve a resultant signal at each location and therefore cannot be used to extract signal without the help of delay-and-sum.

add a section on fft windows and the effects it can have on the results of the MUSIC algorithm

### 4.3. Simulation Package

To ensure all implementations of beamforming algorithms perform as intended and for the purpose of testing all changes to algorithms and ideas relatively quickly I built a simulation package that would be able to simulate any number of different spatial signals and additional noises for any chosen microphone array geometry rather than having to setup a real test environment with an array and data acquisition.

This simulation package was designed with the purpose of being as flexible as possible and adding as many features as required while designing the array and testing algorithms throughout the software design phase of the project. The following sections outline the primary functions of the simulation package including the Array Generator, Delay Calculator, Spatial Signal Generator, and File Playback.

### 4.3.1. Array Generator

The first required function of the simulation package was to be able to generate microphone array geometry or import them from a file. The array generator has options to generate a uniform linear array, grid array, circular array or an under-brink array for the purpose of quickly generating any number of different array geometries. Figure 10 shown below is of the LabVIEW block diagram for the array generator.

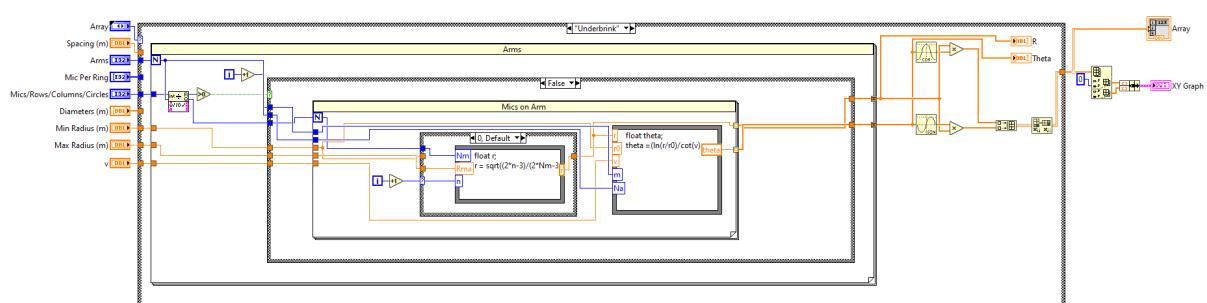


Figure 10: Array Generator LabVIEW Block Diagram

The LabVIEW block diagram only shows a partial snippet of the program and is the function for generating a custom under-brink array. Starting from the left, it takes the inputs of Arms, Minimum Radius, Maximum Radius and Arm Angle. These are processed through a math-script function block to calculate both the radius and angle of each particular microphone as according to the equations presented in Section 5.3, once processed the values are converted from cylindrical co-ordinates to Cartesian co-ordinates. On the right hand side of the diagram is the outputs, the XY graph that is shown in Figure 11 and the array of Cartesian co-ordinate.

Below in Figure 11 the user interface for the Array Generator is shown. As well as giving various options for array customisation a live plot of the current array is displayed on the right hand side of the user interface.

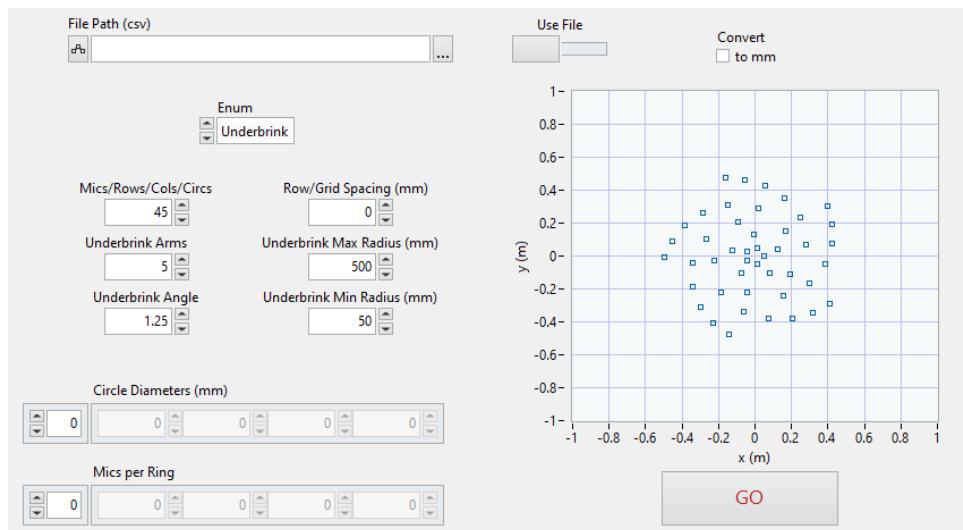


Figure 11: Array Generator LabVIEW Front Panel User Interface

#### 4.3.2. Delay Calculations

**probably re write** At the core of all beamforming algorithms is the relative delays between each microphone for an impinging signal. This function shown in Figure 12 applies Eqn. 4.5 and 4.6, by taking in the microphone array in Cartesian co-ordinates and generating an array of delays relative to the first microphone for a signal arriving at a particular Azimuth and Elevation.

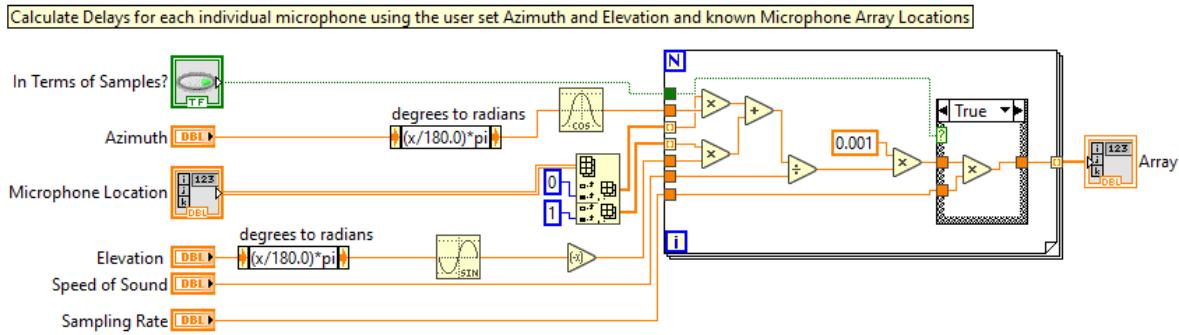


Figure 12: Microphone Relative Delay Generator LabVIEW Block Diagram

By iterating through this function for every single combination of Azimuth and Elevation a 3 dimensional array of delays is produced for, this includes the time or sample delay of every microphone for a signal coming from all possible combinations of Azimuth and Elevation in front of the microphone array. This function is at the core of all beamforming algorithms and simulations for recreating spatial signals.

#### 4.3.3. Spatial Signal Generator

Using the generated microphone array geometry and the generated array of microphone delays from Section 4.3.1 and 4.3.2 respectively equivalent spatial signals can be recreated for the purpose of simulating and testing microphone array performance. Figure 13 shows the block diagram of the LabVIEW VI that performs the signal generation. This VI can recreate up to two separate spatial signals, with added white noise and has a choice of equivalent sampling rate and the number of samples capture. This allows for a more accurate representation of array performance and especially for the ratio of processing time to signal length and the effects of the signal length on algorithm and array performance.

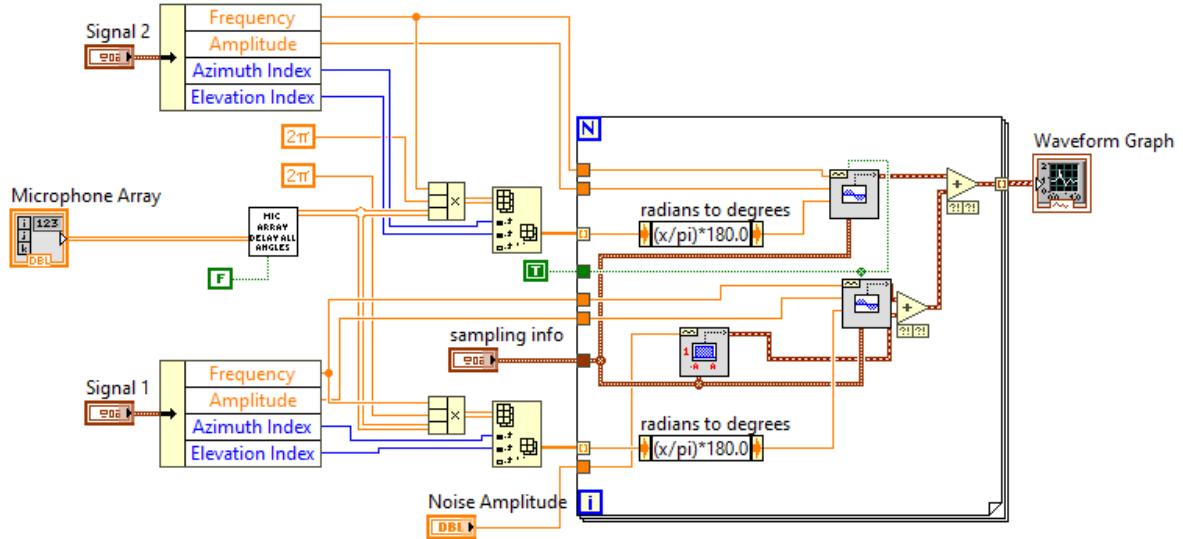


Figure 13: Spatial Signal Generator LabVIEW Block Diagram

The spatial signals are simulated by calculating what the phase delay would theoretically be for a signal coming from a spatial location. For instance, to simulate a signal of frequency  $f$  that is coming from a azimuth of  $\theta$  and elevation of  $\phi$  the phase can be calculated using Equation 4.22 by converting time in seconds to phase in degrees. The phase delay is calculated for each microphone to generate a separate signals for each to simulate the real world situation.

$$\tau_\varphi = \frac{\tau(\theta, \phi) \cdot f \cdot \pi}{180} \quad (4.22)$$

Figure 14 shows the user interface for the spatial signal generator, besides giving a way to choose the values it has helped with algorithm development as it gives a way to visualise what perfectly delayed signals should look and therefore how the delay-and-sum algorithm works.

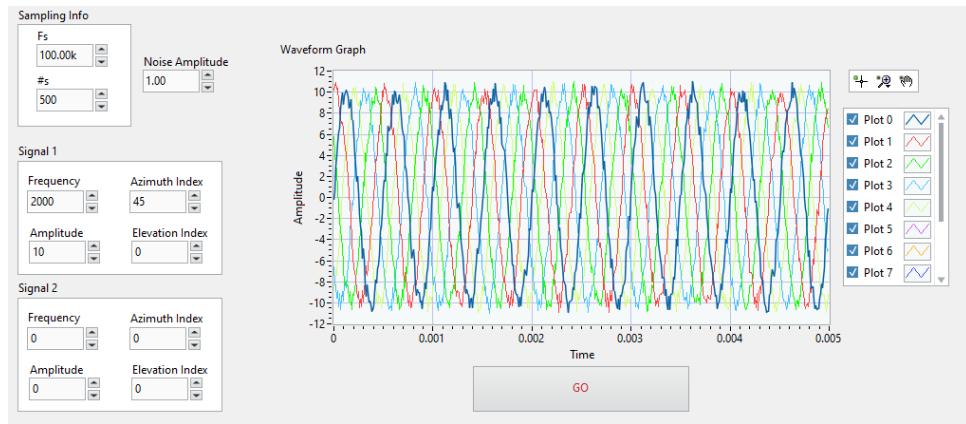


Figure 14: Spatial Signal Generator LabVIEW Front Panel User Interface

#### 4.3.4. File Playback

The file playback function was added as a nice easy way to replay back data from testing and also gives the ability to choose any section of the file to perform beamforming with and perform post-processing.

Figure 15 shows the user interface for playing back files, displaying the sound waves on the right and by zooming in on this plot different sections of the waveform can be used for processing.

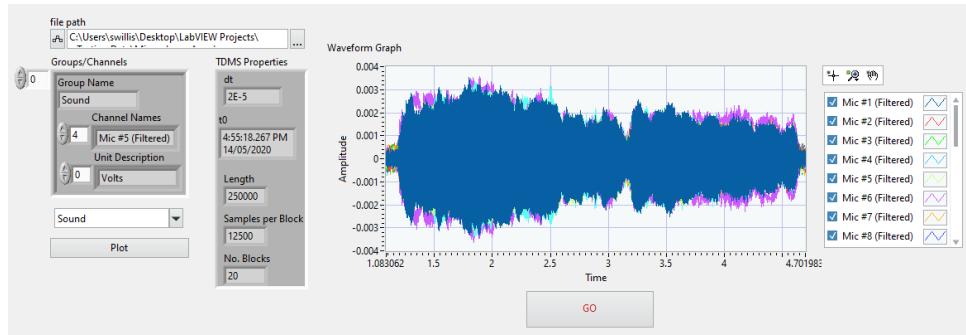


Figure 15: File Playback LabVIEW Front Panel User Interface

#### 4.3.5. Simulation Testing and Validation

With the known phase delay an array of sinusoidal waves are able to be generated with the correct phase that would theoretically be received at each sensor of the array. When this array of signals is input into any beamforming algorithm it ideally gives a performance gauge of the algorithm itself and the geometry of the array. When the

output heat-map is compared against the 3D frequency response heat-map it is clear whether the results are as expected or not, Figure 16 demonstrates this. The white is the estimated location of the signal source going down to blue then black as the least likely.

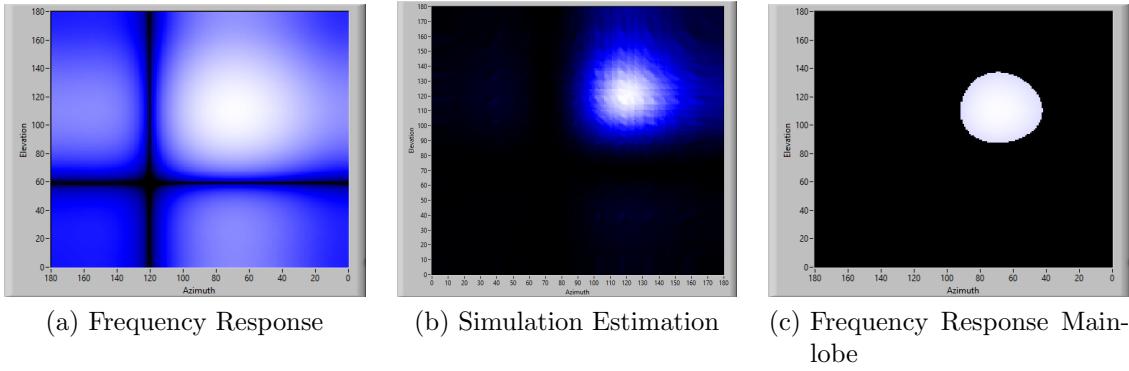


Figure 16: Comparison of Simulation Results to Expected Results

Comparing the mathematical model used in the frequency response and the implemented delay-and-sum algorithm in the simulation both returned nearly the same results, this indicates that the delay-and-sum algorithm is working as intended and that the simulation works as intended as well. The two main side-lobes in Figure 16a can also be seen in Figure 16b towards the top left and bottom right for further verification. Additional Figures from this simulation are included in Appendix A

#### 4.3.6. Algorithm Simulation Comparison

This section details the core differences found in simulating both the Delay-and-Sum Algorithm and the MUSIC Algorithm. Figure 17 shows the comparison of the two algorithms response to a single 1863Hz frequency signal at 13 Degrees to the left of centre Azimuth and 16 Degrees below the centre Elevation.

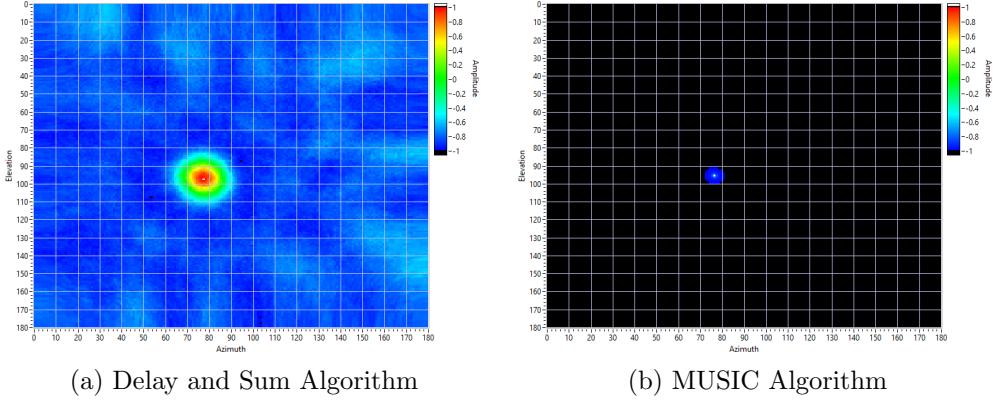


Figure 17: DAS and MUSIC Algorithms compared at 1863Hz

The differences between algorithms is very clear. To make the comparison as fair as possible the input signals were of different lengths to make the compute time equal which starts to show the computational differences between the two. The input signals for MUSIC was 8 seconds long while the input signals for delay-and-sum were 0.1 seconds. MUSIC algorithm outperforms the delay-and-sum drastically, as it estimates the signal location to pin-point accuracy, the delay-and-sum's most likely estimation is only a few degrees off the real location, though looking at the estimation results it could realistically be anywhere within the 7 degree radius hot-spot.

Figure 18 shows a test of the ability of both algorithms to separate two signals of the exact same frequency. Two single frequency 1863Hz signals were produced both on the horizontal and 12 Degrees either side of the centre giving 24 Degrees of separation, 24 degrees was chosen as it was the smallest degree of separation that both algorithm could separate the signals. The same length of input signals was used as the previous example to keep it comparable. This test shows slightly different results but still the MUSIC algorithm performs better as it has a definite differentiation between the signals but the estimation of location of the left signal is not correct, whilst the delay-and-sum has only just differentiated the signals but is giving a broader area of location estimations.

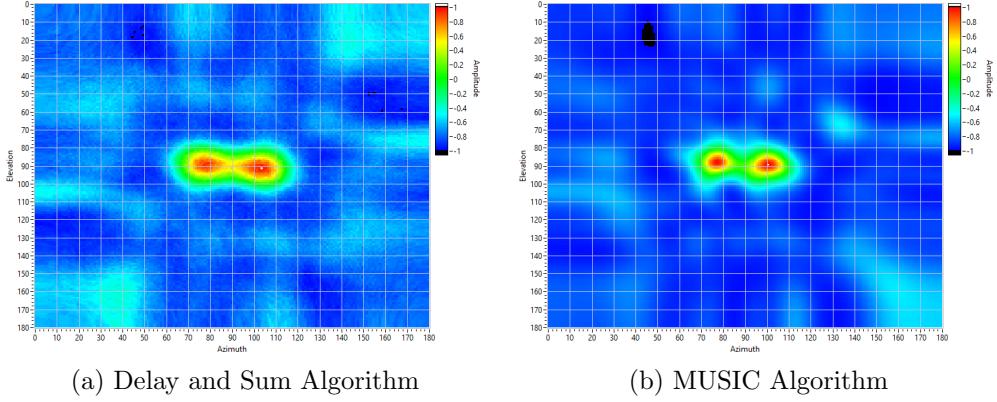


Figure 18: DAS and MUSIC Algorithms compared at 1863Hz Source Separation

Overall through extensive testing it was found that the delay-and-sum algorithm was a much more robust algorithm but is essentially attempting to find the strongest sound source in its view which can lead to misleading results, as well as this it is much more computationally intensive compared to the MUSIC algorithm. The MUSIC algorithm performed significantly better and quicker than the delay-and-sum algorithm in simulation showing great potential but does require more knowledge of the signal of interest for it to be effective as well as fast.

## 5. Hardware Research and Development

### 5.1. 2D Frequency Response

A Frequency response is a visual way of analysing the effectiveness of microphone array geometries to a certain frequency coming from particular directions and allows for a quick analysis of array geometries. A frequency response for a uniform linear array of 10 microphones with a spacing of 8cm subject to a 2kHz signal is pictured in Figure 19 and the same response when digitally steered to  $130^\circ$  in Figure 20.

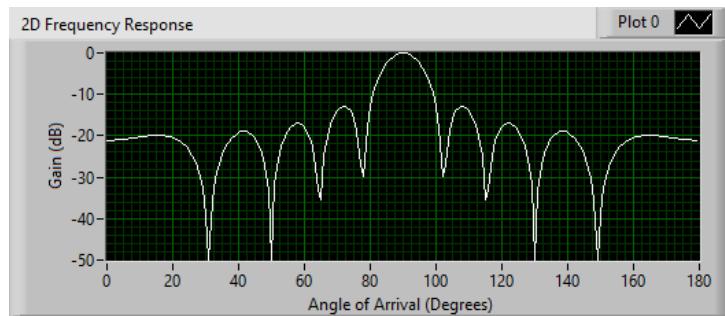


Figure 19: 2D Frequency Response

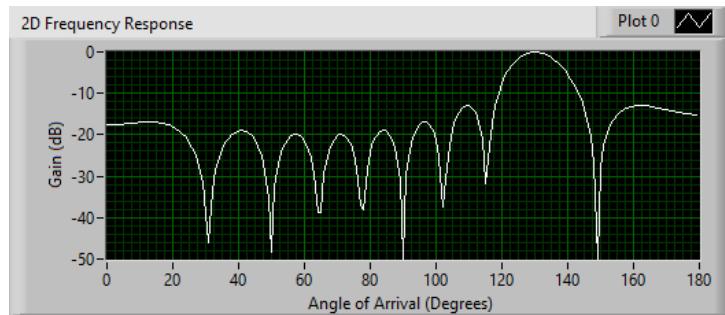


Figure 20: 2D Frequency Response Steered to  $130^\circ$

When evaluating a 2D Frequency response the main points of interest are the main-lobe, its beam-width and the side lobes. The main-lobe is the lobe with the greatest amplitude seen at  $90^\circ$  in Figure 19 the beam-width is the width from where the main-lobe drops to -20dB as this is half the power of 0dB, this is effectively the resolution of the array at that frequency. Side lobes are important to show the effective gain of interference that appear to be coming from angles other than the angle of interest due to spatial aliasing, these can be seen in Figure 19 and 20 by the 7 or 8 lobes with peaks around -20dB gain. The goal of microphone array geometry design is to minimise the side lobes minimizing interference, and decreasing the beam width to give a better array resolution.

Equation 5.1 was adapted from Benesty et al. (2008) and has been applied to calculate 2D Frequency responses in Figures 19 and 20.

$$Response(\theta) = 20\log_{10}\left(\left|\frac{\sum_{n=1}^N \exp\left(\frac{-2\pi j x_n (\cos(\theta) - \cos(\varphi))}{c}\right)}{N}\right|\right) \quad (5.1)$$

$\theta$  defined as the angle of arrival, where  $x$  is the location in the  $x$ -direction from the origin, and  $\varphi$  is the steered angle of the beamformer.

Another useful visualisation using Equation 5.1 is a form of Frequency response that repeats the 2D Frequency response above over a range of frequencies, this makes it easier to spot if there are any problem frequencies due to the array geometry. Using the same array as Figure 19, Figure 21 shows the frequency response from 0Hz to 6kHz.

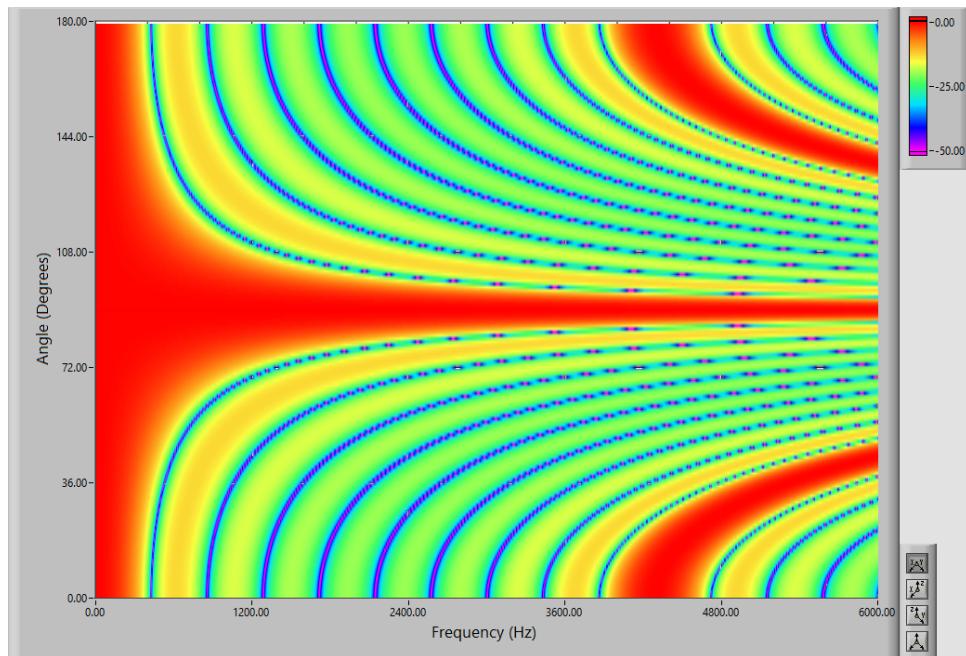


Figure 21: 2D Frequency Response over Varying Frequencies

Figure 21 demonstrates precisely how useful it can be for spotting problem frequencies for particular array geometries. We can see at around 4kHz another main-lobe begins, this will make it impossible to determine where the signal is truly coming from as it would seem as it is coming from 3 directions this is due to the microphone array sensor spacing being a multiple of the signal wavelength causing spatial aliasing to occur, creating misleading results. As can be calculated with Equation 2.2 just below 4kHz the

wavelength is 8cm, exactly the same as the microphone spacing, providing reason for the spatial aliasing.

Another important property of this array is where it becomes use-able, before 400Hz the array has no directionality and cannot determine the direction of a source of interest. The lack of directionality is due to the aperture of the array as mentioned in Section 2.2, below 400Hz the signal wavelength is larger than the array aperture hence it is useless for directionality below this point.

2D Frequency Response analysis of microphone arrays is useful if the geometry of the array is symmetrical. The 2D Response is only looking at the angle of arrival across one of the axes, in this case the angle of arrival is moving across the array azimuth and therefore we cannot see the response of a signal moving through the elevation. For arrays that are symmetrical this doesn't matter as the response will be the same for both azimuth and elevation. A more useful analysis for non-symmetrical array geometry is 3D Frequency Response.

## 5.2. 3D Frequency Response

3D Frequency response is based on Equation 5.2 using a modified version of Equation 5.1, adding a way to analyse the  $y$  location of each microphone and its respective response due to that.

$$Response(\theta) = 20\log_{10}\left(\left|\frac{\sum_{n=1}^N \exp\left(\frac{-2\pi j(x_n(\cos(\theta)-\cos(\varphi_A))+y_n(\sin(\phi)-\sin(\varphi_E))}{c}\right)}{N}\right|\right) \quad (5.2)$$

The main difference between the equation for 2D and 3D Frequency Response is the added factor for the steering and response along the elevation so a response can be found for all spatial locations in front of the array. Figure 22(a), 22(b), 23 and are the results for a 3D Frequency response for a grid array with 16 Microphones, 4 rows, 4 columns with a row and column spacing of 8cm subject to a 2kHz signal and digitally steered towards the origin of the array.

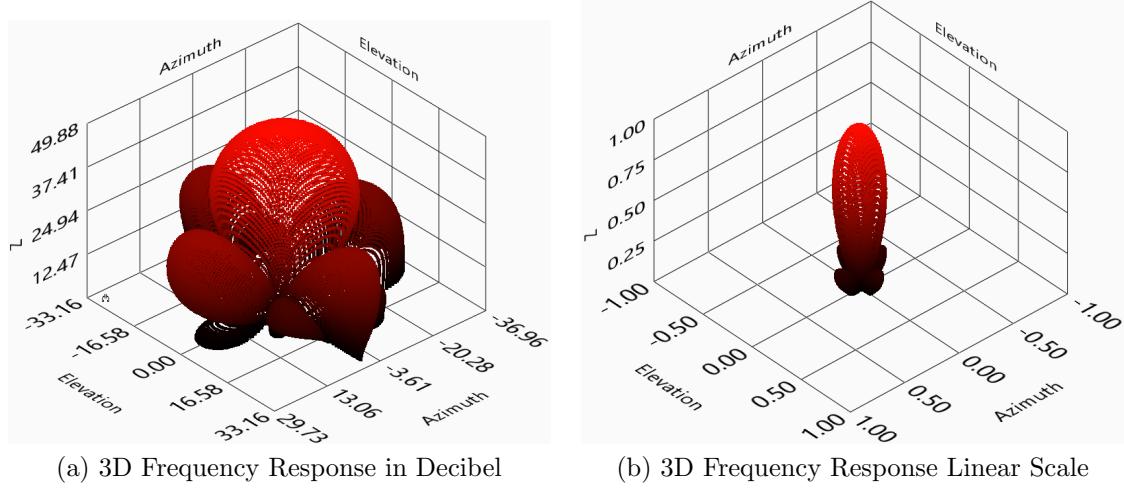


Figure 22: 3D Frequency Responses

Figure 22(a) and 22(b) are both 3D Frequency Responses from the same array with different scaling, the decibel scaling can be useful for a lot of analysis as it attenuates the side-lobes a lot more than a linear scale. These figures allow easy visualisation of the side-lobes and nulls that appear at any particular frequency, these are just a 3D representation of the heat-map in Figure 23, the heat-map is the most useful for visualising array performance due to the ease of interpretation.

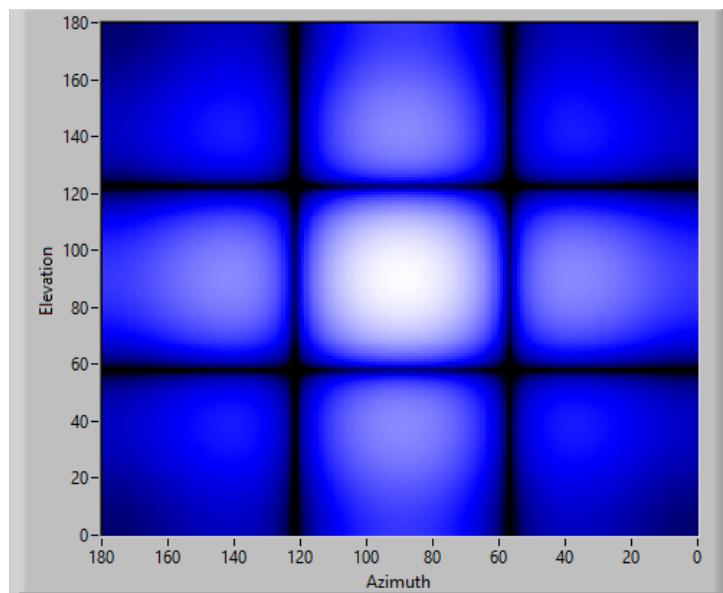


Figure 23: 3D Frequency Response Heatmap in Decibel

When analysing an array geometry using a heat-map like the one pictured in Figure 23

the high points are in white going through blue to black at the lowest. This grid array in particular has 1 main-lobe pictured in the centre as the white area and 4 particularly high side-lobes on all sides of the main-lobe with nulls in between. This array shows a good response at this frequency but at different frequencies it is clear it will experience spatial aliasing with such prominent side-lobes. These side-lobes can also be clearly seen in Figure 5.2(a). The one drawback of the 3D Frequency response is only being able to analyse a single frequency at a time.

### 5.3. Geometry Design

The main constraints with the design of microphone array is the number of handle-able synchronised high sample-rate analog inputs.

A comparison of various array geometries was performed by Zebb Prime and Con Doolan of The University of Adelaide, this included various spiral array geometries that were evaluated by comparing their minimum side lobe levels and beam widths. A comparison of all the arrays for both near-field and far-field was presented but as the purpose of the array in this report is for far-field signals the near field results were not taken into account. The comparison showed the Underbrink geometry design was overall the best performing closely followed by the Brüel & Kjær (B&K) array (Prime and Doolan, 2013). Hence for the choice of array for the proof of concept design the Underbrink array was chosen as it shows promising resolution and minimal side lobes especially for the number of sensors being used.

The Underbrink array is shown in Figure 1, it is a "multi-spiral design where the microphones are placed in the centre of equal area segments" (Prime and Doolan, 2013). Where a minimum and maximum radius ( $r_0$  and  $r_{max}$  respectively) must be defined as well as the number of spiral arms ( $N_a$ ) and microphones per spiral ( $N_m$ ). This array is based upon the following set of equations.

The radius of each individual microphone:

$$r_{m,1} = r_0, \quad m = 1, \dots, N_a \quad (5.3)$$

$$r_{m,n} = \sqrt{\frac{2n-3}{2N_r-3}} r_{max}, \quad m = 1, \dots, N_a \quad n = 2, \dots, N_m$$

The angle of each individual microphone:

$$\theta_{m,n} = \frac{\ln(\frac{r_{m,n}}{r_0})}{\cot(v)} + \frac{m-1}{N_a} 2\pi, \quad m = 1, \dots, N_a, \quad n = 1, \dots, N_m \quad (5.4)$$

$$\text{where, } v = \frac{5\pi}{16}$$

To be able to generate many different configurations of the Underbrink array a Python script was initially written, this script also converted the coordinates from polar to Cartesian making it easier to work with the LabVIEW Simulations. As progression was made with LabVIEW this was converted to a LabVIEW module for better integration to the full simulation package.

### 5.3.1. First Design

For the initial testing of microphone arrays and verification that it would be possible the available hardware I designed an underbrink array with an inner radius of 25mm and outer radius of 250mm, 3 arms with 4 microphones each. The reason for 12 microphones was due to a hardware limit as there were only 12 available high speed analog input cards for data acquisition at the time. Through the initial design process it was found that the underbrink array performed better when there was an odd number of arms as it prevent the array from being symmetrical and significantly reduced the side lobe levels and spatial aliasing as is further investigated in Section 5.3.2, a scatter plot of the designed array is pictured in 24.

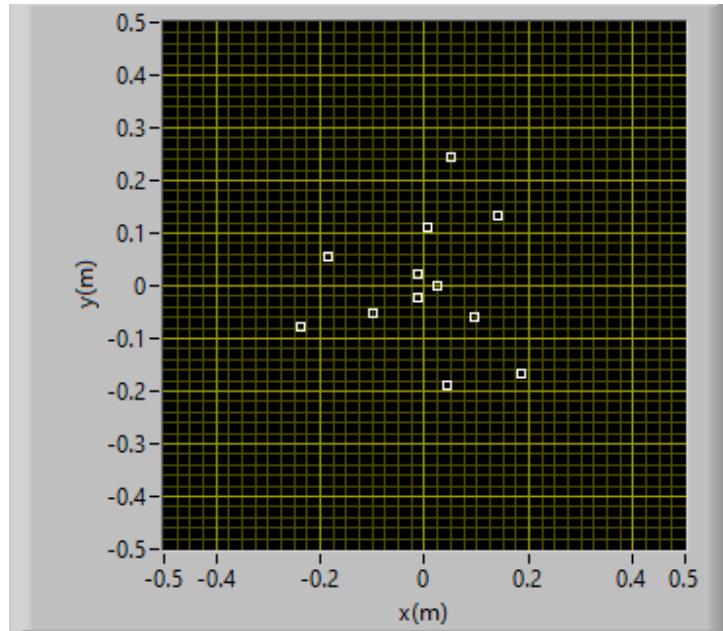


Figure 24: Chosen Underbrink Array

The choice of the inner and outer radius dimensions were based off of the frequency response and ease-of-building, giving very minimal side-lobe levels and acceptable resolution between 1kHz and 4.5kHz without being overly large. Designs were considered with larger radii to have a better response at lower frequencies but for the current purpose of testing algorithms and whether the microphone array will actually work the frequency range is not a problem.

Figure 25(a) and 25(b) show the frequency response for this custom Underbrink array at 2kHz, compared to the grid array shown in Section 5.1 the Underbrink shows very minimal side-lobe levels or obvious side-lobes and no spatial aliasing giving a promising response for the amount of sensors hence the choice for further testing.

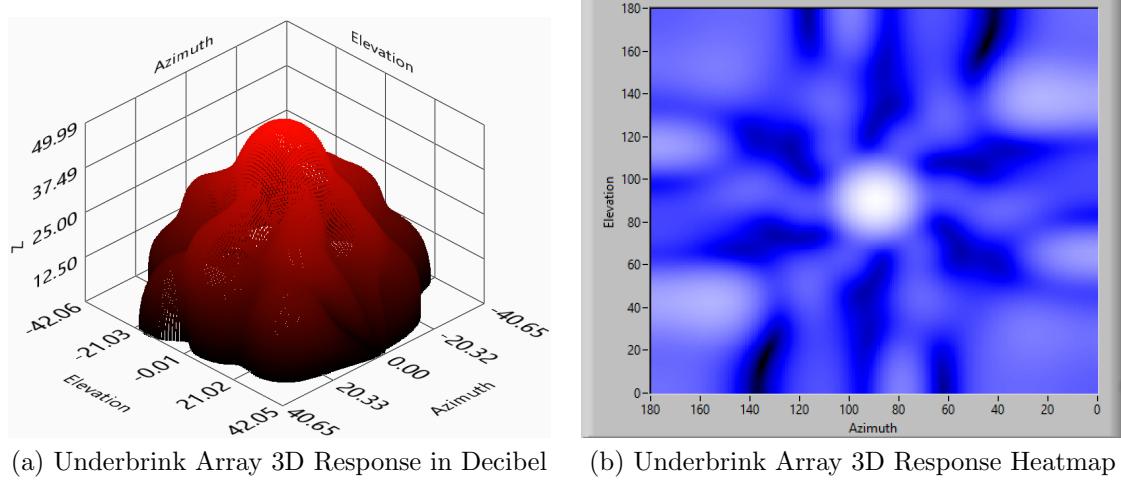


Figure 25: Underbrink Array 3D Responses

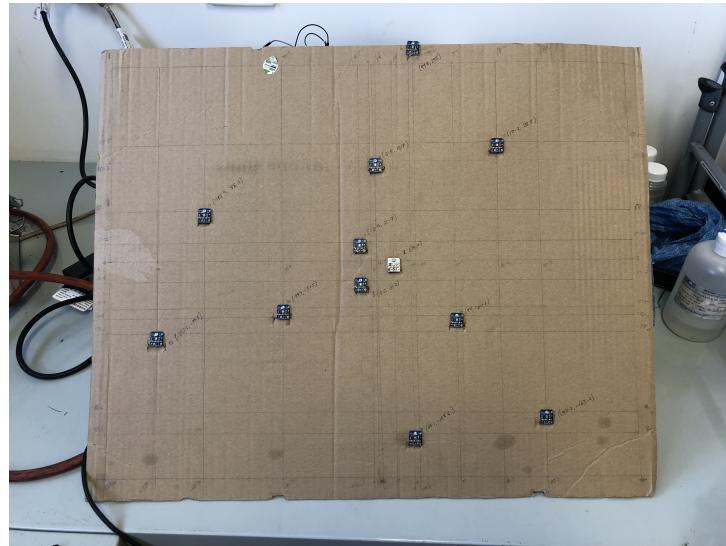


Figure 26: Initial Microphone Array

### 5.3.2. Second Design

For the second iteration of the microphone array I was now limited to 48 synchronous channels. This was to be achieved using two separate National Instrument cRIO-9036's both filled with 8 NI-9232's capable of 3 analog channels sampling at 102.4kHz.

Through simulations of the Underbrink array I again found that an odd number of arms performs better than an even number when using a similar number of microphones. This is found to be due to the symmetrical geometry produced by an even number of arms

that causes the problem of spatial aliasing. A series of microphone array geometries used to demonstrate this is displayed in Figure 27, these geometries range from 42 to 48 microphones being used from 5 to 9 arms and all with a minimum radius of 25mm and maximum radius of 0.5m as this gave the best possible frequency response without the array becoming too large to easily move and transport. The theoretical frequency response of an array of this size will be from 686Hz to beyond the 10kHz limit of the microphones.

Figure 28 displays the 3D Frequency Response of each of the arrays from Figure 27 when subject to a pure 4kHz signal at the absolute centre of the array. This demonstrates that purely having more microphones will not always result in a better performing array and that other factors will see better improvements in performance. All of the geometries have a near identical beam-width, this was expected as the 4kHz is well within all of their frequency response, and with all of the arrays having equal inner and outer radius' no array should have an advantage. Both microphone arrays utilising the full 48 microphones show significantly larger side lobe levels than the arrays with less microphones due to the even number of arms on the array.

Taking into consideration the results of the various array geometries the geometry shown in Figure 27.a with 5 arms was chosen for the new array design, this was due to two reason; firstly the array showed similar side lobe levels to the 7 arm and 9 arm geometries but had fewer side lobes, secondly with a desired inner radius of 25mm it would not be possible to fit more than 5 microphones nicely.

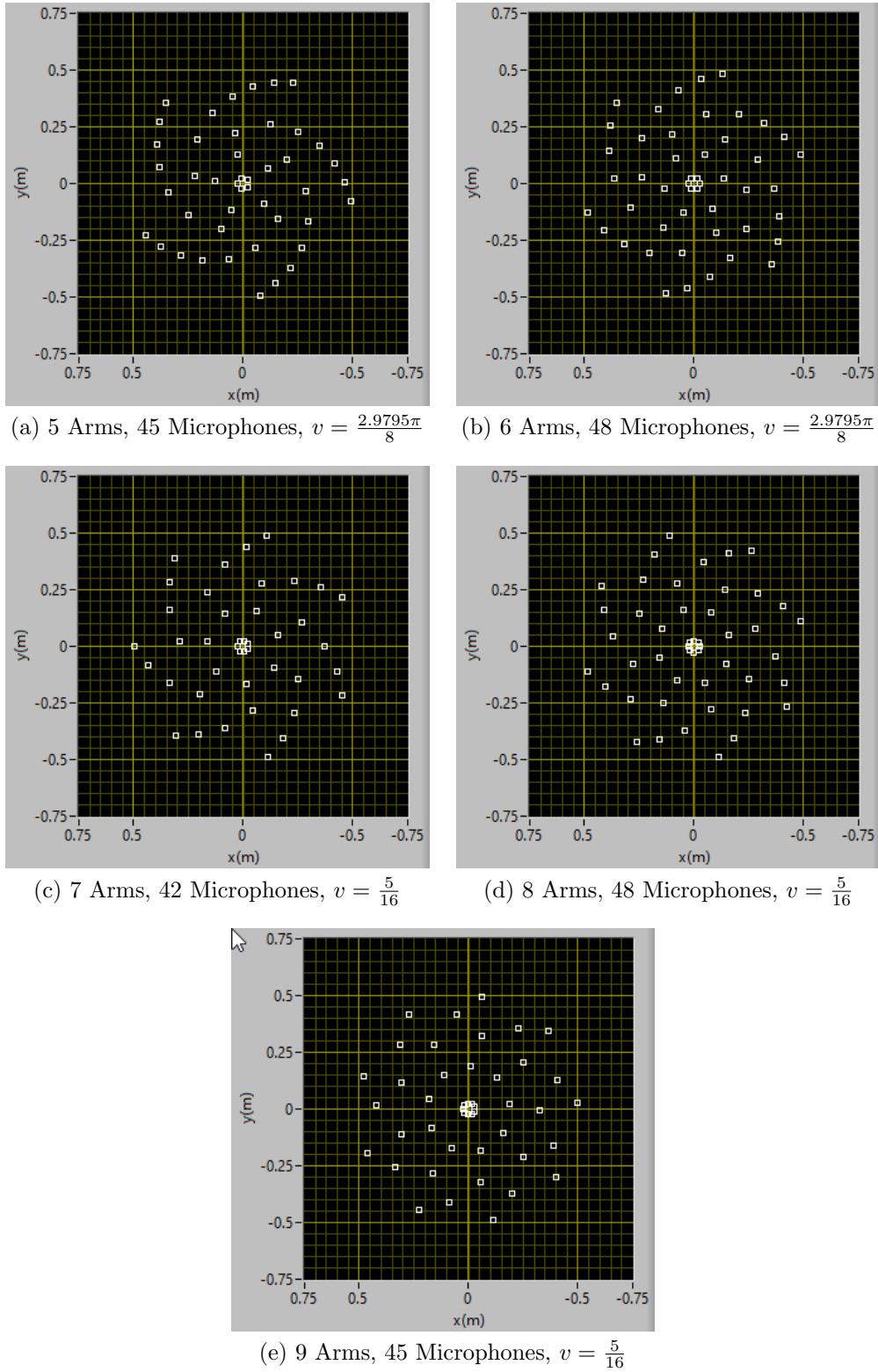


Figure 27: Series of tested microphone array geometries

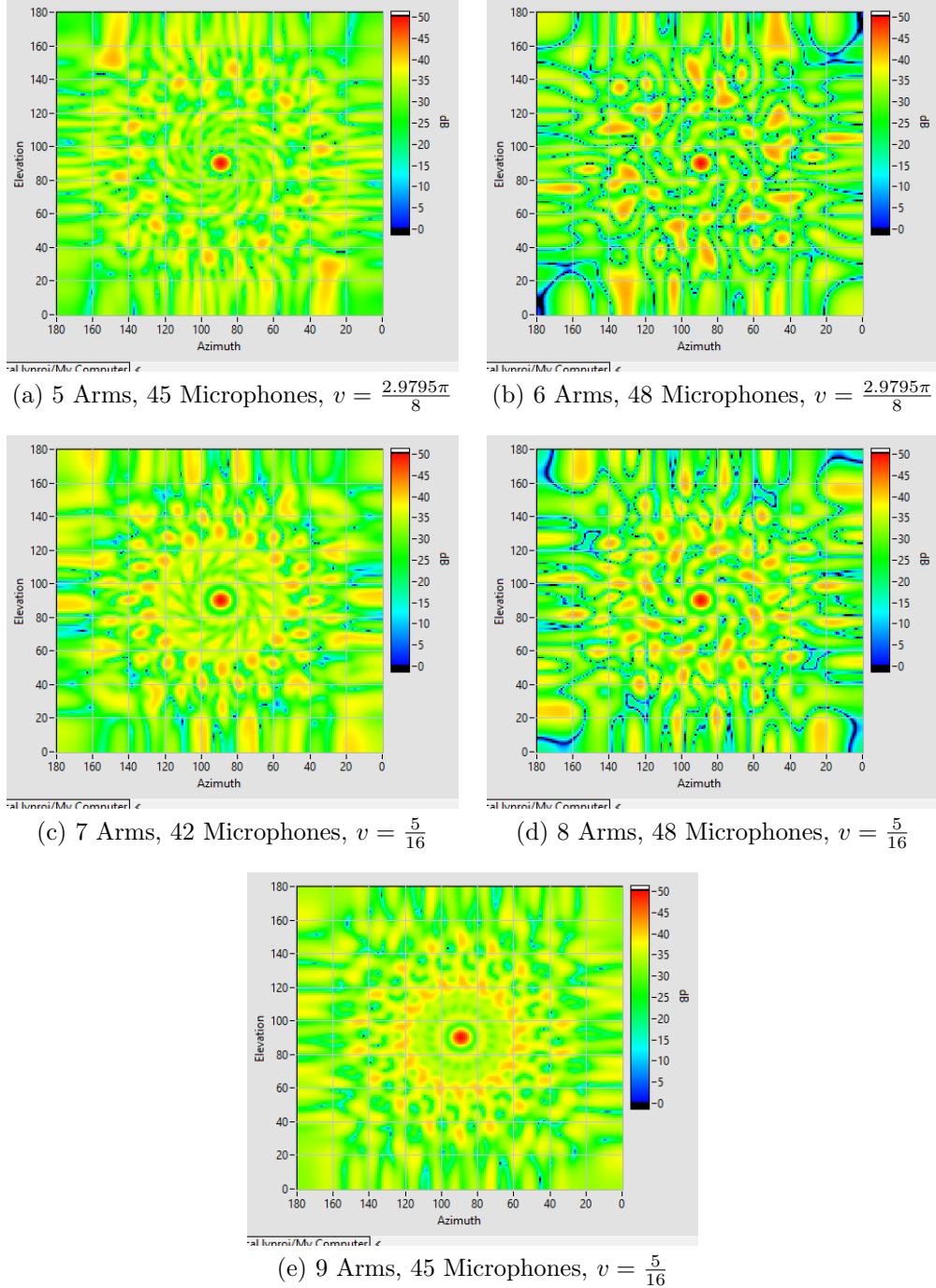


Figure 28: 3D Frequency Response Results

Now with the desired design chosen further analysis was performed to ensure adequate performance from the array. Using the same technique shown in Section 5.1 producing a series of 2D Frequency responses through all the use-able frequencies to see a more

accurate representation of the array response. This was performed for both the 2D response of the Azimuth and Elevation as a 3D frequency response is just a combination of these, performing them separately provides similar results but the ability to see the response over the desired frequencies. Figure 29 shows the frequency response across the azimuth of the array ranging from 0 to 10kHz and Figure 30 showing the same frequency range across the elevation of the array.

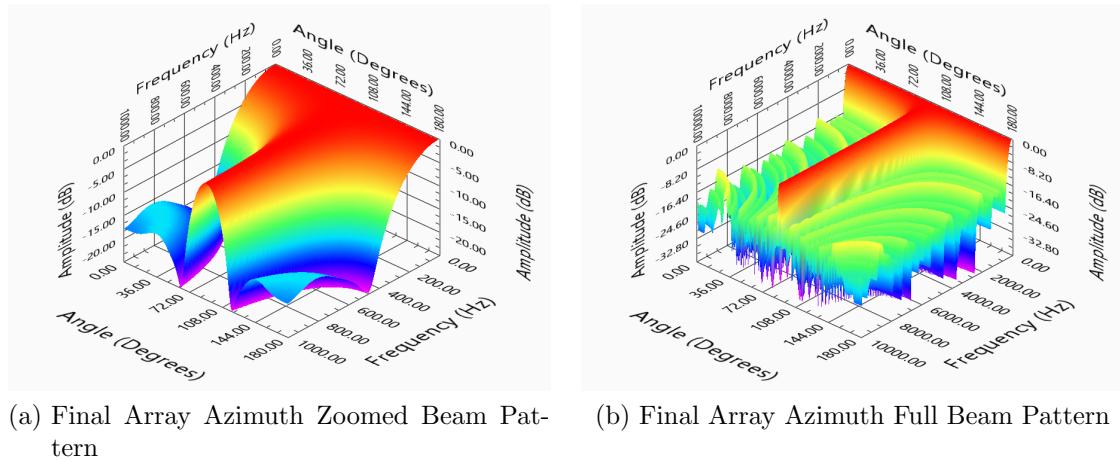


Figure 29: Final Array Azimuth Frequency Sweep Beam Patterns

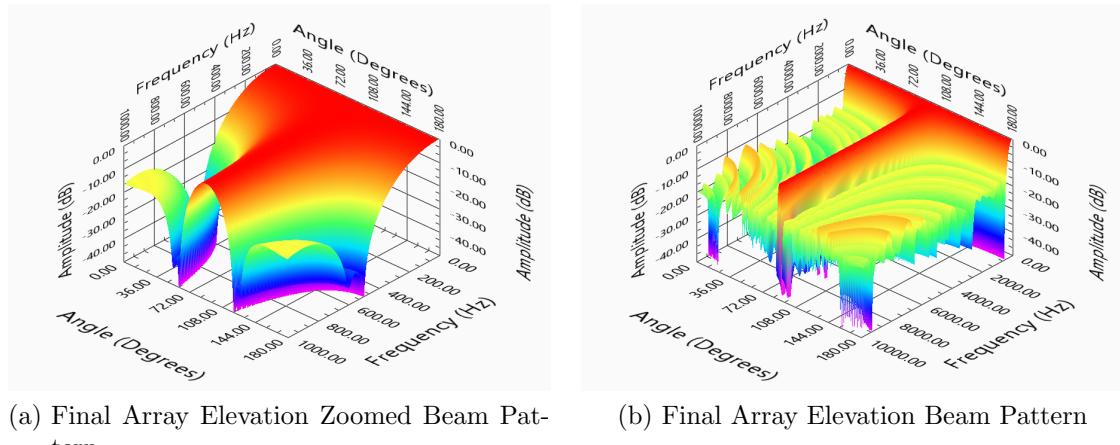


Figure 30: Final Array Elevation Frequency Sweep Beam Patterns

As shown in both of the sweeping 2D frequency responses' the array has very minimal side-lobe levels from 600Hz through to 10kHz. The array begins to have directionality around 750Hz due to the array diameter. Following Eqn. 5.5 with the array diameter of

1m it would equate to 686Hz as can be verified when looking at the frequency response. This relationship is due to spatial aliasing and the requirement for there to be two sensors within the wavelength of the signal as described in Section 2.2.

$$dia. > \frac{2c}{f} \quad (5.5)$$

Now that the design had been finalised and deemed appropriate for the microphones and use case it was modelled in Solidworks as can be seen in Figure 31 (a) due to the complex geometry of the design it was laser-cut out of 6mm plywood in four separate pieces and glued together. The design allows for the microphones to be screwed directly into their correct position with a much greater accuracy than if it was hand-made, this results in much tighter geometry tolerances and therefore higher accuracy beamforming.

A lot of considerations were made when modelling the array for the best possible performance, one of the biggest issues with microphone arrays is noise and conflicting signals causing interference with the expected signals generally known as reverberation, this results in conflicting information being returned by the array. To avoid the problems of reverberation as much of the array surface area between microphones was removed to avoid reverberation from sound bouncing back off of the wooden surface whilst still keeping enough structural rigidity. Around the whole array several small holes were cut out for easier cable management and a flat spot on the top for a level as for accurate results the array must be level.

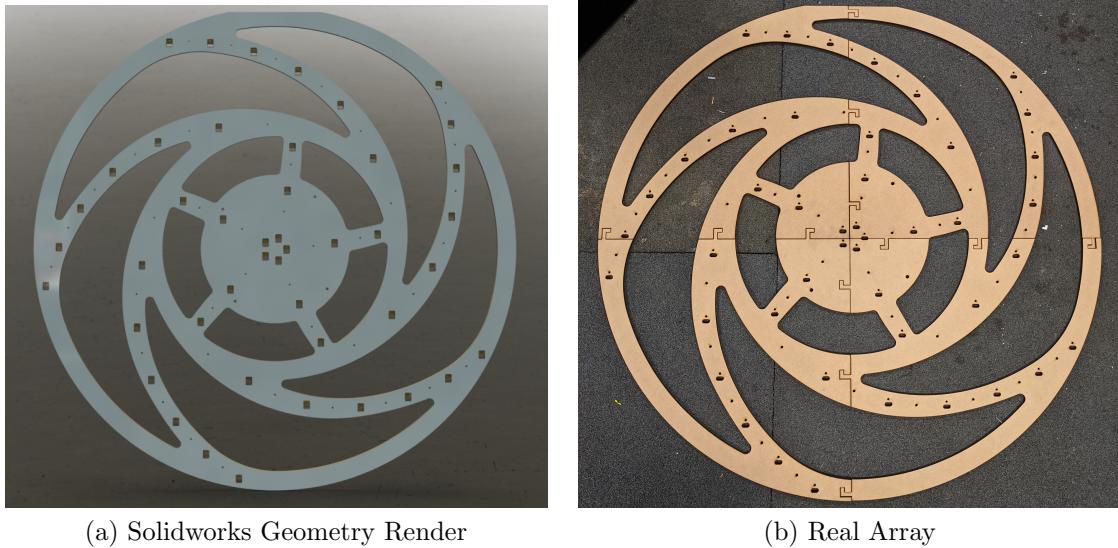


Figure 31: Final Array

### 5.3.3. Design Comparison

Below in Figure 32 a comparison of the first testing array and final design at 3kHz. As expected the final design has much lower side-lobe levels and smaller main-lobe giving the array higher resolution and a higher signal to noise ratio, a significant improvement over the original array.

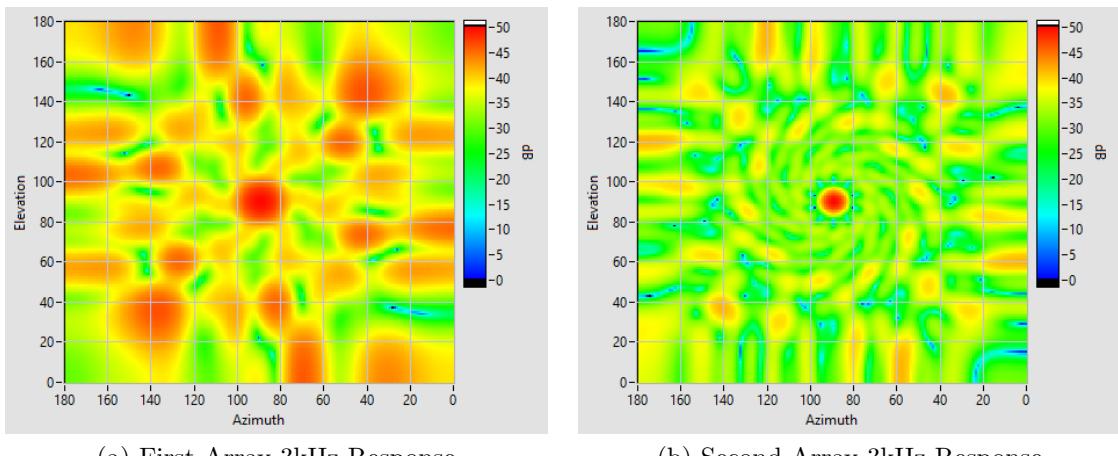


Figure 32: Array Comparison 3D Responses

## 5.4. Microphones

Before the wide availability of MEMS (MicroElectrical-Mechanical System) Microphones arrays were required to be set up with conventional analog microphones, this made the costs and size for arrays very expensive, large and heavy, limiting the widespread availability of this technology with such a high barrier to entry without even considering the data acquisition. With the now every-growing mobile phone industry MEMS Microphones are incredibly cheap starting at less than a dollar each, this allows for large arrays to be built for very low costs (Sanders, 2017). The SPW2430 MEMS microphones were selected for my array, they boast a flat frequency response between 100Hz and 10kHz making them a perfect pairing for the microphone array (Adafruit, n.d.), they are cost efficient and can be bought directly from many local and international distributors on breakout boards that are able to be directly wired into the National Instrument hardware as they have an analog DC output that is easily digitized for analysis and post processing.

## 5.5. Data Acquisition

### 5.5.1. First Array

The initial testing array consisted of 12 of the SPW2430 MEMS Microphones mounted on a large piece of cardboard. There were four NI 9232 C-series modules available for analog input in a NI cDAQ-9178 for data acquisition, at the time of the photos three of the c-series modules were being used for other projects. The NI 9232 C-series modules are designed for sound and vibration inputs and take in three signals each simultaneously, with built in anti-aliasing filters for the selected sample rates up to 102.4kHz per channel (National-Instruments, 2019). The NI cDAQ-9178 allows for the data acquisition from the NI-9232 modules and the synchronisation of the acquired data across multiple modules, it communicates with a computer via a USB for plug-and-play simplicity (National-Instruments, 2019). Figure 33 and demonstration of how the components are put together.

One of the most important outcomes taken from the initial testing of this array was the need for proper wiring, the wiring used can be seen in Figure 33 is a high gauge wire that comes with 3-wires in a flat ribbon with no shielding or way to prevent environmental electrical noise. This resulted in a lot of noise across all the channels and was something that must be changed for the next iteration.

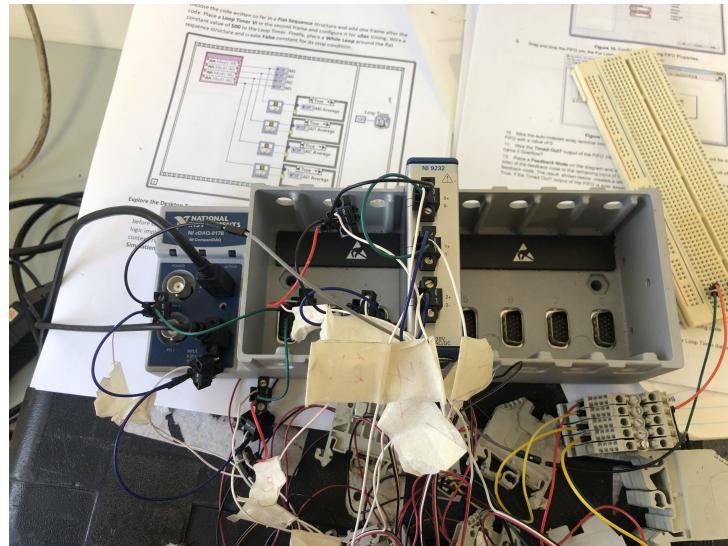


Figure 33: NI cDAQ with a single NI 9232 Installed and Wired Up

### 5.5.2. Second Array

The second array was designed to be used with two NI cRIO-9036's Real Time Targets with FPGA's and 16 NI-9232 C-series Modules. This configuration would have allowed for up to 48 Channels of synchronised 102.4kHz input signals being processed on the cRIO's themselves with potential to move some processing to the FPGA. Due to unforeseen circumstances this hardware was made inaccessible and unavailable and a change of hardware was required. The hardware was changed back to the NI cDAQ-9178 and 8 NI-9229's C Series Modules, this allows for up to 24 synchronous channels of 50kHz sample rate (National-Instruments, 2020). Hence the final microphone array only consisted of 24 microphones with all processing taking place on an attached computer. As the array was designed for the use of 45 microphones spanning over 5 arms it had to be cut down, the inner 20 were microphones were chosen and then 4 of the 5 on the next ring of microphones as shown in Figure 34.



Figure 34: Final Microphone Array with 24 Microphones

In attempt to prevent the problem of noise from the external electrical environment as was present in the initial design due to the choice of wiring for this design each microphone had two separate shielded twisted pair cables. This allowed for a single twisted pair for the signal going back to the data acquisition card and a single twisted pair going back to power for each microphone. Once the cables were properly shielded it resulted in a dramatic reduction in noise. Figure [FinalWiring](#) shows the microphones all cabled and screwed in.

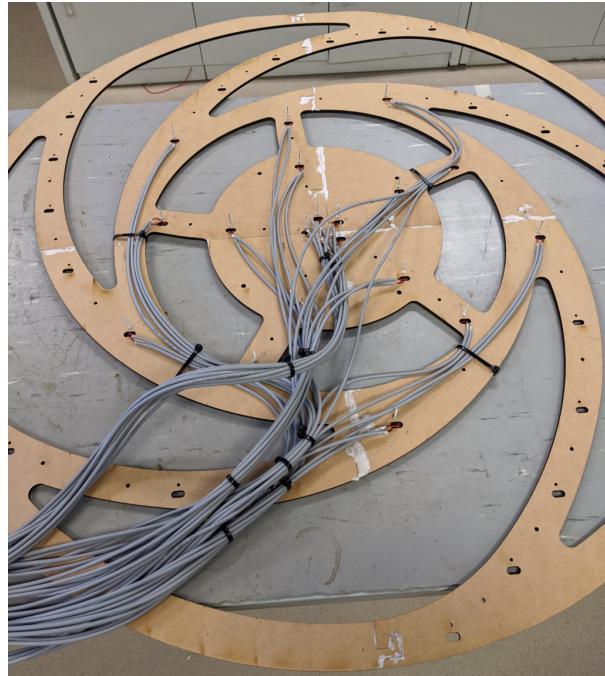


Figure 35: Final Microphone Array Wiring

A comparison of the intended final array and the used final array is shown in Figure 36, the designed final array has a much better response as would be expected with almost double the microphones but the final microphone array has an adequate response and is still preferred to the first design.

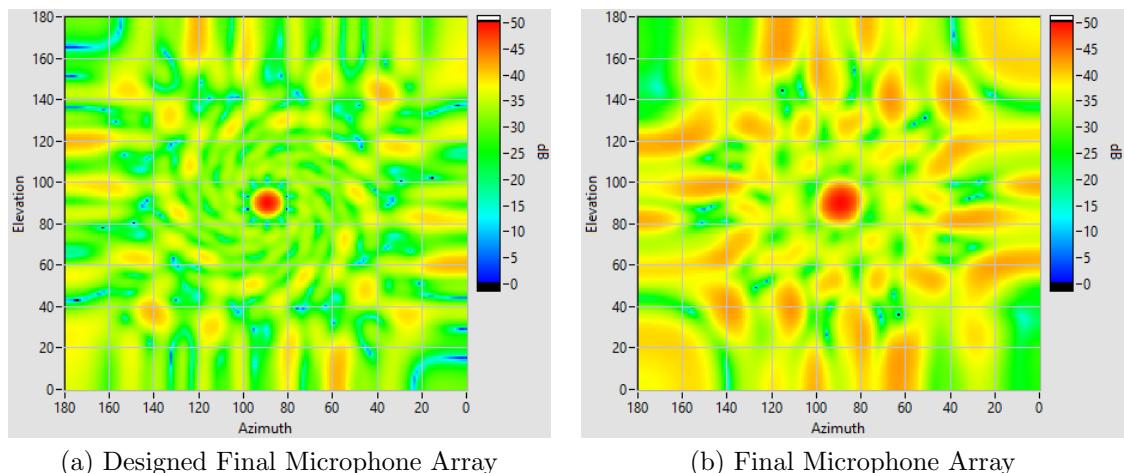


Figure 36: Comparison of Frequency Responses at 3kHz

## 6. Application to Machine Defects

Vibration analysis is used to evaluate the condition of machinery and determine whether defects due to misalignment, bearing degradation, looseness or any other potential factors that could be present. These defects generally cause vibration or impacting within the shaft or bearing housing that can be analysed and diagnosed, diagnosis requires the measurement of the vibration, velocity or displacement of the shaft or bearing housing over a period of time so that the time waveform and the frequency spectrum can be obtained and then analysed. Instead of using an accelerometer to measure the vibration it is possible to use a microphone as these defects generate noise the same way they generate vibration, with an array of microphones this will give the ability to pinpoint certain defects of machinery.

Focusing on the frequency spectrum as the features and patterns present can be directly related to those that would be picked up by a microphone array and therefore located. When analysing spectrums for defects there are various patterns and features to look out for that are typical of certain fault cases, some of these can be:

- Harmonic patterns
- Periodic events
- Sidebands
- Separation of close frequencies

Harmonic patterns are multiples of a specific frequency for example if there is a frequency at 25Hz it's harmonics would be 50Hz, 75Hz, 100Hz and so on. The first frequency (25Hz) is known as the fundamental frequency. Harmonics can be synchronous, sub-synchronous or non-synchronous. Synchronous includes the running speed frequency and all of its harmonics, sub-synchronous is below the running speed frequency and non-synchronous are all the frequencies between harmonics, all of these can indicate different types of faults. Common reasons for synchronous energy in the spectrum is imbalance, misalignment, looseness, blade/vane pass frequencies or gears, for sub-synchronous energy is commonly due to other machinery, hydraulic instability, or cage frequencies of bearings and for non-synchronous energy energy is commonly due to rolling element bearing defects, resonances, cavitations or other components and machines. [hand book](#)

Sidebands occur when there are two equally space frequencies on the higher and lower side of a frequency for example if there is a frequency a 125Hz and two more frequencies

at 130Hz and 120Hz these two frequencies would be called sidebands, specifically 10Hz sidebands. A single frequency with sidebands in a time waveform would look like amplitude modulation, and if there are also harmonics with sidebands present this would look like amplitude modulation with a distorted fundamental carrier wave. Common reasons for the appearance of sidebands are gear tooth wear, gear backlash, motor rotor faults or bearing ball defects. [mobius inst + handbook](#)

Knowing these common occurring defects of machinery that can be identified through the frequency spectrum with the addition of a microphone array beamforming with the MUSIC algorithm it is possible to determine locations of certain defects by targeting that particular set of frequencies, as well as this the sound can then be extracted from that location for further analysis.

## 7. Testing

### 7.1. Early Testing of First Design

Early testing involved myself whistling in-front of the array, but this did give promising results. For the final testing of the first design I was standing above the microphone array whistling while acquiring one second of data, using the delay and sum algorithm on this one produced the heat-map shown in Figure 37. This result accurately estimated the source to be right at the top of the array in the centre which is accurate to where I was, also present are two relatively major side-lobes in the bottom right quadrant of the heat-map.

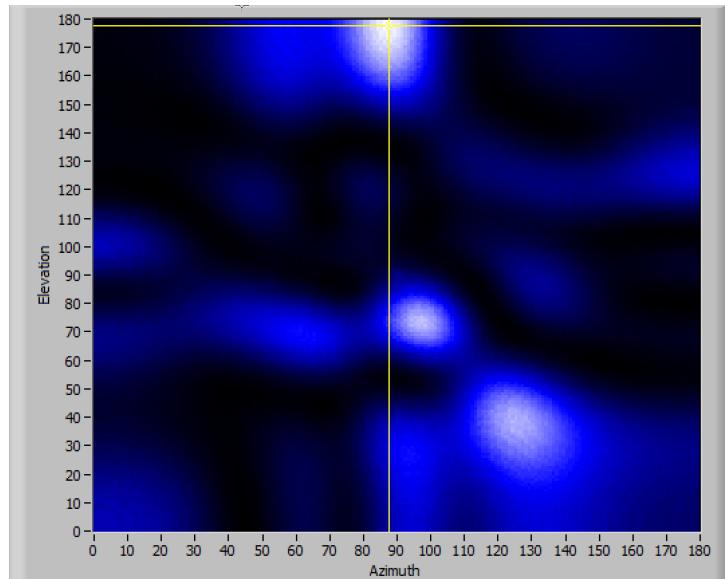


Figure 37: Result of Whistle testing

After performing a Fourier transform on the signal it was found that my whistle frequency was 1675Hz this is information as well as its spatial location was required to test the result in simulation and 3D frequency response for verification. Knowing the location and the frequency a 3D Frequency response was setup with those parameters producing Figure 38. As can be seen this produced a very similar looking heat-map which would indicate that the algorithm and hardware are both working on the test array as that would be the only way to produce such similar results. The main-lobe and side-lobes all look nearly identical between the real test and the 3D Frequency Response.

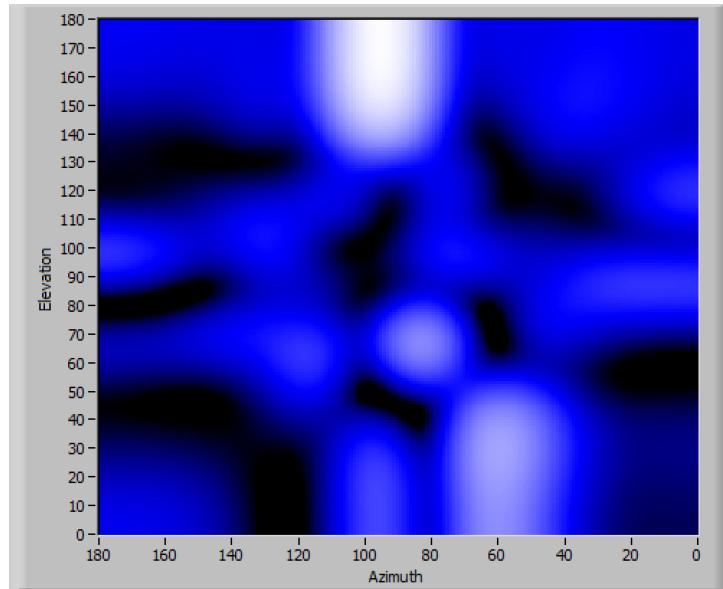


Figure 38: Whistle test 3D Frequency Response

To further validate the results from the test run the parameters were again used in the simulation. Figure 39 shows the results from this. Again showing the same and very promising results as the 3D Frequency response as was expected.

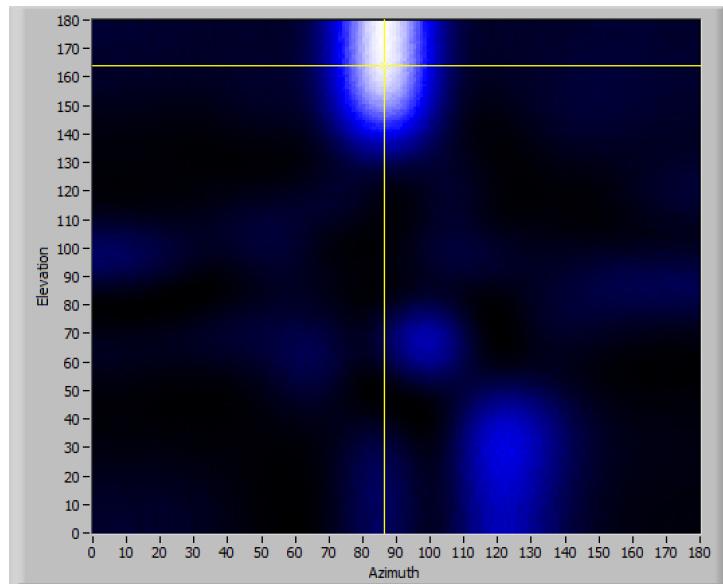


Figure 39: Whistle test Simulation

This whistle test and following simulations validated the array design and that it was working with delay and sum algorithm, though due to the external electrical noise in the

wiring and the small number of microphones no more testing was performed or required and moving onto the design of a larger array and improved algorithms were the next steps.

## 7.2. Final Design Testing

Before the final testing took place the array was tested to at the least be able to determine the correct side of the array a signal was arriving from, these results can be found in Appendix B. Following the success of the array to estimate the correct general location of incoming signals further test cases as described in the following sections were carried out.

### 7.2.1. Design Verification

To verify the accuracy and performance of the microphone array testing was performed in the lab using a speaker and a laser distance measurer. There were two tests performed one with a speaker at 70 Degrees Azimuth and the other with the speaker at 170 Degrees Azimuth from leftmost side of the array both at the centre of Elevation, these angles were calculated with basic trigonometry using the laser distance measurer, the 70 Degree test was 4m away from the origin of the array whilst the 170 Degree test was 3m from origin.

Figure 40 displays the results of the delay-and-sum algorithm to both of the test, this algorithm was chosen for the purpose of verifying the array due to it being much less error prone than MUSIC with less need for user input and has previously been tested and shown to work with a real array. The results show that the array does indeed operate as expected as well as the algorithm. The 70 Degree test was close to exact along the azimuth and was estimated to be a few degrees above its true elevation. Inaccuracies begin to show at the edge of the array for the 170 Degree test with the azimuth estimation being around 10 Degrees off but the elevation estimation was as expected.

If we refer back to Figure 20 the reason for the inaccuracies at the edge of the array can be explained, as the array is steered towards the edge the main-lobe of the 2D frequency response becomes larger in the exact same way as is seen in the 170 Degree test. This can be attributed all the way back to Eqn. 4.4 where it was determined that the time difference of arrival for signals is a cosine function and hence as the angle of arrival approaches either 0 or 180 Degrees the time difference of arrival between microphones

becomes increasingly smaller and increasingly difficult to separate between angles creating the large main-lobe as steering towards the edges of the array as seen in Figure 20 and 40, this will occur in both the delay-and-sum and the MUSIC algorithm as it comes to the characteristics of sound waves. To show the difference between two points on an array results from two calculations using Eqn. 4.4 are shown in Table 1, this table shows the difference in the time difference of arrival when moving 1 degree across the azimuth towards centre from both 40 Degrees and 170 Degrees, the difference in time delay being much smaller when at 170 Degrees.

Table 1: Array Time Delay Comparisons

Angle 1	Angle 2	$\Delta\tau_{1,2}$	$\Delta\%_{1,2}$
40	39	$1.616e-05$	-
170	169	$4.632e-06$	-28.66

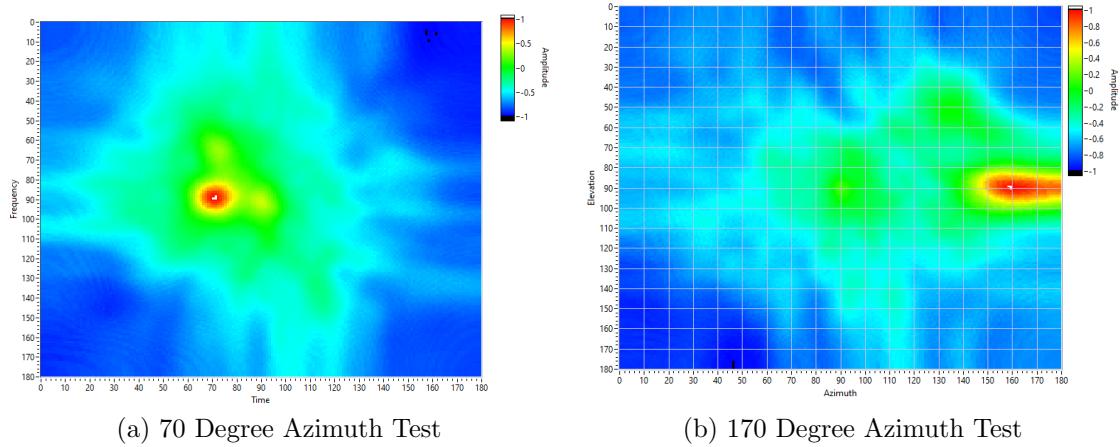


Figure 40: Array Verification Test

### 7.2.2. Algorithm Comparison

The next round of testing involved testing and comparing the accuracy of both the delay-and-sum and MUSIC algorithm by comparing the results to an identical sound source. Again using the laser distance measurer a speaker was placed at 47 Degrees Azimuth and centre of the Elevation playing a 1871Hz tone.

For both of the results shown in Figure 41 the algorithms were using the exact same 0.1s snippet of sound, the MUSIC algorithm just processed for the exact 1871 frequency

whilst delay and sum is required to process the full snippet. The delay and sum resulted in a 3 Degree accuracy to the true source location and the MUSIC algorithm resulted in a 2 Degree accuracy, the resolution of the MUSIC algorithm is much higher though as well as having a significantly lower noise and number of side-lobes present. As well as showing better resolution and accuracy the MUSIC algorithm was over 10 times as fast as the delay and sum algorithm further cementing its place for determine the location of specifically chosen frequencies.

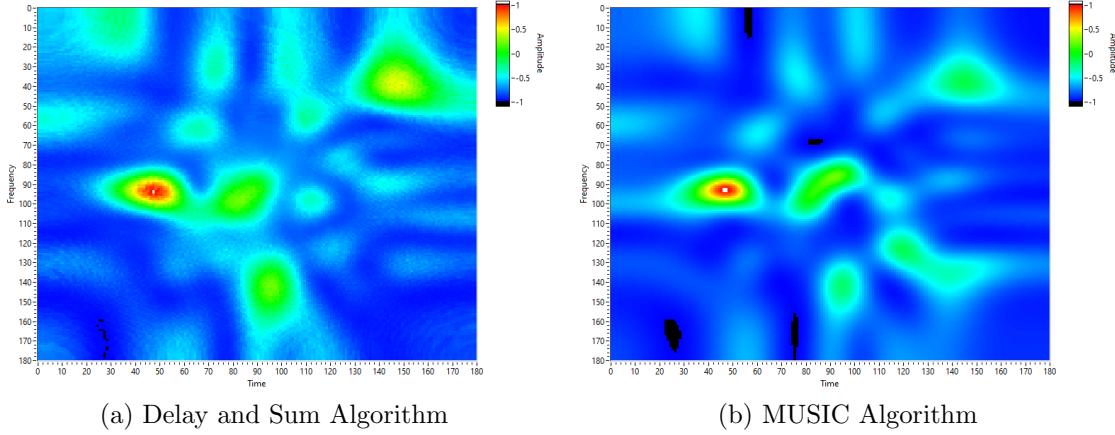


Figure 41: Real Algorithm Comparison to 1871Hz Signal

### 7.2.3. Defect Detection

Now to verify the ability of the array to differentiate different noise and spatially locate them a simple test was performed in a lab environment. To the right hand side 3m from the array a speaker was playing a loud 2.5kHz sinusoidal tone and in front of the array to the left 8m away was a turbine balance test rig running at 3000rpm, the purpose of the test was to see if it would be possible to differentiate the sources.

Figure 42 shows the signals captured by the microphone array over a 4 second period. As can be seen in the frequency spectrum, there is a significant amount of noise in the sub 1kHz frequency range, a large spike at 2.5kHz, a small bunch of signals at 5kHz and 6kHz and more noise around 9-11kHz. As the lab was relatively quiet the peaks that appeared at 5kHz and 6kHz were assumed to be coming from the balancing test rig, Figure 43 provides a closer look at some of the assumed signals from the balancing test rig.

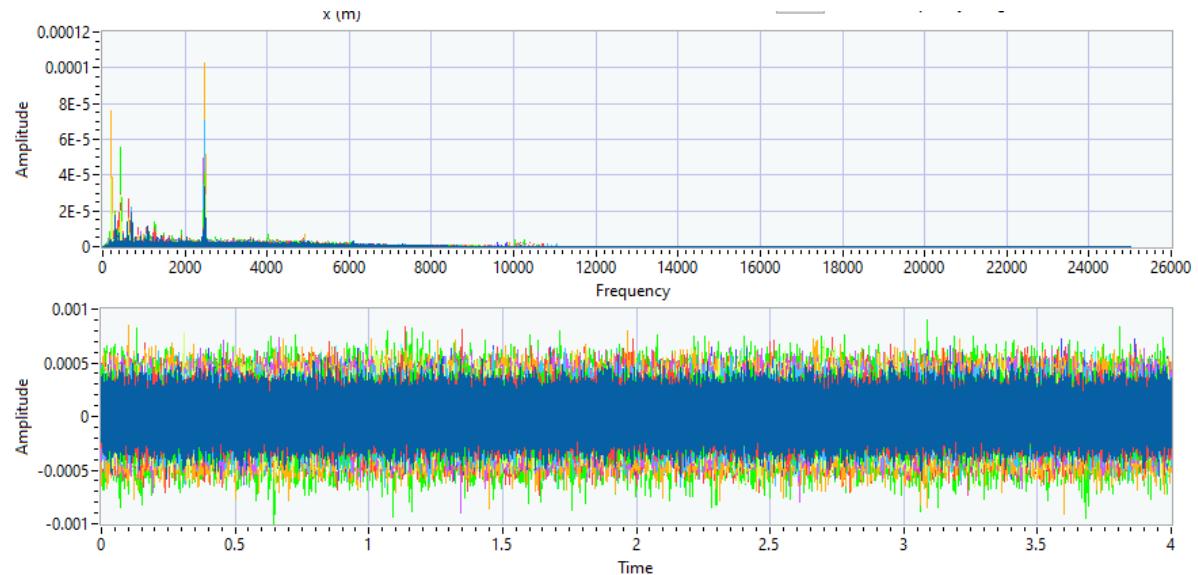


Figure 42: Microphone Signals

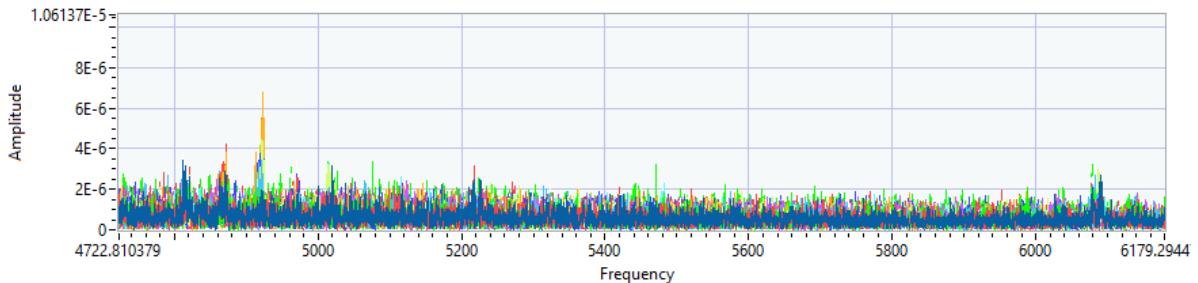


Figure 43: Zoomed Microphone Signals

In the following Figure 44 it is shown that the array succeeded in differentiating the signal source locations and identified the correct locations for the signals. The color scale for the intensity map was changed for Figure 44 for less confusion when interpreting the result, instead of showing all side-lobes and the main-lobe this scale only shows the main lobe and highest side-lobes if they are present. In Figure 44(a) a single frequency of 2474Hz was searched for using the MUSIC algorithm and for Figure 44(b) a set of frequencies was used to search for, these being: 4812Hz, 4870Hz, 4921Hz, 6080Hz, and 6090Hz, these being the prominent signals that are present in Figure 43. As shown this method of differentiating signals by selecting certain frequencies worked well, the 2.5kHz signal was located accurately though the estimated area is relatively large, on the other-hand the turbine balance test rig was estimated very precisely and accurately though there is a large level of side-lobes creating some uncertainty within the results. Impressively the

array could pick out the turbine test rig using the particular frequencies even when they were barely above the noise floor of the environment.

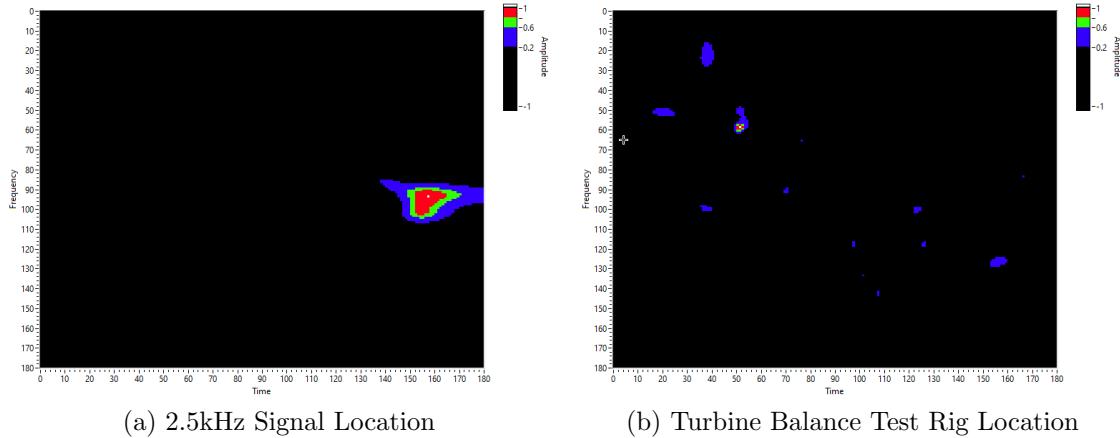


Figure 44: Differentiate signal testing results

This test shows promising results with the array proving that is able to differentiate particular noise sources by frequency and even when the particular frequencies of interest are only just above the noise floor. As well this test has reinforced the potential of the microphone array for finding defects in machine by using a particular set of frequencies to pin-point spatially.

### 7.3. Site Testing

## **8. Recommendations**

## 9. Conclusion

more industrial testing, fpga, more microphones etc would be better The main goal of this project is to design an acoustic array that is to be used for the early detection of defects in machinery. This goal is looking promising and achievable, with the current delay-and-sum algorithm implemented already, by just increasing the number of sensors and improving the geometry of the array there will be noticeable improvements to the beamforming accuracy and resolution and with some fine-tuning it should be use-able on machinery. This combined with an FPGA or new algorithms will decrease computational load and potentially lead to better results as new algorithms will lead to further refinement.

In conclusion the current progress towards an acoustic array for condition monitoring is promising. With the next portion of work being devoted to refinement and expansion of ideas, and further signal analysis techniques.

## References

- Adafruit, n.d. Adafruit silicon mems microphone breakout - spw2430.  
URL <https://www.adafruit.com/product/2716>
- Adve, R., 2013. Direction of arrival estimation. University of Toronto.  
URL <https://www.comm.utoronto.ca/~rsadve/Notes/DOA.pdf>
- Amin, S., Ahmed-Ur-Rahman, Saabah-Bin-Mahbub, Ahmed, K. I., Chowdhury, Z. R., 2008. Estimation of direction of arrival (doa) using real-time array signal processing. In: 5th International Conference on Electrical and Computer Engineering.
- Bakker, S., n.d. Condition monitoring of rotating machines.  
URL <https://www.istec.com/en/condition-monitoring-rotating-machines/>
- Benesty, J., Chen, J., Huang, Y., 2008. Microphone Array Signal Processing. Vol. 1. Springer.
- Greensted, A., 2019. The lab book pages.  
URL <http://www.labbookpages.co.uk/audio/beamforming/delaySum.html>
- Krishnaveni, V., Kesavamurthy, T., Apama.B, 2013. Beamforming for direction-of-arrival (doa) estimation-a survey. International Journal of Computer Applications 61.
- Laws, W. C., Muszynska, A., 04 1987. Periodic and Continuous Vibration Monitoring for Preventive/Predictive Maintenance of Rotating Machinery. Journal of Engineering for Gas Turbines and Power 109 (2), 159–167.  
URL <https://doi.org/10.1115/1.3240019>
- MathWorks, 2020. Music super-resolution doa estimation.  
URL <https://www.mathworks.com/help/phased/ug/music-super-resolution-doa-estimation.html>
- McCowan, I., 2001. Robust speech recognition using microphone arrays.
- National-Instruments, 2019. C series sound and vibration input module.  
URL <https://www.ni.com/en-au/shop/select/c-series-sound-and-vibration-input-module?modelId=122184>
- National-Instruments, 2020. C series voltage input module.  
URL <https://www.ni.com/en-au/support/model.ni-9229.html>

National-Instruments, n.d. Acquiring an analog signal.

URL [https://download.ni.com/evaluation/pxi/Acquiring\\_Analog\\_Signal.pdf](https://download.ni.com/evaluation/pxi/Acquiring_Analog_Signal.pdf)

National-Instruments, 2019. cdaq-9178.

URL <https://www.ni.com/en-au/support/model.cdaq-9178.html>

Näsholm, S. P., 2012. Adaptive beamforming.

URL [https://www.uio.no/studier/emner/matnat/ifi/INF5410/v12/undervisningsmateriale/foils/adaptive\\_beamforming.pdf](https://www.uio.no/studier/emner/matnat/ifi/INF5410/v12/undervisningsmateriale/foils/adaptive_beamforming.pdf)

Orman, M., Pinto, C. T., Oct 2013. Usage of acoustic camera for condition monitoring of electric motors. In: 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013). pp. 1–4.

Paulraj, A., Ottersten, B., Roy, R., Swindlehurst, A., Xu, G., Kailath, T., 1993. 16 subspace methods for directions-of-arrival estimation. In: Signal Processing and its Applications. Vol. 10 of Handbook of Statistics. Elsevier, pp. 693 – 739.

URL <http://www.sciencedirect.com/science/article/pii/S0169716105800823>

Prime, Z., Doolan, C., 2013. A comparison of popular beamforming arrays. In: Proceedings of ACOUSTIC 2013. Australian Acoustical Society.

Sanders, M., 2017. Study on the application of digital mems microphones for aerocoustic noise source localization for large drones.

Schaedler, J., 2017. Circles, sines, and signals.

URL <https://jackschaedler.github.io/circles-sines-signals/s>

Shahbazpanahi, S., Jing, Y., 01 2018. Recent advances in network beamforming. pp. 403–477.

SM-Instruments, n.d. Portable sound camera seesv-s205.

URL <http://smins.co.kr/en/products/portable-sound-camera/portable-sound-camera/>

Tang, H., 2014. Doa estimationg based on music algorithm.

URL <http://www.diva-portal.org/smash/get/diva2:724272/FULLTEXT01.pdf>

Travis, J., Kring, J., 2006. LabVIEW for Everyone: Graphical Programming Made Easy and Fun, 3rd Edition. Prentice Hall.

Weisstein, E., n.d. Orthogonal vectors.

URL <https://mathworld.wolfram.com/OrthogonalVectors.html>

Yan, S., 2005. Broadband beamspace doa estimation: Frequency-domain and time-domain processing approaches. EURASIP Journal on Advances in Signal Processing 2007.

## A. Simulation Figures

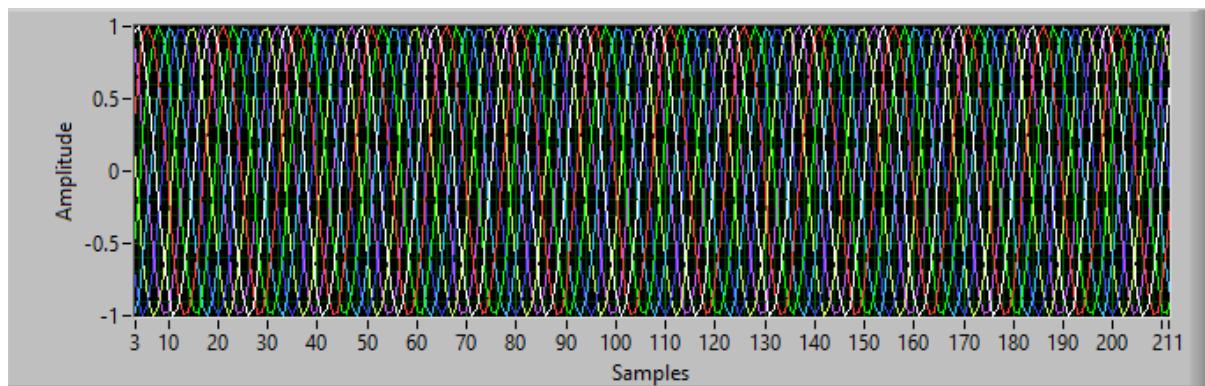


Figure 45: Array of signals when Steered to the Centre of the Array

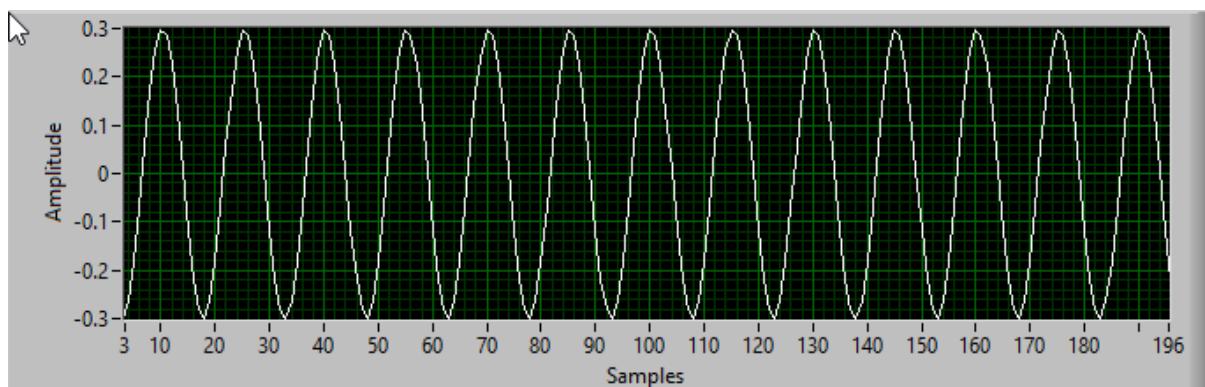


Figure 46: Array of signals at Centre Summed and Normalised

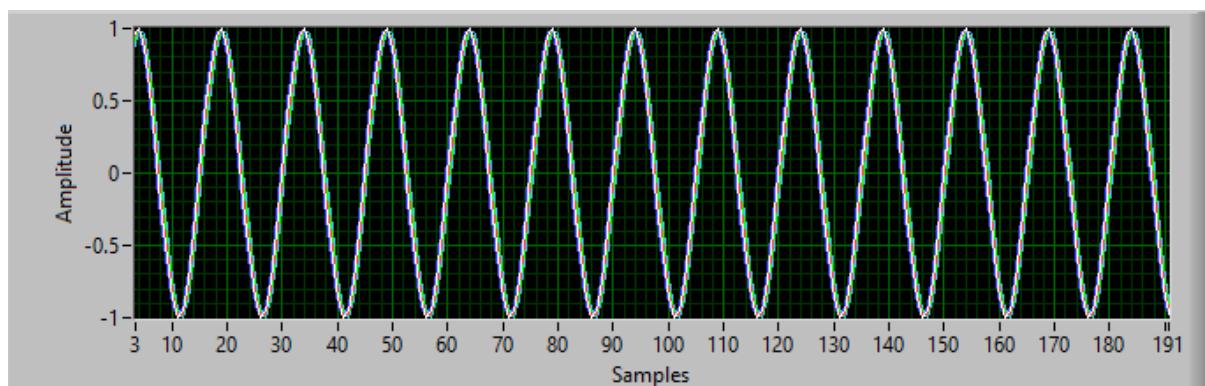


Figure 47: Array of signals when Steered to the Source Location

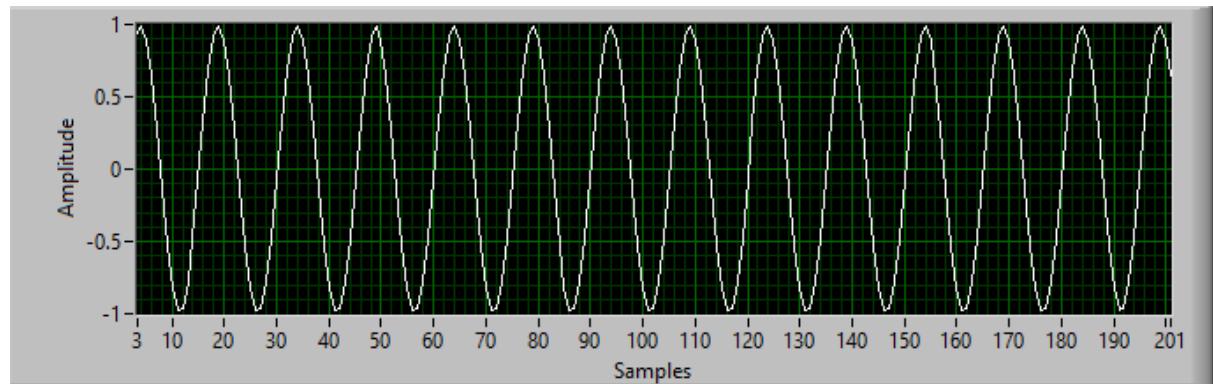


Figure 48: Array of signals at Source Location Summed and Normalised

## B. Testing Figure

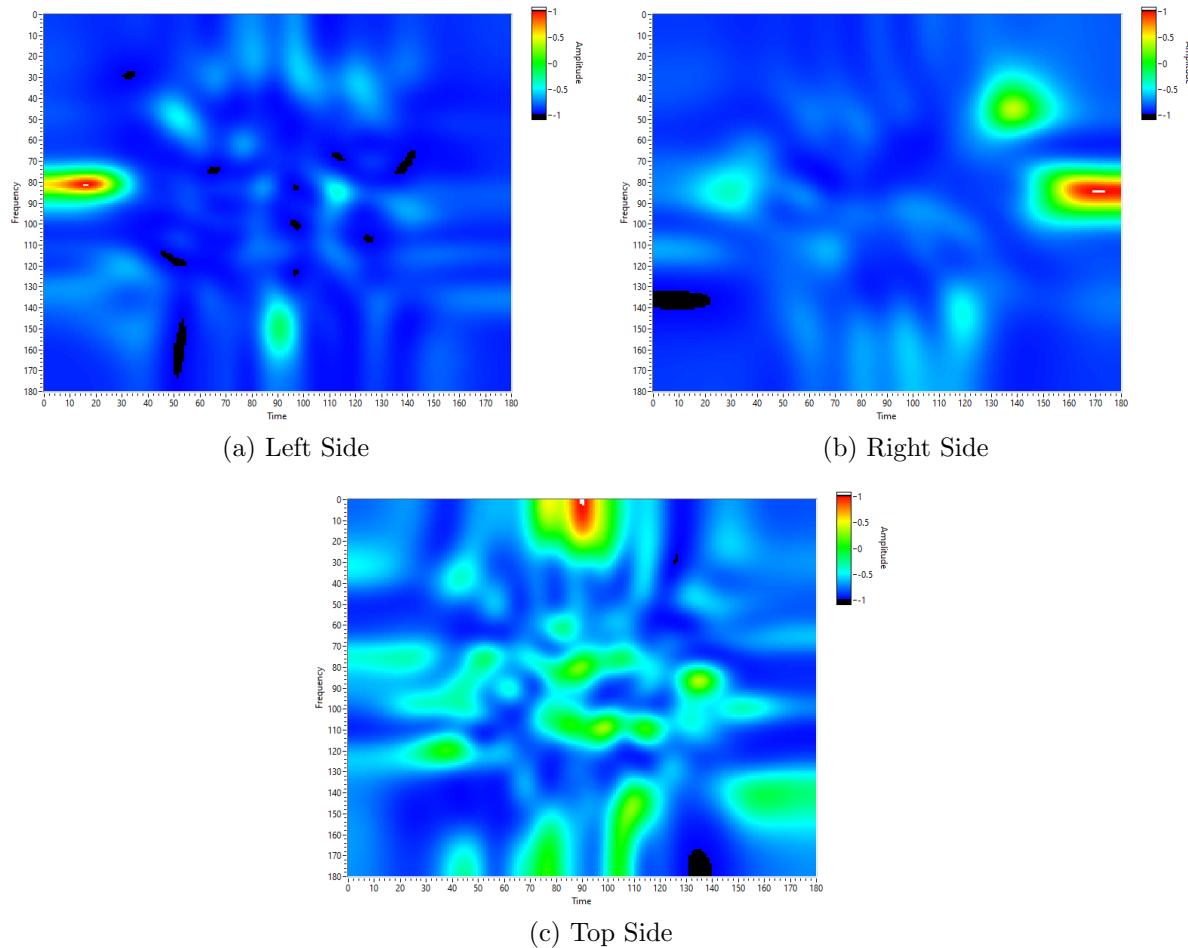


Figure 49: Initial Calibration Results