

Towards Targeted Acoustic Beamforming: Enabling Directionality in Acoustic Interference Patterns using LIDAR-Driven Phased Speaker Arrays

Jairo E. Huaylinos Lozano^a, Aniq Islam^a, Mathew Erickson^a, Emma Griffith^a, Justin Wang^a

^aDepartment of Electrical and Computer Engineering, University of Maryland, 8223 Paint Branch Drive, College Park, 20740, Maryland, U.S.A.

Abstract

We present an application of LIDAR-targeted phased-speaker arrays to the problem of acoustic beamforming. The system ingests two parameters —the distance from the speakers’ center, and angle from the mounting plate. After rigorous quantitative and qualitative characterization, the speakers’ radiation pattern and acoustic behavior conforms closely to simulation and expert intuition. This intermediate success enables our automation of the former manual parameter ingestion, i.e., LIDAR-driven targeting. Our system realization is highly performant and aligns closely to simulation; moreover, the LIDAR-driven aspect further convincingly demonstrates that the phased-speaker delays are both calculated and implemented accurately.

Keywords: Beamforming, LIDAR, interference patterns, speakers

PACS: 0000, 1111

2000 MSC: 0000, 1111

Nomenclature

θ	Angle between phased array and observer
A	Edge-to-Edge distance between speakers (m)
D	Distance from center of array to listener/observer (m)
f_s	Sampling frequency of the EVAL-ADAU1467Z DSP board (48 kHz)
v_s	Speed of sound (343 m/s)
W	Width of speaker (m)

1. Introduction

Phased array systems have revolutionized antenna technology due to their numerous advantages over mechanically steered antennas. Our project aims to provide a comprehensive learning experience that equips us with practical skills and knowledge directly applicable to our long-term aspirations in industry and research. Our approach involves building an 8-speaker phased array system that utilizes Digital Signal Processing (DSP) algorithms to direct sound waves to any direction without mechanical repositioning. To achieve this goal, we utilize the EVAL-ADAU1467Z DSP board and develop custom beamforming software using SigmaStudio, a user-friendly graphical software development tool. In addition to the software, we have designed, manufactured, and assembled a custom printed circuit board (PCB) equipped with audio amplifiers to drive our 8 speakers. Our system is highly functional, with significantly enhanced audio in the targeted direction up to 15dB vs. non-targeted directions. Through this project, we gained hands-on experience in hardware and software design, manufacturing, and testing, which will be valuable in our future pursuits in the field of electrical engineering.

1.1. Contents

The rest of the paper is organized as follows. Section 2 provides a survey of the background research that we have done on the theory of beamforming. Section 3 is a self-contained treatment of the mathematics and theory behind the radiation pattern and acoustics. In Section 4, we discuss the hardware elements that are designed, created, and implemented in our realization. After all the design details have been concluded, we share the set-up, test procedures, and simulation models involved in the verification of our hardware/software interface of our final design in Section 5. Section 6 contains a thorough treatment of the results of our experiments and a comparison to the expected results that our simulations yield. Finally, we end the paper with Section 7, in which we provide a heuristical analysis regarding our level of success and present our lines of future work.

2. Background

Phased array systems have gained widespread popularity due to their ability to steer and direct the radiation patterns of antennas without the need for mechanical repositioning. This technique finds its application in various fields such as wireless communications [1], medical imaging [2], and sonar systems [3]. In recent years, phased array systems have also been used in the field of acoustics, where they have been found to be highly effective in controlling the directionality of sound waves [4].

Our project focuses on designing and implementing an 8-speaker phased array system using DSP techniques. The use of DSP in phased array systems has gained significant attention in recent years, with researchers exploring various algorithms and techniques to improve the performance of these systems[4][5].

One popular technique used in phased array systems is beamforming. Beamforming involves the manipulation of the phase and amplitude of the signals emitted by individual elements in a

phased array to steer the direction of the resultant wavefront. Various beamforming algorithms have been developed to optimize the performance of phased array systems.

In our project, we utilize the Delay and Sum beamforming algorithm due to its simplicity and effectiveness in directional sound control. One of the most influential factors that we had to consider in order to help us narrow down which algorithm to use was hardware capabilities of the DSP board itself. Many of the other algorithms that we looked into required the use of modules that weren't supported in Sigma Studio for our particular DSP board. All the components that we needed to implement the Delay and Sum algorithm are supported, which further informed our choice.

The Delay and Sum algorithm involves a two stage process. In the first stage, calculations have to be performed in order to determine the differences in distance that the traveling waves from each individual speaker takes, to reach a certain reference point of the user's choice. This involved using trigonometry and included variables such as the speakers' diameter, separation distance between speakers, the angle between the observer and the phased array, etc. After this calculation, the second stage involved converting the differences in distance to a delay in samples being sent out from each speaker. By carefully choosing how many samples to delay each speaker by, we are able to sum up all the constructive interference patterns at a specific location, allowing sound to be amplified at that reference point. This in turn allows us to form a focused beam of sound in a direction of our choice, while also attenuating the sound in other directions.

Additionally, to power the 8 speakers in our system, we designed and assembled a custom PCB equipped with audio amplifiers. The PCB design involved considerations such as minimizing noise and optimizing power delivery to the speakers.

In summary, our project aims to design and implement a functional phased array system for directional sound control using DSP techniques. The use of beamforming algorithms and the EVAL-ADAU1467Z DSP board in our implementation demonstrates the potential of DSP in improving the performance of phased array systems. The design and assembly of a custom PCB also highlights the importance of hardware design considerations in achieving optimal system performance.

3. System Design and Theory of Operation

3.1. System Block Diagram

To create the overall system, an audio input is fed into a DSP board. It is split into 8 outputs, each with a separate phase delay, which is fed into the DSP by a Raspberry Pi. The 8 outputs of the DSP correspond to the 8 inputs of the amplifier PCB, and then output to each of the speakers in the linear array. A diagram of the overall system design can be seen in Figure 1.

3.2. Delay Calculation Theory

First we must determine the variables that go into the beamforming algorithm. These include geometric measurements that help determine the relative positioning of the speakers, both in relation to each other as well as in relation with the observer. Now, assume without loss of generality that the center of the phased array forms the origin of a 2-D Cartesian plane. (The interested reader may prove that there is a unique isometry from an arbitrary coordinate axes to a fixed target point, which we take to be

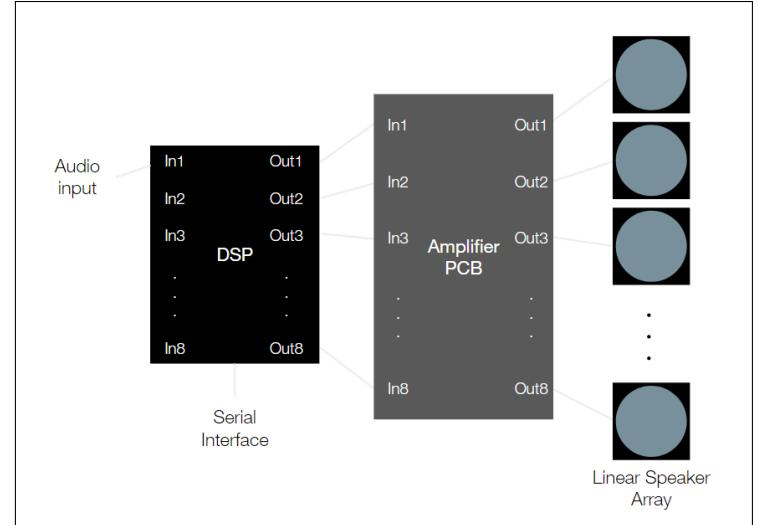


Figure 1: Diagram of overall system design

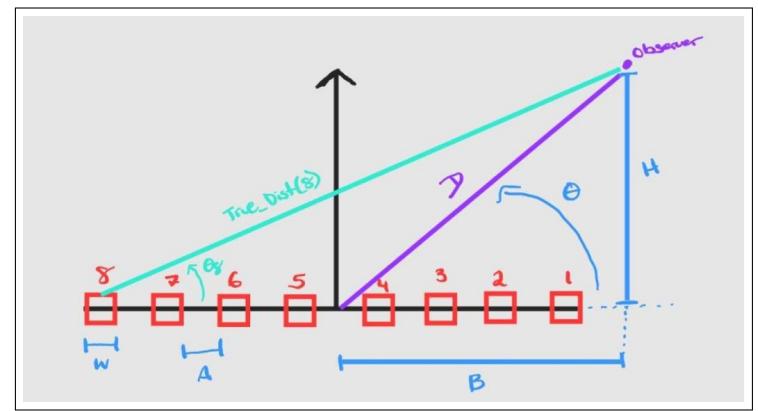


Figure 2: Geometry of The Linear Phased Array

the center of the speaker array.) The next task is to determine the x-value and y-value of the Cartesian point that is represented by the listener's position. The x-value can be determined by the following:

$$B = D \cos \theta \quad (1)$$

However for the sake of our calculations, we will only consider the absolute value of this quantity:

$$B = |D \cos \theta| \quad (2)$$

The y-value of the listener's position can be obtained with the following:

$$H = D \sin \theta \quad (3)$$

With these values, we can determine the value of the angle that is created between the phased array and the line segment that connects each individual speaker with the observer's position. Let this quantity be called $\theta(i)$, which represents the aforementioned angle for the i -th speaker:

$$\theta(i) = \arctan \frac{H}{\max(B, (3.5 - (i - 1)) * (W + A)) - \min(B, (3.5 - (i - 1)) * (W + A))} \quad (4)$$

In Figure 2, $\theta(8)$ has been shown as an example of what this measurement is for speaker $i = 8$. With these values of $\theta(i)$, we can calculate the actual distance of that line segment between each

speaker and the observer. We can label this quantity as $Dist(i)$:

$$Dist(i) = \frac{H \sin \theta}{\sin \theta(i)} \quad (4)$$

Because we took the absolute value when calculating B, our model works well for $\theta \in (0, 90]$ degrees. Assume that all the $Dist(i)$'s are stored in an array of size N ($N = 8$ for our experiment, as we have 8 speakers). If the desired θ of the listener is within the range of [91, 180), then by the symmetry of the array, all we have to do to get the true values of $Dist(i)$'s is to calculate the delays for $180 - \theta$, and then perform a reversal of the array such that the first element and last element are swapped, the 2nd element and $(N - 2)$ element are swapped, and so on. Essentially, our `True_Dist(i)` becomes:

$$True_Dist(i) = \begin{cases} Dist(i), & \text{if } \theta \in (0, 90] \\ Dist(N+1-i), & \text{if } \theta \in [91, 180) \end{cases} \quad (5)$$

Now with the `True_Dist` array, we can find which speaker will be selected to be the reference for all our delay elements. The speaker that delivers its sound wave to the listener last, i.e. has the longest `True_Dist` value, will be the reference speaker. This speaker will have a delay of 0 seconds/samples.

$$Ref = \max_{i \in [1, 8]} True_Dist(i) \quad (6)$$

To calculate the delay time needed for every speaker, we simply perform the following calculation which takes into account the speed of sound under nominal settings. All the delay times for each speaker will also be stored in an array, `Delay_Time`:

$$Delay_Time(i) = \frac{Ref - True_Dist(i)}{v_s} \quad (7)$$

Finally to get the fractional delay in samples for each of the speakers, we consider the sampling frequency of our DSP board. These values will be stored in their own array, `Delay_Samp`:

$$Delay_Samp(i) = Delay_Time(i) * f_s \quad (8)$$

With these values, we can begin implementing an FIR design in order to form the desired beam.

3.3. Software Implementation

The FIR filter that implements a fractional delay is condensed into a single block within the SigmaStudio software as a "Fractional Delay" module. From the SigmaStudio side of the software we simply have to feed each of the signals that get fed into the speakers through a fractional delay module as can be seen in Figure 3.

Even though there are default values for the delays in the firmware, producing a fixed beam, we must change them to change our beam angle. To calculate delays and send them to the DSP board in one step, a Raspberry Pi was used as the interface between a PC and the DSP board. The DSP board uses Serial Peripheral Interface (SPI) to communicate with its programmer as well as external computers/microcontrollers. SPI is a full-duplex serial interface commonly used for communication between microcontrollers and peripherals [6]. Generally, consumer PCs do not have exposed SPI interfaces, although they are present in many peripherals on-board common laptop computers [7]. While there may exist adapters from USB to SPI (and

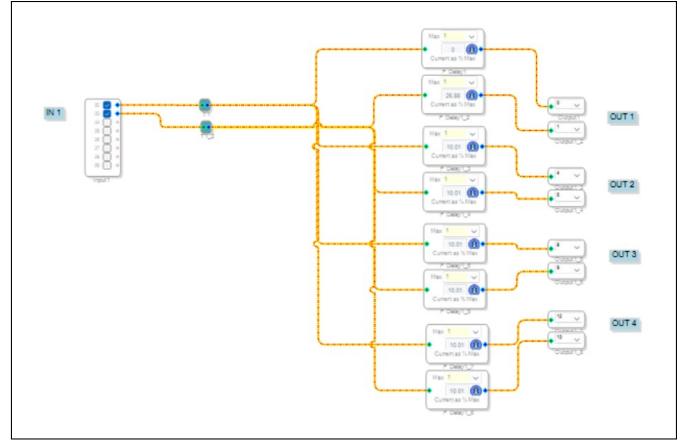


Figure 3: SigmaStudio Schematic for Phased Array

we may have inadvertently already been in possession of one,) the easiest solution was to use a computer meant for embedded purposes, like a Raspberry Pi, which already has a SPI interface exposed [6]. The DSP board's firmware is written to the board's EEPROM by its programmer via the SPI port from SigmaStudio, but the parameters are at default values until the board is booted again with the Raspberry Pi connected. A script was written on the Raspberry Pi that calculates the delay time for each individual speaker based on the user parameters, as previously mentioned. The SPI protocol is implemented as described in [8]. From SigmaStudio, the addresses of parameters are obtained from the register map. Data is sent in the 8.24 fixed-point format to represent fractional delay values. A maximum delay is set at 100 samples in our SigmaStudio Firmware. Thus, the resulting fraction that we send is a 24-bit fixed-point value representing the proportion of 100 samples by which we delay each speaker. These values are then sent to the DSP board by a *Safeload Write* command, which safely loads parameters into the DSP without the risk of the digital filter becoming unstable due to the abrupt coefficient change [8]. If the filter coefficients were to change as filtered outputs are computed, the behavior is not well-defined. In our case, changing delays during processing would result in an undefined delay during transitions. The *Safeload Write* procedure avoids this by loading the commands into a buffer and writing them to the DSP when processing is not taking place [8]. The parameters written are the data, the desired address of the data, and the length of the data (number of 32 bit words) in that order. A digital logic analyzer (DLA) capture of a safeload write of 0x00 to address 0x15 of length 1 word is seen in Figure 4. This operation was verified by observing the output of the programmer writing a parameter and verifying that the output of the SPI bus on the Raspberry Pi is identical for that same parameter as an input.

This protocol was implemented using the Linux spidev Python wrapper, which abstracted the SPI timing and DMA operations. The output of our program to change the delays sent to the DSP board based on an observer angle and distance was that seen in Figure 5

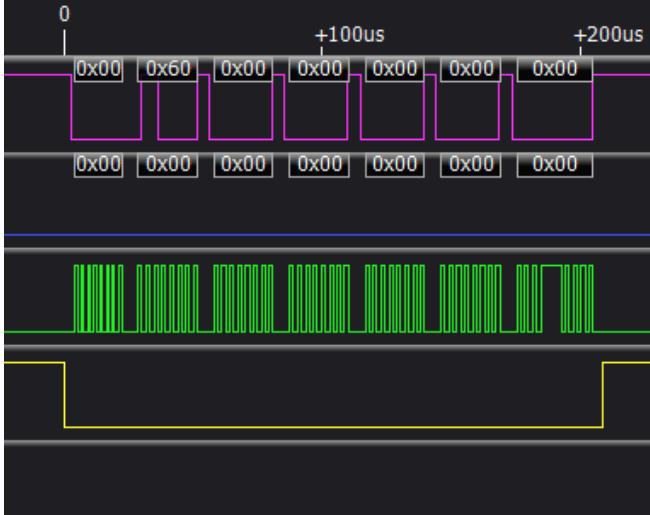


Figure 4: DLA capture of the SPI transaction format. It is apparent from the clock phase and polarity that the chip operates in SPI mode 0 [6].

```
pi@raspberrypi:~/ENEE408J $ python Change_Delays.py
Angle in degrees of desired beam direction:
45
Distance of listener in meters from center of array:
1
Addresses of parameters to change: [ 0x15 0x17 0x19 0x1b 0x1d 0x1f 0x21 0x23 ]
Data Message: [ 0x0 0x60 0x0 0x0 0x62 0x53 0xd9 ]
Wrote delay of [38.4091961] samples to speaker 1
Data Message: [ 0x0 0x60 0x0 0x0 0x57 0x2a 0x82 ]
Wrote delay of [34.04923898] samples to speaker 2
Data Message: [ 0x0 0x60 0x0 0x0 0x4a 0xc5 0xb4 ]
Wrote delay of [29.20795726] samples to speaker 3
Data Message: [ 0x0 0x60 0x0 0x0 0x3d 0x55 0x1c ]
Wrote delay of [23.95799508] samples to speaker 4
Data Message: [ 0x0 0x60 0x0 0x0 0x2f 0x2f 0x17 ]
Wrote delay of [18.36256695] samples to speaker 5
Data Message: [ 0x0 0x60 0x0 0x0 0x1f 0xef 0xf4 ]
Wrote delay of [12.47551523] samples to speaker 6
Data Message: [ 0x0 0x60 0x0 0x0 0x10 0x3c 0x59 ]
Wrote delay of [6.34208476] samples to speaker 7
Data Message: [ 0x0 0x60 0x0 0x0 0x0 0x0 0x0 ]
Wrote delay of [0.] samples to speaker 8

Angle in degrees of desired beam direction:
'C
Quitting...
```

Figure 5: Output of Python program to change speaker delays over SPI.

4. Hardware Design and Implementation

4.1. Amplifier Circuit

In order to drive our 8-speaker setup, an 8-channel amplifier circuit was implemented to reduce the space taken up by individual amplifier boards. The required specifications for each amplifier channel were as follows:

- Provide at least 20dB voltage gain driving an 8Ω , 4W speaker load
- At maximum power/gain, drive the load with a total harmonic distortion of < 1% across the audio frequency range
- Individually controllable speaker volume
- Relatively flat magnitude/phase response across the audible range
- Ability to turn individual speakers on and off for testing

Due to the other complexity involved in the project, we elected to use an off-the-shelf audio amplifier integrated circuit (IC) to avoid the debugging process of designing one's own amplifier circuit. The Texas Instruments TPA1517NE [9] was selected based upon these criteria.

Additionally, we elected to design our own custom PCB for this amplifier. While the IC was the correct package (DIP-20) for use in a breadboard, the use of a PCB had several benefits. PCBs enable the use of small surface-mounted components, decreasing required area. Additionally, parasitics can be much more easily controlled when designing one's own circuit layout. Namely, signal and power trace lengths can be sized according to impedance and current requirements, and paths to ground can be shortened dramatically due to the two-layer nature of most common PCBs. The schematic of the PCB is equivalent to this schematic found in [9], reproduced in Figure 6.

The layout for this board was designed with a few factors in

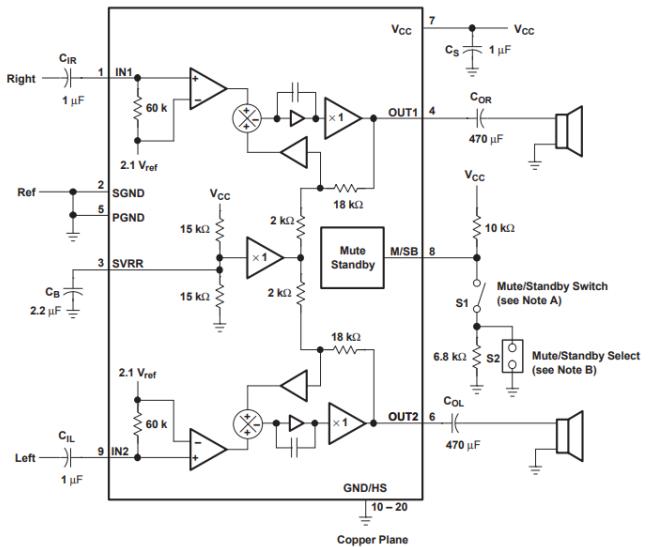


Figure 6: Amplifier circuit schematic with internal block diagram [9, Fig. 23]

mind. As this is a Class AB amplifier, transistors will constantly be biased such that current will flow through them. Therefore the amplifiers have a relatively low efficiency and will dissipate a fair amount of heat, and must be treated accordingly. Thus, power and ground traces are very thick, and 10 pins of the IC are dedicated solely to heat-sinking. These pins are connected to two large ground pours, one on the bottom layer and another on the top layer, filling in areas where signal traces do not exist. The layouts, created in Altium Designer, are shown in Figure 7 and Figure 8. Signal trace lengths are kept to a minimum, which is much, much shorter than a wavelength in the audible frequency range. The bottom consists of a solid ground plane which creates a low-impedance node and should help to reduce the effects of ground loops, which are often the source of noise in audio equipment.

We found that it is unnecessary to do any kind of controlled impedance traces, and trace length mismatch will not cause any appreciable phase shifting to a waveform in the audio frequency range. This reduced board complexity and cost. The board was assembled by a combination of hand soldering and hot-air reflow.

Although the ICs have integrated heat-sinking to two large

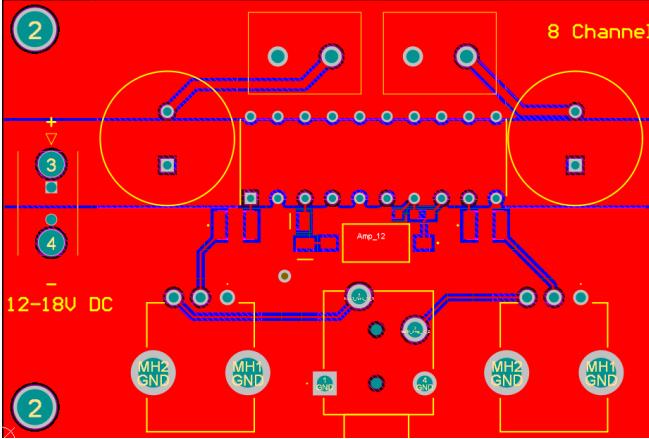


Figure 7: One-chip section of 8-channel amplifier PCB, top layer.

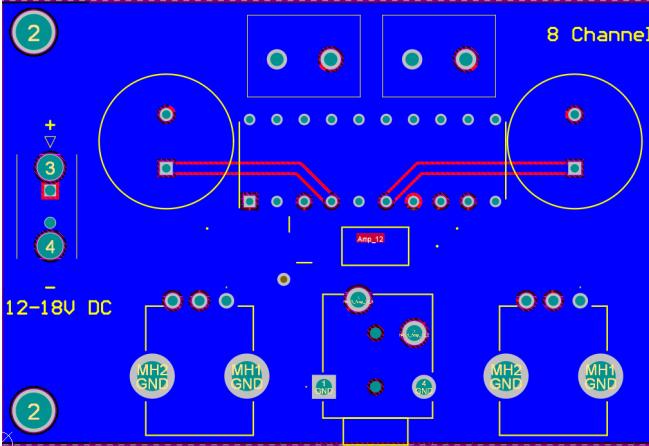


Figure 8: One-chip section of 8-channel amplifier PCB, bottom layer.

ground pours as previously mentioned, initial power-on tests of the board revealed that the package body itself still got fairly warm. This led to the addition of external heat-sinks attached by thermal glue to the body of the IC.

4.2. Digital Signal Processing (DSP) Board

The DSP board responsible for applying all DSP operations in real-time was selected based on the following criteria:

- Must accept at least one audio input
- Must have 8 onboard analog outputs (DAC outputs)
- Must be able to apply different fractional delays to several signals simultaneously
- Must be able to interface with an external system to write parameters to RAM

While the list of specifications is short, the second requirement made our search more difficult than originally anticipated. Many of the DSPs we found in a reasonable price range had fewer analog outputs than we required. Many DSP boards, including this one, have many input and output channels. However, the channels are in the form of Inter-IC Sound (I2S) interfaces, which require a hardware codec to interpret properly, as seen in [8]. However, we were able to find an evaluation board for the Analog Devices ADAU1467Z [8], which met our specifications. The



Figure 9: The assembled amplifier board. The sections above are repeated four times, with common power and ground.

board is programmed in Analog Devices' proprietary software, SigmaStudio. However, this does not stop us from writing parameters from other software, as the interface to the board is a well-documented Serial Peripheral Interface (SPI).



Figure 10: The EVAL-ADAU1467Z evaluation board. The four analog input channels are on the bottom, while the 8 analog output channels are on the sides. [8, Fig. 1]

4.3. Speaker Drivers

Our speaker drivers were selected for three particular characteristics:

- Speakers must be as small as possible
- Speakers must audibly reproduce human speech harmonics
- Speaker impedance must reasonably match the output impedance of the speaker amplifier

The first requirement allows us to experiment with speaker spacing freely, without being severely limited by the size of the speaker. While we eventually found that it was best to have some spacing between each speaker, our initial designs placed each speaker directly next to each other to reduce array size. Because our target frequency range contains human speech, we wanted to be able to reproduce those frequencies fairly well. While the fundamental frequency of human speech is usually less than 200Hz, we found that speech can be reproduced fairly well without the fundamental included [10]. This expanded our available speaker options, as smaller speakers tend to be poor at reproducing low frequencies.

Based on these criteria, we selected the CSS-40408N speaker from CUI devices [11]. These speakers are also poor at reproducing low frequencies, as seen in Figure 12. This speaker meets all specifications, however the impedance of the speaker (8Ω) will



Figure 11: The CUI Devices CSS-40408N speaker

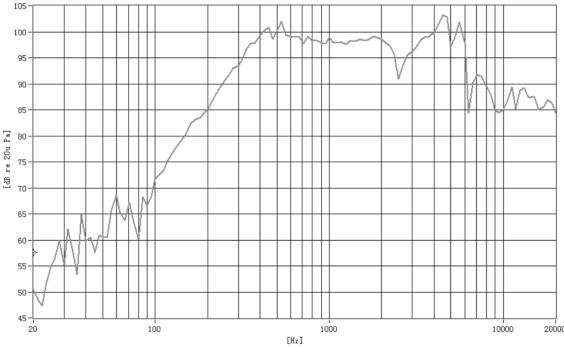


Figure 12: CSS-40408N frequency response [11]

not allow the maximum amount of power to be delivered from the amplifier, which has a 4Ω output impedance [11][9]. This is an acceptable mismatch, as it still allows about 2W of power to be delivered per speaker, as seen in Figure 13.

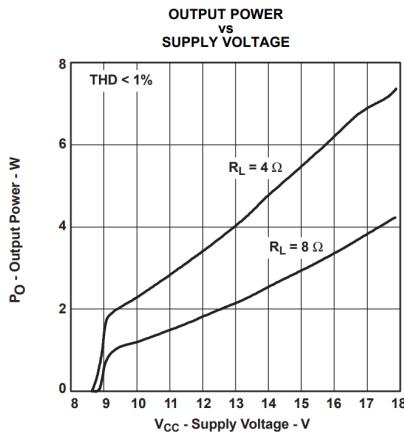


Figure 13: TPA1517NE output power vs. voltage for both 4Ω and 8Ω loads. Our system operates at about 12.5V [9, Fig. 19]

4.4. Overall System Design

Combining the elements described in the previous subsections, the complete system is assembled including other structural materials like laser-cut particle board and 3D-printed parts. Figure 14 shows the front view of the speaker array, while Figure 15 shows the top view of the DSP board.



Figure 14: Front view of the linear speaker array

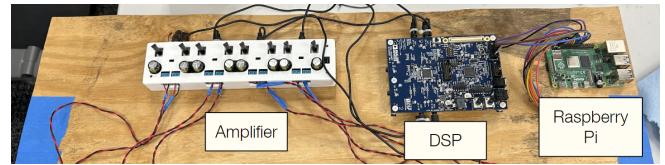


Figure 15: Top view of the amplifier, DSP board, and Raspberry Pi system

5. Testing Methodology

In order to validate our designs, we performed several tests on different parts of the system. In this section, we describe the methodology of each test performed as well as the parameters that are being investigated.

5.1. Amplifier Hardware Testing

Before plugging in the amplifier board, the ripple and the turn-on transient of the power supply was measured. Primarily, we wanted to ensure that the ripple was at a generally tolerable level ($100mV_{pp}$) and there were no large spikes in voltage upon being plugged in, as the amplifier PCB has no transient voltage suppression (TVS) diodes. The ripple was measured with an AC coupled oscilloscope probe directly at the power supply outputs. The turn-on transient was measured with a similar physical setup, however using DC coupling.

An important parameter of the amplifier PCB is its frequency response. While the frequencies being played through each speaker should be the same, it is important to know the phase offset that is generated by the amplifier. For this test, the Analog Devices ADALM2000 was used in the network analyzer mode, analyzing frequencies from DC to 1MHz. One oscilloscope channel was connected to the input of the function generator, and another was connected to the output of the amplifier. The ADALM2000 automatically swept through 100 frequencies per decade, gathering magnitude and phase data at each point. The results of these tests are discussed in Section 6.1, and the setup for the test is seen in Figure 17.

5.2. DSP Board Testing

For this test, we wished to verify the functionality of the DSP board through our amplifier PCB. The set-up for this experiment can be seen in Figure 16. We first began by feeding in a simple 400Hz tone into each of the 4 inputs of the amplifier board. We were able to hear the 400Hz tone being fed out of each of the 8 speakers. Next we wished to verify the compatibility of the amplifier board with our DSP board.

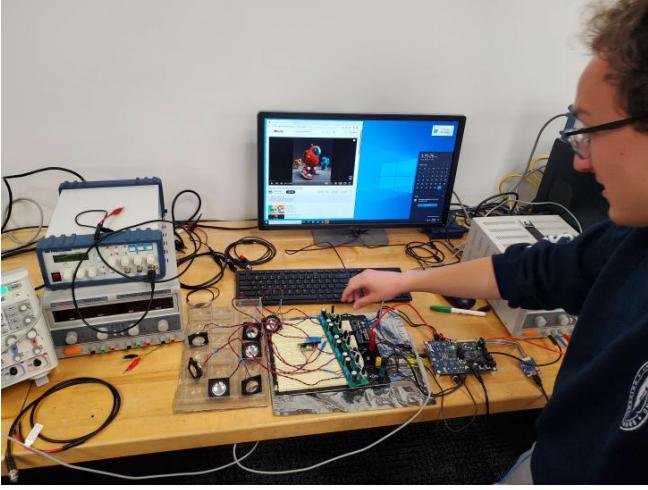


Figure 16: First Functionality Test of Amplifier/DSP Board Interface

We first played an input signal from a PC and fed it as an input one of the stereo input channels of the ADAU1467Z. In SigmaStudio, we made a simple program that simply ties the two-channel input directly to four of the stereo output channels of the board. Each of the four stereo outputs were then fed into the four stereo inputs of the amplifier board. Instead of restricting ourselves to a single tone, we played music from the PC, and were able to verify that each of the 8 speakers were playing the same music at the same volume. A rapid stress test was successfully conducted to acquire a rough bound on the maximal thermal load from the ICs that the heatsinks were capable of mitigating under a high volume setting.

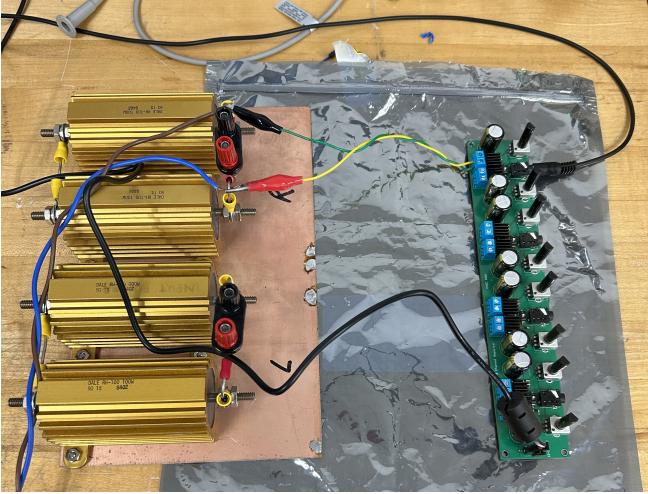


Figure 17: The test setup for the amplifier frequency response measurements. The speaker load has been replaced with a 10Ω power resistor.

5.3. Beamforming Modeling/Simulation

The mathematical model for acoustic beamforming is derived from that for N-slit diffraction, where each slit is representative of an isotropic emitter:

$$E_n = a_n \sin(\omega t + n\Psi) \quad (9)$$

In this equation, a is the amplitude of the signal (normalized to $1/N$ for this purpose), ω is the angular speed, and Ψ is the phase

difference. The radiation pattern is then calculated by summing the individual emissions, resulting in the overall magnitude:

$$E = \left| \frac{\sin(\frac{N\pi d \sin \theta}{\lambda})}{N \sin(\frac{\pi d \sin \theta}{\lambda})} \right| \quad (10)$$

To find the overall radiation pattern including a phase, we first find the phase offset with the equation:

$$\psi(n) = \frac{-2\pi n d \sin \phi}{\lambda} \quad (11)$$

where n is the speaker number, d is the spacing between each speaker, ϕ is the desired beamsteering rubric, and λ is the wavelength. Then, for each viewing angle in the range $\theta = [-90^\circ, 90^\circ]$, we calculate an additional phase shift to account for viewing angle:

$$\Psi = \frac{2\pi d \sin \theta}{\lambda} \quad (12)$$

Then, we sum the magnitude at each speaker and normalize:

$$E(\theta) = \frac{1}{N} \sum_{n=0}^{N-1} \cos(n\Psi + \psi(n)) + j \sin(n\Psi + \psi(n)) \quad (13)$$

[12] [13]

5.4. Beamforming Test Environment

After performing the simulations so see what the expected behavior of the phased array should be, we moved onto the full system test of our design:

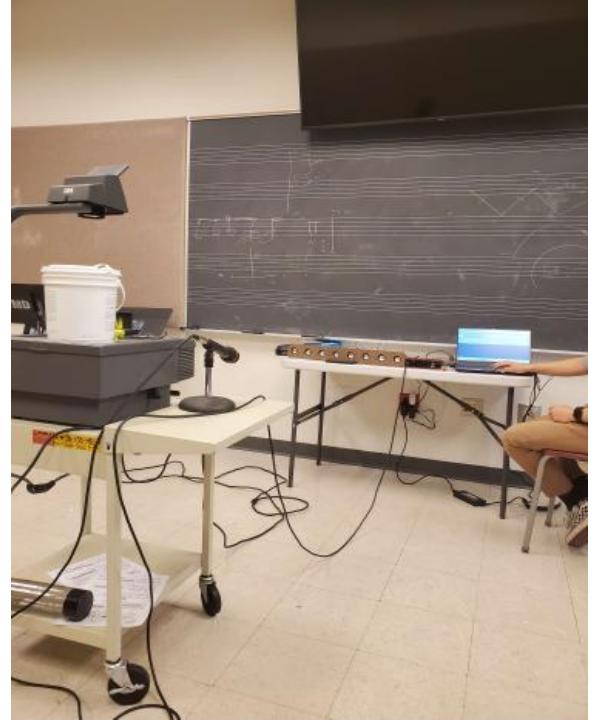


Figure 18: System Testing/Data Collection in Clarice

We elected to perform the sound intensity measurements in the practice rooms of The Clarice Smith Performing Arts Center rather than the ENEE408J laboratory. The motivation for this choice was due to the large amount of reflection off of the hard surfaces in the lab, resulting in less accurate data. The practice

rooms within the Clarice building come equipped with sound dampening material on the walls, which allowed us to eliminate much of the reflection. Although difficult to see in the figure above, the ground was marked with chalk to show the different angle offsets from the edge of the phased array. A microphone was mounted on top of a moving table, vertically level with the phased array. Throughout this test, the microphone/table were moved around to fixed angles that were marked with the chalk. While sweeping through the desired angles, the microphone was always kept at distance of 2 meters from the center of the phased array.

The marked up angles of interest for us were 45° , 90° , and 135° , since those resulted in the most prominent difference in magnitude between the main and side lobes according to our simulations. At each of these angles, we measured the amplitude of the received signal at the microphone when the phased array was set to create a beam at each of the three aforementioned angles. This led to a total of nine measurements. These measurements are shown, processed and discussed in Section 6.3, Table 1.

One difficulty encountered during our data collection was that the microphone we had on-hand for the test was omni-directional, which meant that most of our measured results were almost identical, with only subtle differences in amplitude. To rectify this problem we stopped using the microphone shown in Figure 18 and instead used the directive microphone on an iPhone. This microphone was able to more effectively capture sound with a narrower beam width.

6. Results and Discussion

6.1. Amplifier Testing Results

The power supply ripple/swinging noise and the turn-on transient were measured, as seen in Figures 19 and 20. The power supply noise is unfortunately greater than the specified $100mV_{pp}$, which causes an audible hiss when attached to the amplifier. However, the hiss was not audible enough to warrant a change in power supply. The turn-on transient of the power supply did not contain any measurable voltage spikes, which is desirable as the amplifier board does not allow for voltage spikes above 18V.

The amplifier's frequency response was measured as described in Section 5. The results can be seen in Figure 21. The frequency response is separated into three regimes. From DC to 600Hz, the gain is reduced and the phase is nonzero and decreasing. This is to be expected, as the capacitor's reactance is inversely proportional to the angular frequency of the voltage signal across it. Consequently as frequency increases from 600Hz to 10kHz, the impedance of the capacitor is negligible compared to the input resistance of the amplifier, leading to a near-zero phase offset and constant 20dB gain. From 10kHz onwards, we begin to see the parasitic capacitances of the amplifier circuit begin to play a role. The fixed gain-bandwidth product of the amplifier, where capacitive coupling of the transistor terminals causes gain to drop. We see that gain goes below 1 well before the phase of the amplifier reaches -180° , so the amplifier is stable as expected. Total Harmonic Distortion (THD) was not measured directly, as given that

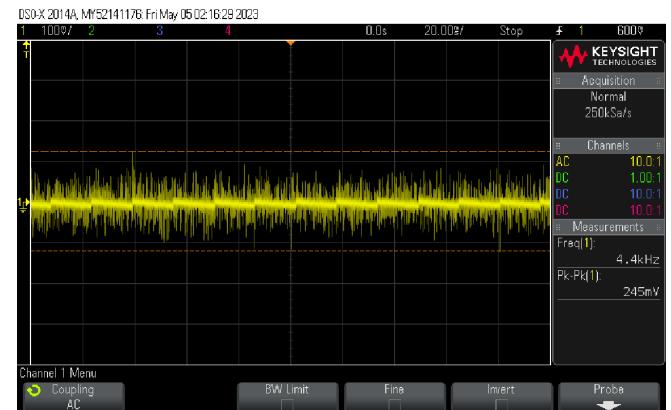


Figure 19: Power supply ripple and switching noise. The maximum noise voltage is $245mV_{pp}$.

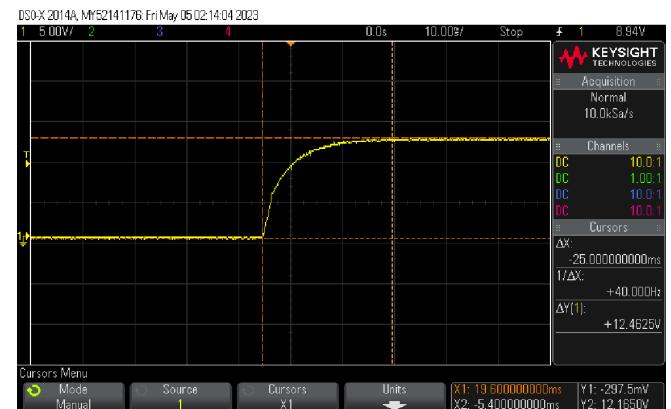


Figure 20: Power supply turn-on transient. There is no noticeable voltage spike upon power-on.

our amplifier was performing at rated specifications, it is safe to assume that the amplifier is operating at $< 1\%$ THD as per the datasheet, and the audio quality was more than sufficient for our purposes.

6.2. Qualitative Evaluation of Model Predictions

The simulation plots depicted below in Figure 22 reveal fascinating insights into the behavior of our phased array system. As we increase the frequency, we observe that the main lobe becomes concentrated, spanning a significantly shorter viewing angle range. This signifies that higher frequencies possess the capability to achieve greater directivity, rendering them more advantageous for our specific application.

Furthermore, we observe that greater frequencies influence the separation of between main and side lobe magnitudes. At 200Hz, we observe a maximum difference of 7dB, whereas at 3200Hz, the difference increases to 19dB.

An additional noteworthy observation is made in the two apparent main lobes at 3200Hz. As the wavelength decreases beyond this frequency threshold, the constructive interference engenders the formation of extra lobes. Specifically, at 3200Hz, a secondary lobe emerges at 135° when the viewing angle is set at 45 degrees. Similarly, when the frequency reaches 6400Hz, two additional lobes become apparent at 90° and 135° . Subsequently, as we further multiply the frequency by 2, more main lobes arise at evenly spaced intervals between the angles of 45° and 135° , forming a pattern of lobes.

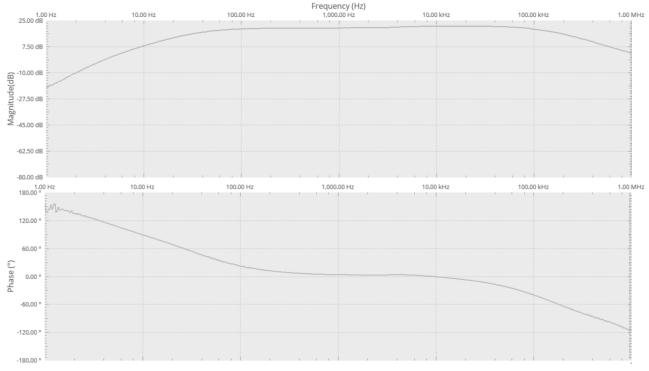


Figure 21: Amplifier board magnitude and phase response. The magnitude response peaks at approximately 20dB.

These simulation results showcase the behavior of our phased array speaker system, highlighting the frequency-dependent changes in its radiation pattern and the intriguing emergence of additional lobes at specific frequency thresholds. This qualitative evaluation provides valuable insights into the performance and potential applications of our phased array system.

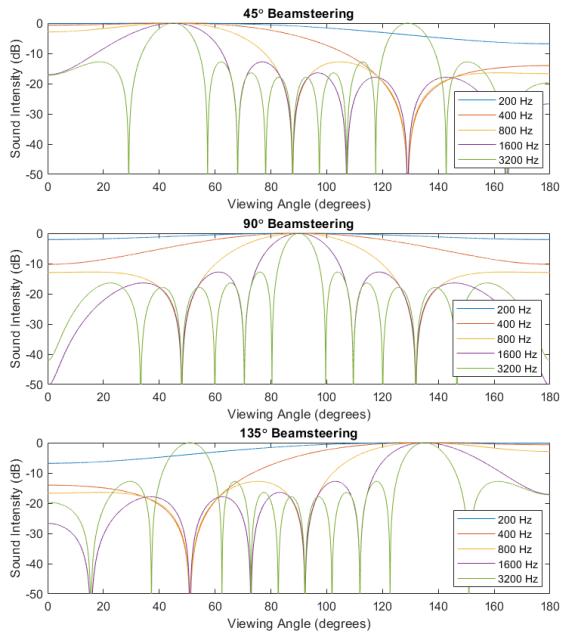


Figure 22: Magnitudes of simulated radiation pattern of 8 speaker linear array across various frequencies with beamsteering angles of 45° , 90° , and 135° .

6.3. Quantitative Evaluation: Checking Against Simulation

The simulations compare favorably to the physical array, indicating that the beamforming was highly effective. Table 1 below shows the quantitative comparison between the measured and simulated results, for a 700Hz tone with beamsteering at 45° , 90° , and 135° . The simulation radiation pattern is shown in Figure 23. Measurements were recorded with respect to the peak magnitude (0dB) at 45° offset from the desired beamsteering angle.

Table 1: Comparison of Measured and Simulated Results

Beamsteering Angle	Measurement (dB)	Simulation (dB)
45°	-15	-22
90°	-11	-21
135°	-15	-22

Both the physical measurement and simulation are highly symmetric, as expected, with the results using a beamsteering angle of 45° mirroring those using a beamsteering angle of 135° . The simulations predicted that there would be a greater difference in magnitude between the main and side lobes than observed in the measured results, with simulated differences of 21 and 22dB as compared to 11 and 15dB in the measured results. There are a number of factors that contribute to these differences. First, the simulation does not account for the full physical nature of the setup. For instance, the model equation uses the far-field assumption to simplify the mathematics involved. Under the far-field assumption, it is presumed that the viewing distance is significantly greater than the length of the array. The viewing distance tested, one meter, while longer than the array length of 0.5614 meters, may not be considered significantly long enough that this assumption holds in its entirety. In addition, we were unable to obtain a highly directional microphone or decibel meter to fully characterize the physical speaker array. Accounting for these small sources of error, beamforming with the linear speaker array was found to be highly effective.

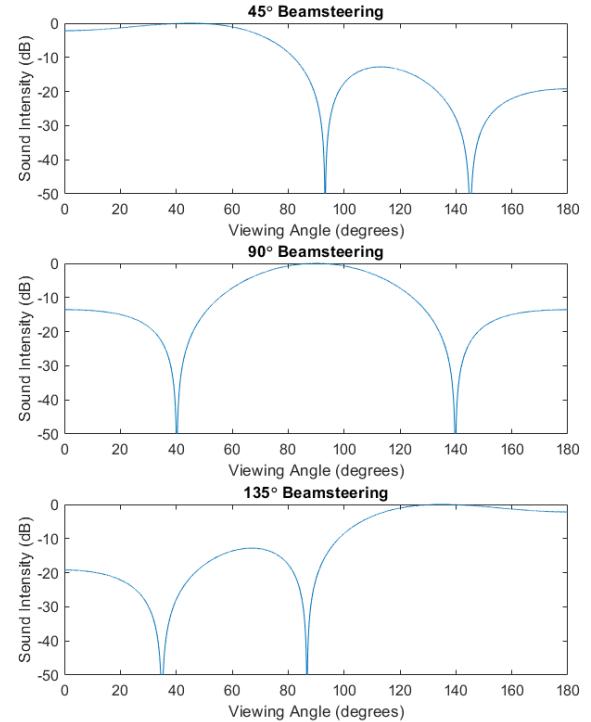


Figure 23: Magnitudes of simulated radiation pattern of 8 speaker linear array at 700Hz with beamsteering angles of 45° , 90° , and 135° .

6.4. Application: LIDAR-Enabled Beamforming

A novel, interesting application of our beamforming technology is to selectively direct sound towards one singular listener, no matter where they may be in a room. The active directing of sound allows for lower perceived volumes for others in the space, potentially reducing disruptions due to loud sounds. To accomplish this, we used an inexpensive 2D LIDAR sensor, the RPLIDAR A1M8, seen in Figure 24. This sensor takes a 2D scan of its



Figure 24: The RPLIDAR A1M8 sensor[14].

surroundings, returning approximately 2000-8000 points of range data per revolution. From this data, we can determine where the closest object is and enter those parameters into our parameter updating Python script. The result of this is that the listener perceives a more or less constant volume when moving across the phased array, as the beam parameters are being updated to follow them. There is no extra data to show for this application, as the beam parameters will be the same as if they were entered manually. The only difference is their source: an automatic sensing process rather than a human operator.

7. Conclusion

Based on the result of our experiments from the previous section, we can say we confidence that our design was a success and worked as intended. However, there are a number of improvements that could be made to our current design.

We can see from the simulations as well as the observations from our live demonstration that the side lobes of the phased array were prominent. In order to combat this, we would like to incorporate a custom FIR filter which handles the implementation of the fractional delays. As of right now, the details of how the Fractional Delay module within SigmaStudio works is unknown. It's possible that the DSP involved in that module block is not the most efficient design for our applications. With our custom FIR filter, there is a strong possibility that we could more effectively fine-tune the fractional delays in order to achieve a narrower beam with attenuated side lobes.

Another potential area of improvement on the hardware side is the amplifier board. It's clear that amplifier board's gain is more

than sufficient for our intended purposes of this design. However, at higher volume settings, the board tends to heat up. This has not been a problem in our final design, since the board doesn't heat up anywhere near dangerous levels. However, the increase in temperature is noticeable if one touches the device directly. In order to increase the longevity of our design, it may be worthwhile to explore how we can optimize the thermal characteristics of the amplifier board, potentially by switching to a high-quality class D amplifier.

We have also noted in the testing procedures how higher frequency input signals resulted in narrower beams. Future work could include a control setting where we can modify the beamforming algorithm in order to cater towards certain bandwidths of the input signal that we feed in. This way, our design can be extended to work well with low-frequency audio signals.

One final concept to explore is how we can implement a beamforming system using a frequency-domain algorithm. Our algorithm works completely in the time-domain, so it would be interesting to see how we can form a beam using a strict frequency-domain analysis of our input signal.

Two big projects that our design can lead into for future work are the following:

1. Dynamic Bandpass Filter

(a) This project involves designing a phased array system that implements a dynamic band-pass filter to amplify an attenuate certain frequencies based on the observer's position. For example, if the observer is located at about 0° then the output signal of the array will be a bass-boosted version of the input audio signal. As the observer increases their theta position, the DSP algorithm will shift the band-pass filter accordingly to higher frequency ranges. As the observer approaches 180° , the output signal will be more and more treble-boosted.

2. Multi-Input Beamforming

(a) This project involves feeding in two (or potentially more) audio signals as an input to the linear phased array system. The user of the array system will choose to destination values for the sound by specifying two distinct pairs of θ and D . The phased array system will beamform signal 1 to one destination and will beamform signal 2 to the second destination simultaneously. This can be achieved by running both signals into independent delay modules, summing the delayed versions up and into a single superimposed signal, and then outputting the summed signal from all the speakers. This is practically the inverse of what is presented in [15].

Acknowledgements

The authors acknowledge the support of Brian Louis Beau-doin, Justin Welter, the ECE Tech-Ops staff, and the unknown people who physically assembled the PCBs.

References

- [1] The cobra judy phased-array radar system on the missile range instrumentation ship usns observation island (t-agm 23), <https://nara.getarchive.net/media/the-cobra-judy-phased-array-radar-system-on-the-missile-range-instrumentation-d23c58>, accessed: 2023-05-05.
- [2] A. A. Nair, T. D. Tran, A. Reiter, M. A. Lediju Bell, A deep learning based alternative to beamforming ultrasound images (2018) 3359–3363doi:10.1109/ICASSP.2018.8461575.
- [3] Pearl harbor's giant 'golf ball' might be moved, <https://www.staradvertiser.com/2015/04/27/hawaii-news/pearl-harbors-giant-golf-ball-might-be-moved/>, accessed: 2023-05-15.
- [4] K. Benson, Phased array beamforming ics simplify antenna design (2019).
- [5] E. Szoka, T. Jackson, Phased array speaker system (2012).
- [6] P. Dhaker, Introduction to spi interface, <https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html> (9 2018).
- [7] Glidepoint tm5957 trackpad, <https://www.cirque.com/glidepoint-tm5957-trackpad>, accessed: 2023-05-19.
- [8] Analog Devices, Evaluating the ADAU1463 and ADAU1467 SigmaDSP Audio Processors, rev. 0 (10 2017).
- [9] Texas Instruments, 6-W Stereo Audio Power Amplifier (2 2007).
- [10] G. Cristina Oliveira R, Acoustic, electroglottographic, and accelerometer measurement in individuals with and without vocal alteration (9 2019). doi:10.1016/j.jvoice.2019.08.004.
- [11] CUI Devices, Speaker (01 2020).
- [12] C. Grassin, Acoustic beamsteering with a speaker array, <https://charlesslabs.fr/en/project-Acoustic+beamsteering+with+a+speaker+array>, accessed: 2023-05-18 (3 2020).
- [13] S. V. Hum, Antenna arrays, Lecture, ECE422, Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada. <https://www.waves.utoronto.ca/prof/svhum/ece422/notes/15-arrays2.pdf> (2018).
- [14] Rplidar a1: Entry-level lidar, <https://www.slamtec.com/en/Lidar/A1>, accessed: 2023-05-02.
- [15] A. Hidri, M. Souad, A. Alaqueeli, H. Amiri, A deep learning based alternative to beamforming ultrasound images (2012). doi:10.4156/jdcta.v016.issue20.72.

Appendix A. Digital Models and Software Implementation

All of the code for DSP interfacing and beamforming simulation is in our project's Github repository, found here. Along with the code, CAD models for all laser-cut and 3D-printed parts are present in the Models folder of the repository. The Altium Designer files for the custom 8-channel amplifier PCB can be found in the Amp_8ch folder.