Active Electronically Scanned Speaker Array

Austin P. Edwards

Abstract

Electronically scanned arrays or phased arrays are becoming increasingly more useful in many different engineering applications. Using speakers as the transmitters in a phased array, the properties of the array can be audibly represented and thus, can be more easily analyzed. Given several design and construction constraints, an inexpensive system was built using piezo-electric buzzers, voltage amplifiers, voltage dividers, and an Arduino Mega microcontroller. The designed array was created with 25 array elements, producing an audio signal of 3136Hz or the musical tone G7, with an element spacing of 0.05m or approximately $\lambda/2$. The performance of the phased speaker array was then analyzed and compared with the theoretical values at three different phasing angles: $\theta_o = 0,\ \pi/4, \text{and -}\ \pi/3$. The results of the comparison between the measured values and those of the MATLAB simulations illustrate and prove the phased array's ability to accurately steer the audio signal.

Introduction

Phased arrays are extremely important in telecommunications, radar, and audio applications. What makes them unique from other forms of antennas is their ability to change the shape and direction of the radiation pattern without physically moving the antenna. To achieve this, each element of the array can transmit a signal of equal frequency with small differences in phase. This shift in phase ensures that there is constructive interference in the desired direction and deconstructive interference at all other angles. This produces an efficient, high gain, narrow main beam with little to no signal or side lobes in other directions. These antennas, being extremely useful having the capability of commendable, agile beams, are used in many applications including 5G antennas, radar on air force fighter jets, and directional audio.

In an active electronically scanned array (AESA), each antenna element is connected to a small transmitter and receiver module under the control of a computer or microcontroller. As opposed to a passive electronically scanned array (PESA) which has all the antenna elements connected to a single transmitter and receiver module through phase shifters, an AESA is a more advanced form of phased array and has numerous advantages.

This project will focus on applying the principles of an AESA to a slightly different and less commonly used application: audio. The goal of this project is to create an array of speakers and use a microcontroller to control the direction of the resultant beam of audio signals by phase-shifting the signals at each array element.

ASEA Fundamental Topics

Array Factor

When designing the active electronically scanned array, many fundamentals including grating lobes, beamwidth, pattern optimization, and aperture distribution must be understood for a successful design.

The array factor (AF) describes the spatial response of the phased array. It is a complex-valued, far-field radiation pattern obtained for an array of N elements. In a linear array without phase delay, the equation for the AF is fairly easy to analyze, consisting of only the aperture distribution, frequency, element spacing, and angle. As element weighting will be addressed later, the aperture distribution (A_n) can be disregarded from the following equation.

$$\mathrm{AF} = \sum_{n=1}^{N} A_n e^{j\frac{2\pi}{\lambda} x_n \sin\theta}$$

Eqn. 1: *Non-Phased Linear Array Factor*

The AF has an obvious maximum value at $\theta = 0°$ or directly in front of the array. This maximum value is equal to the number of array elements present and will act as a good check in later equations to ensure there are no obvious errors. For visualization and analysis, the AF for a

10-element linear array with an element spacing of $\lambda/2$ was plotted using MATLAB (Function code in the appendix).

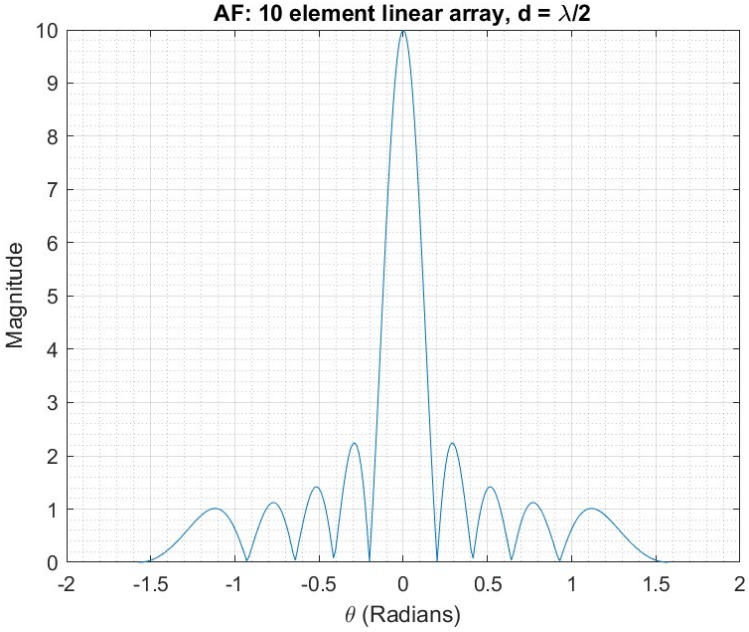


Figure 1: *AF of 10 element linear array,* $D = \lambda/2$

As shown, the maximum value of the AF does indeed equal the number of elements in the array. The element spacing of $\lambda/2$ was carefully selected in order to prevent grating lobes from occurring. Grating lobes are the same magnitude of the main beam, however, they only occur when the array spacing is greater than $\lambda/2$. When setting the element distance to a larger value like d = λ, grating lobes will appear on each side of the plot.

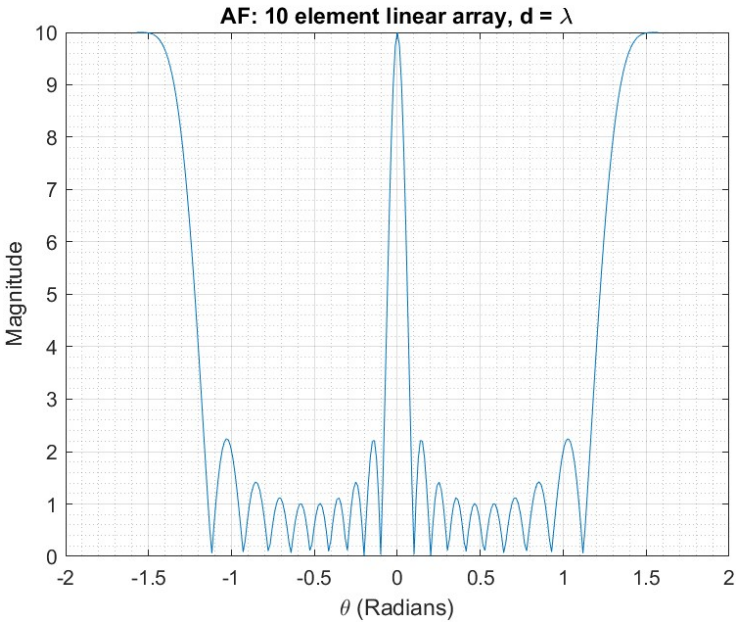


Figure 2: *AF of 10-element linear array,* $D = \lambda$

Although preventing grating lobes is a large priority, decreasing the element spacing too far will begin to increase the width of the main beam.

By adding a phase delay term to the equation for the AF, we can analyze the effect a phase angle has on the spatial response of the array.

$$\mathrm{AF} = \sum_{n=1}^{N} A_n e^{j\left(\frac{2\pi}{\lambda}x_n \sin\theta - \frac{2\pi}{\lambda}x_n \sin\theta_o\right)}$$

Eqn. 2: *Phased Linear Array Factor*

Again, we can see that if both $\theta = 0°$ and $\theta_o = 0°$ (the angle centered directly above the array with no phase delay) the AF reaches a maximum equal to the number of elements in the array. The phase delay angle $\theta_o = 0°$ is determined on the phase delay or short time delay between each array element.

$$\theta_o = \sin^{-1}\left(\frac{c}{2\pi d}\delta t\right)$$

Eqn. 3: *Phase angle as a function of time delay*

This equation will be used later to calculate the time delay needed to control the phase angle using a microcontroller. When adding a phase angle of $\theta_o = \pi/4$, the MATLAB simulation shows the main beam correctly being shifted over by the specified amount.

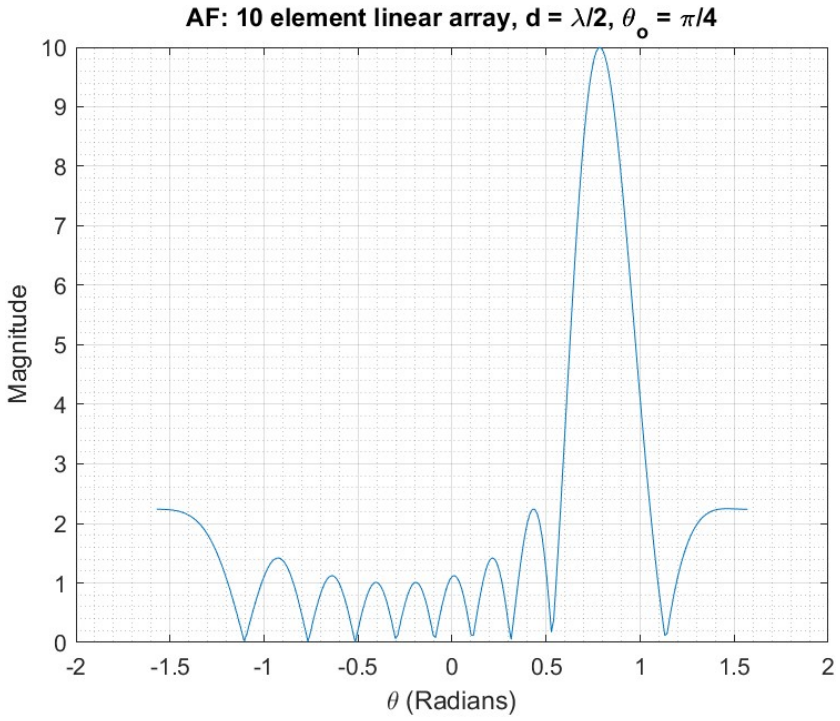


Figure 3: *AF of 10-element linear array,* $D = \lambda/2$, $\theta_o = \pi/4$

Since the distance between array elements is large enough, grating lobes do not make an appearance on the left side of the plot. However, if the phase angle is large enough, a copy of the main beam will reappear and cause unwanted signals. This occurrence of grating lobes at extreme phase angles will not be actively avoided during this project as the element spacing would have to be decreased significantly to leading to an wider less-directional main beam. Since the grating lobes will only occur at an angle equal to or larger than $\theta_o = \pi/3$, no effort will be put forth to prevent them.

The AF can also be expressed to describe the spatial response of a 2-dimensional NxM array.

$$\mathrm{AF} = \sum_{l=1}^{N*M} C_l e^{j\left(\frac{2\pi}{\lambda}x_l \sin\theta\cos\phi - \frac{2\pi}{\lambda}x_l \sin\theta_o \cos\phi_o\right)} e^{j\left(\frac{2\pi}{\lambda}y_l \sin\theta\sin\phi + \frac{2\pi}{\lambda}y_l \sin\theta_o \sin\phi_o\right)}$$

Eqn. 4: *Phase 2-dimentional Array Factor*

After some slight analysis, it is obvious that this equation satisfies our initial condition that the maximum value is equal to the number of elements in the array. When θ and $\theta_o = 0°$, all exponential terms cancel and leave a simple summation of the elements contained in the NxM array multiplied by the aperture distribution. By plotting this equation in MATLAB, our analysis can be confirmed.

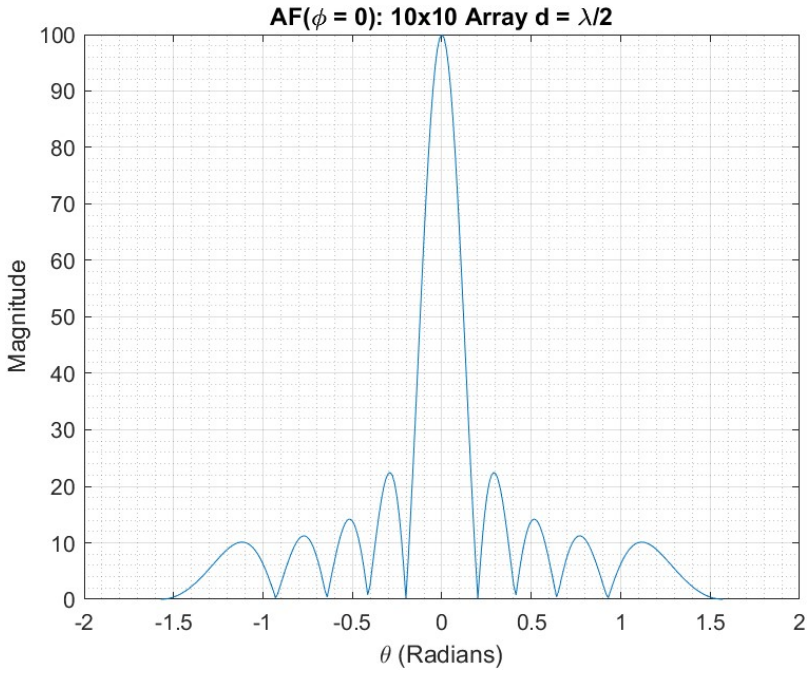


Figure 4: *AF of 10x10 rectangular array, , $D = \lambda/2$, $\theta_o = 0$*

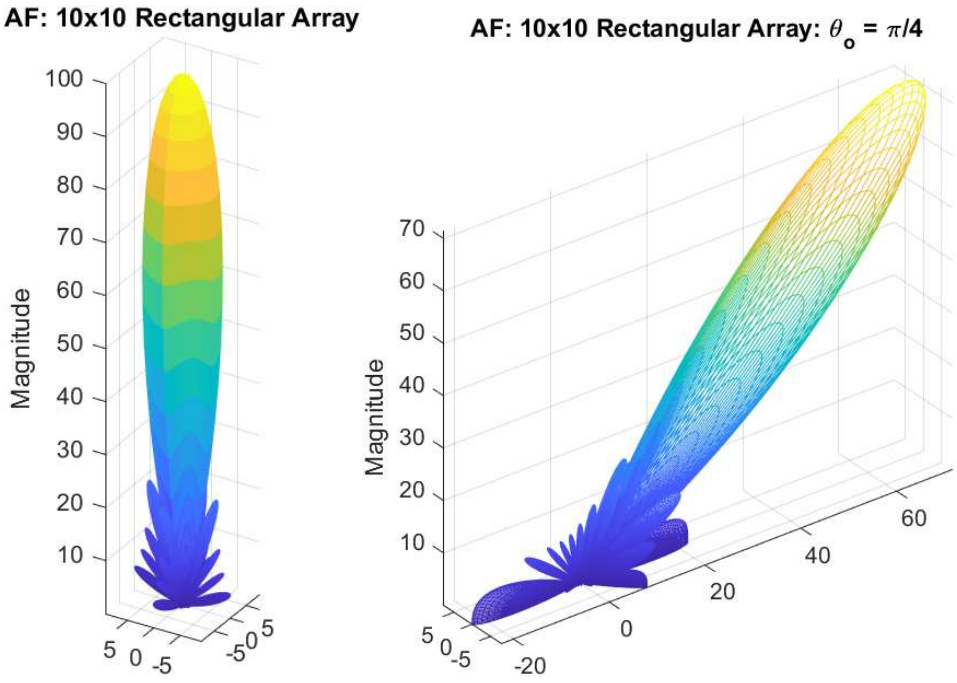


Figure 5: *AF 3D Representation of 10x10 rectangular array,* $D = \lambda/2$, $\theta_o = 0$ *and* $\pi/4$

Element Pattern (EP) and Complete Pattern Representation

The AF does not completely describe the spatial response of the array. Each of the elements in the array has an element pattern (EP) that is the elements' individual spatial response. A good expression for modeling this element pattern is a cosine function raised to a power called the element factor (EF).

$$\mathrm{EP} = \cos^{\frac{\mathrm{EF}}{2}}\theta$$

Eqn. 5: *Element Pattern*

The complete pattern representation to describe the complete spatial response of the array can be found by simply multiplying the AF and EF together.

$$F(\theta) = \mathrm{EP} * \mathrm{AF} = \cos^{\frac{\mathrm{EF}}{2}}\theta * \sum_{l=1}^{N*M} C_l e^{j\left(\frac{2\pi}{\lambda}x_l \sin\theta\cos\phi - \frac{2\pi}{\lambda}x_l \sin\theta_o \cos\phi_o\right)} e^{j\left(\frac{2\pi}{\lambda}y_l \sin\theta\sin\phi + \frac{2\pi}{\lambda}y_l \sin\theta_o \sin\phi_o\right)}$$

Eqn. 6: *Complete Pattern Representation*

The equation for the complete pattern representation can then be converted into dB as the measurement of audio (decibels) is on a logarithmic scale.

$$F_{\mathrm{dB}} = 10\log_{10}(\mathrm{EP} * \mathrm{AF})^2$$

Eqn. 7: *Converting F to dB*

By converting all three equations (AF, EP, and F) into dB and plotting in MATLAB, we can easily compare and determine the effect that element pattern has on the complete pattern representation.

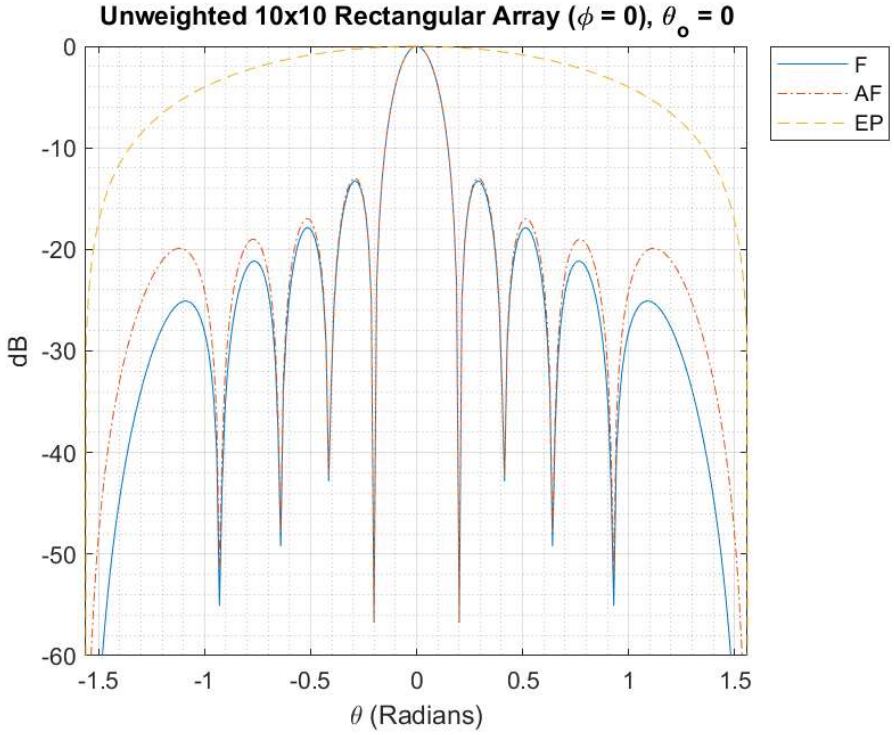


Figure 6: *AF, EP, and F of 10x10 rectangular array,* $D = \lambda/2$, $\theta_o = 0$

The EF roll-off attenuates the AF pattern, decreasing the size of the sidelobes at large values of θ. This difference has the most impact at large phase angles where the EP scan loss must be accounted for in analyzing the array performance. In addition to scan loss, the EP causes a slight shift in the beam angle. This second effect is quite negligible; thus, it can be discarded from the analysis.

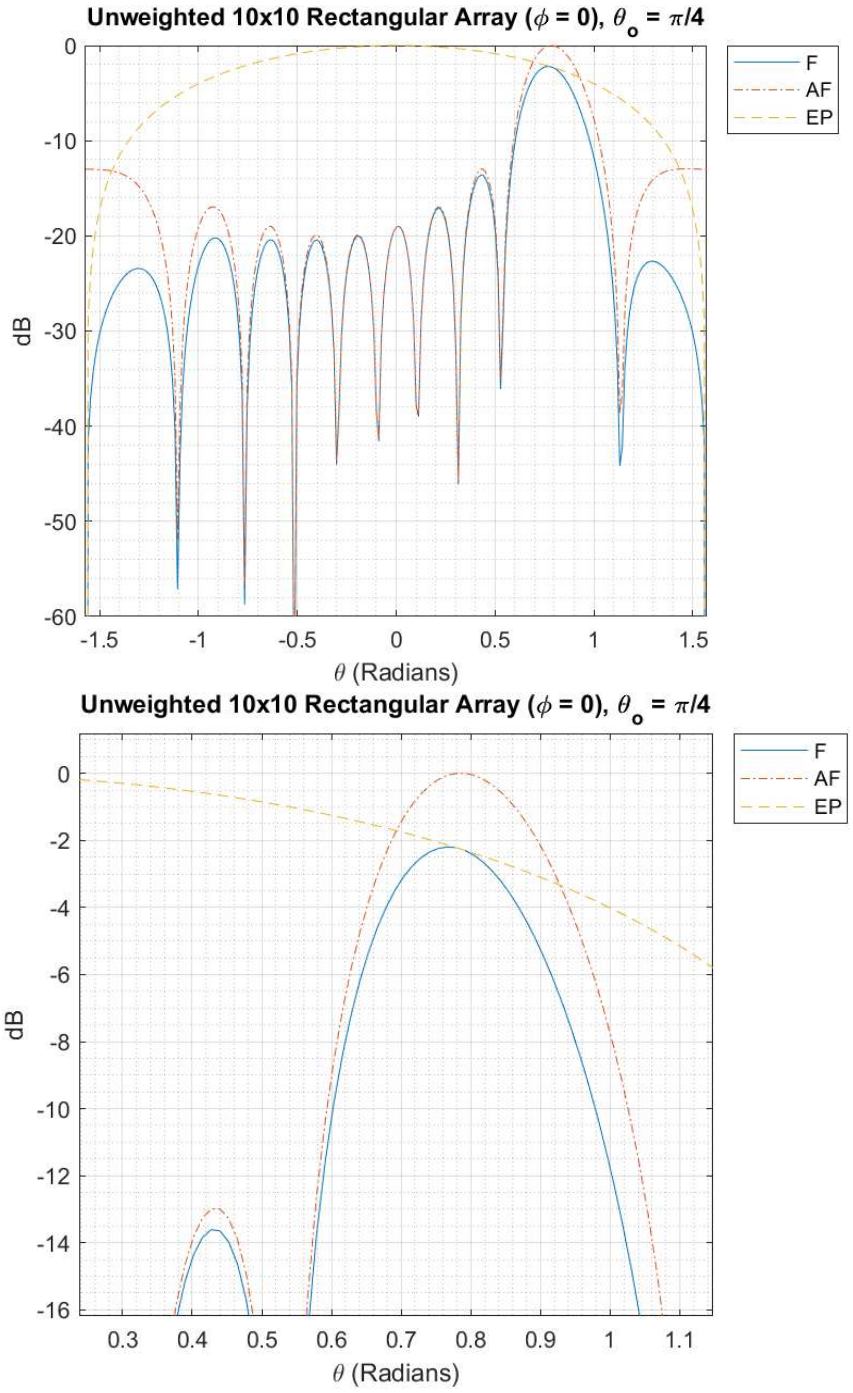


Figure 7 & 8: *AF, EP, and F of 10x10 rectangular array* $D = \lambda/2$, $\theta_o = \pi/4$

The 3-dimensional representation looks almost exactly the same as the AF, but as shown above, the sidelobes decrease in magnitude at greater values of θ. Furthermore, the EP causes a slight decrease in the main beam magnitude as θ_o increases.

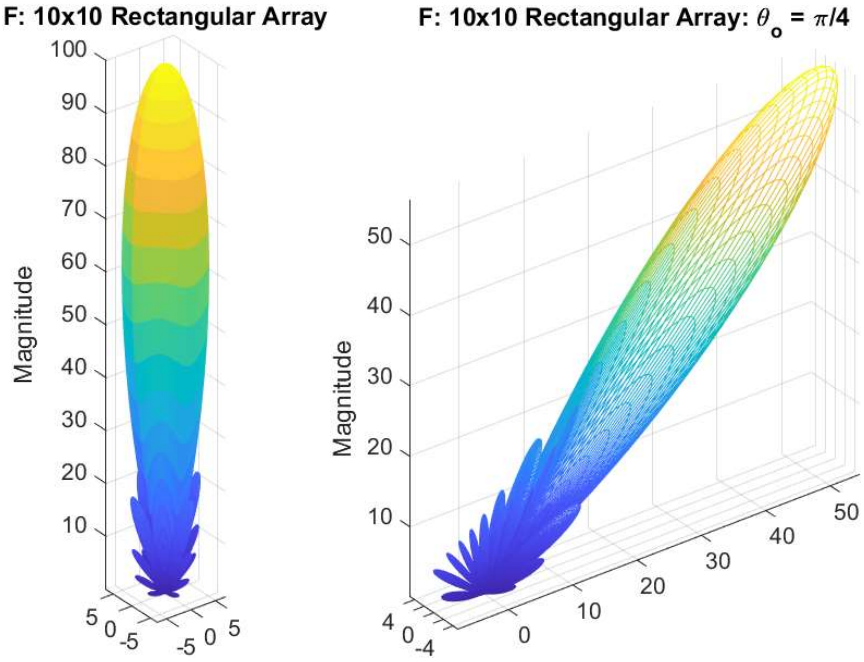


Figure 9: *F 3D Representation of 10x10 rectangular array,* $D = \lambda/2$, $\theta_o = 0$ *and* $\pi/4$

Aperture Distribution

It was shown that by varying the phase of each element in the array, the main beam can be steered spatially. In addition to modifying the individual elements phase, the element amplitudes or 'weights' can be modified to lower sidelobes. For all previous calculations and simulations, the aperture distribution used was a constant value of 1 for every array element and produced sidelobe levels of about 14dB below the peak of the main beam (Figure 6). With an added phase angle of $\theta_o = \pi/4$, the side lobes produced are only 12dB lower than the main beam. This difference in dB levels decreases as the EF roll-off attenuates the F pattern or as θ_o increases.

In order to reduce these sidelobe levels even more, different forms of element weighting can be applied. Although numerous types of weightings can be used, this project will focus on Taylor weighting as it is known as the most efficient aperture distribution. Taylor weighting gives complete control on the level of the first side lobe as well as the number of nearly constant leveled sidelobes adjacent to the main beam. For this project, this second value or the number of constant leveled sidelobes will be set to 1 in order to let the magnitude of the side lobes naturally decay with greater values of θ. The following Taylor weights were calculated using a modified version of the taylorwin() function in MATLAB (Function code in the appendix).

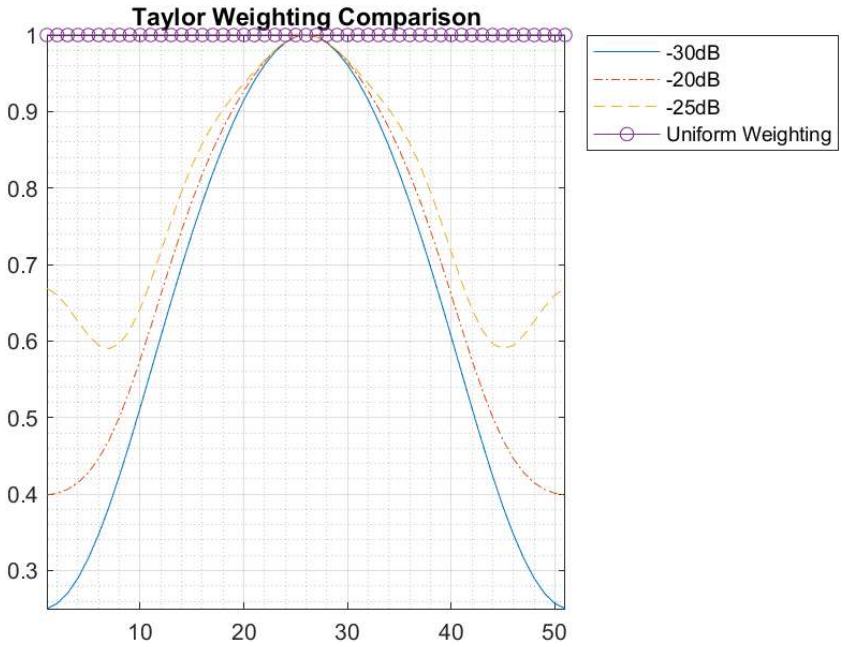


Figure 10: *Taylor weighting comparison.*

Figure 10 displays 3 different levels of Taylor weighting with sidelobes at -30dB, -25dB, and -20dB applied to 50 elements to better show the Taylor weighting pattern. In order to decrease the sidelobe levels the most significantly, the -30dB Taylor weights will be used throughout the rest of this project. This weighting system can be easily applied to a 2-dimensional 10x10 rectangular array by multiplying the weighting values of a 10-element linear array across both axes.

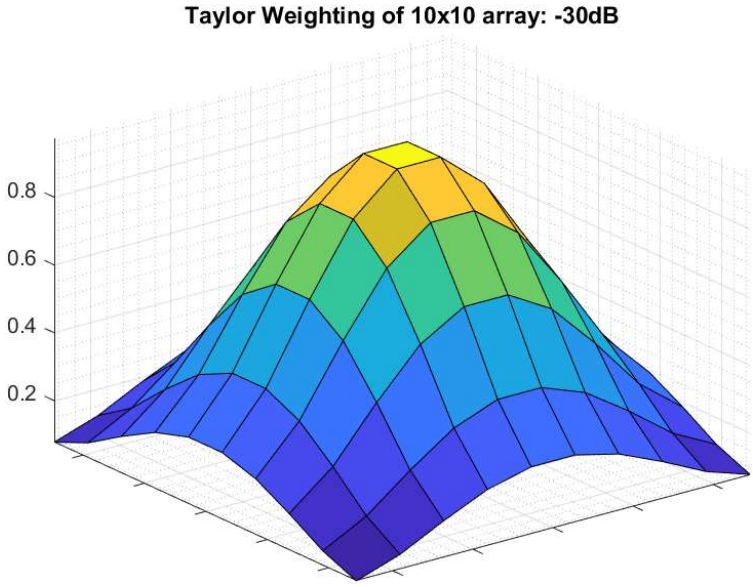


Figure 11: *3-dimensional magnitude representation of Taylor weighting on a 10x10 rectangular array*

Element weighting, however, has a specific drawback. The more element weighting is applied to decrease the sidelobe levels the wider the main beam becomes. Since the -30 Taylor weighting does not widen the main beam significantly, it is the obvious choice between these 3 Taylor weighting system.

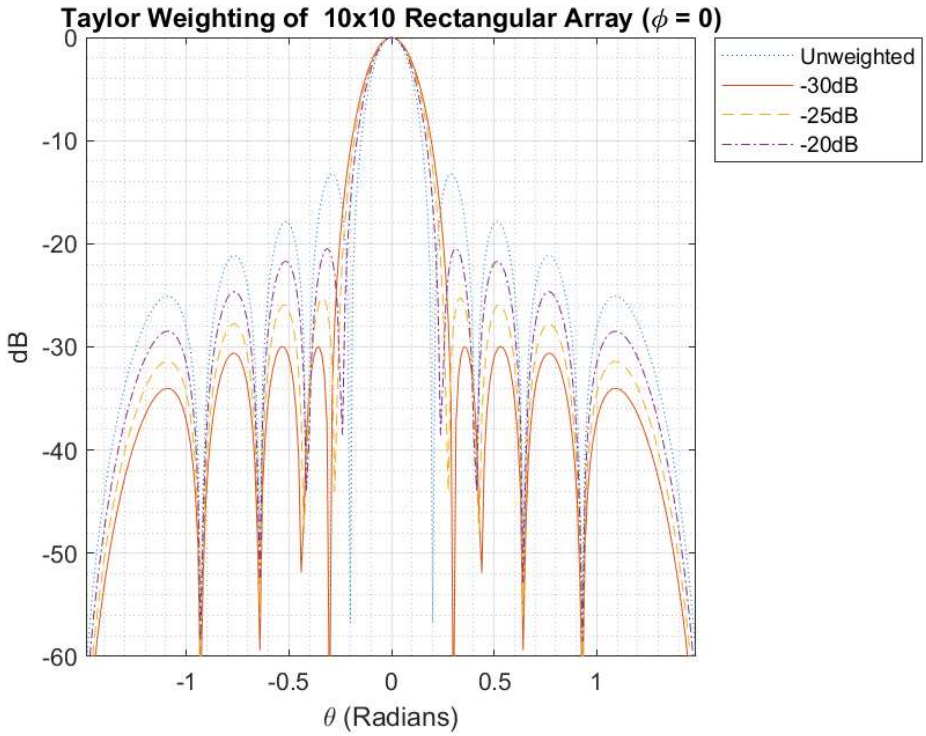


Figure 12: *Effect of Taylor weighting on the main beamwidth.*

Phased Speaker Array Design

There are several limiting factors on the design of the electronically scanned speaker array. Since the array will contain numerous array elements, the speakers chosen for the project would have to be fairly inexpensive. Furthermore, to demonstrate the phasing principle, all the speakers will have to be transmitting the same frequency of sound. Given these constraints, passive piezo-electric buzzers were chosen to act as the array elements.

Passive piezo-electric buzzers generate a sound by using a piezo crystal, a material that changes shape when voltage is applied to it. The crystal is pushed against a diaphragm and creates an audible tone. Instead of a normal audio signal, passive piezo-electric buzzers require a square wave voltage pulse train, transmitting at the frequency of the chosen audio frequency. For simplicity, the base frequency used will stay the same throughout the project: 3136Hz or the musical note G7.

Since a microcontroller will need to output square wave voltage pulse trains, the next limiting factor is the microcontroller's number of internal clocks. The microcontroller used, the Arduino Mega, has a total of six clocks running at 16MHz. One out of the six available clocks will be required for adding small time delays to each output signal to control the speaker's phase angle.

In Arduino IDE, however, the tone library (holds the functions required to create the square wave voltage pulse trains) limits the maximum number of usable clocks to four. This limitation was easily bypassed by editing the tone library C code to include all 6 of the Arduino Mega clocks. The number of square wave pulse trains to power the phased array using the Arduino Mega was now limited to 5. Since a 5-element array is not ideal in any aspect, some modifications to the array must be made.

By limiting the phasing capabilities to only the θ direction (essentially steering the main beam from left to right when the array is pointed parallel to the ground), the elements contained in each column of the array can be connected together and driven by the same square wave pulse train. This expands the number of usable array elements from a 5-element array to a 5xN element array. For simplicity and cost efficiency, the array size chosen for this project is a 5X5 rectangular array (other latices were analyzed however because of their symmetry, they required more clocks/array element than a rectangular array).

Using a frequency of 3136Hz, the distance between each element to eliminate grating lobes must be equal to or less than $\lambda/2$.

$$D = \lambda/2 = c/2f = 0.055m$$

For simplicity, the element distanced used for both axes of the rectangular array was .05m.

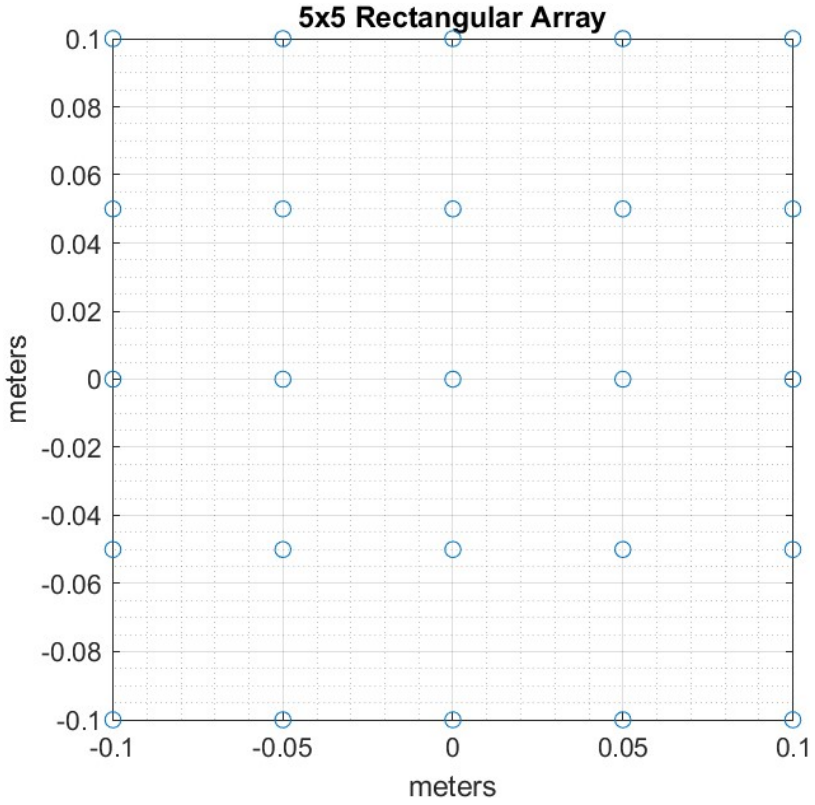


Figure 13: *Speaker array element pattern*

Since the microcontroller cannot power all 25 piezo-electric buzzers, five simple voltage amplifiers need to be designed to power the 5 phasing columns of the array. A 1KΩ resistor is placed between the amplifier and the microcontroller to limit the current being drawn from the Arduino Mega. As phasing will be applied using small time delays and all the array elements will be connected to the same type of amplifier, a simple BJT inverting voltage amplifier can be used.

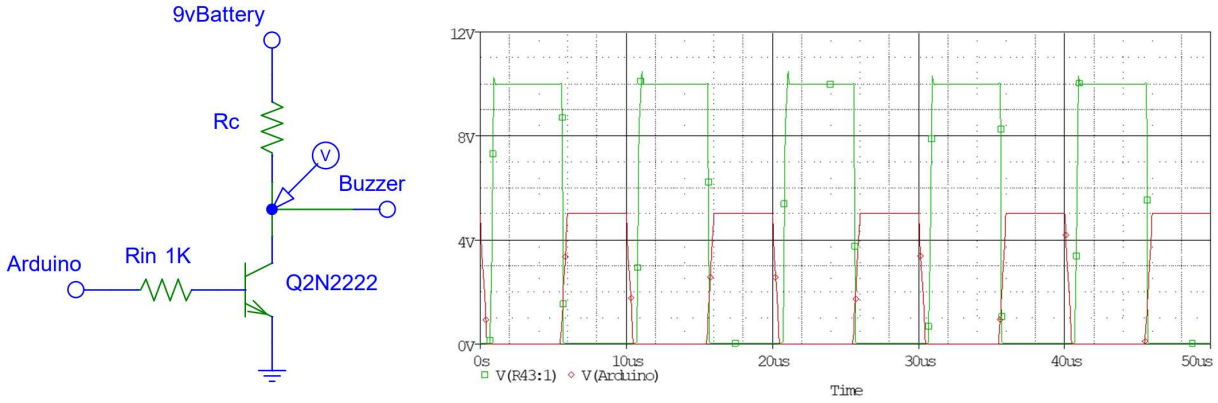


Figure 14: *Inverting BJT Voltage Amplifier*

Along with amplifiers, voltage dividers are required to both limit the volume of the piezo-electric buzzers and apply the Taylor weighting aperture distribution. For accuracy, the voltage across the piezoelectric buzzer without the secondary voltage divider will be used as a reference to calculate the values of the resistors needed to properly weight the array. Using a multimeter, the voltage across the unweighted piezoelectric buzzer is found to be 3.71V with an internal resistance of 15.2Ω.

Taylor Weight to Excitation Voltage (multiplying by 3.71V)										
	Column 1	C1	C2	C2	C3	C3	C4	C4	C5	C5
Row 1	0.116	0.43V	0.264	0.98V	0.340	1.26V	0.264	0.98V	0.116	0.43V
R2	0.264	0.98V	0.603	2.24V	0.777	2.88V	0.603	2.24V	0.264	0.98V
R3	0.340	1.26V	0.777	2.88V	1.000	3.71V	0.777	2.88V	0.340	1.26V
R4	0.264	0.98V	0.603	2.24V	0.777	2.88V	0.603	2.24V	0.264	0.98V
R5	0.116	0.43V	0.264	0.98V	0.340	1.26V	0.264	0.98V	0.116	0.43V

Figure 15: *Taylor weighting to excitation voltage*

Using the Taylor weights or ratio of V_{out}/V_{in} the voltage divider resistor values can easily be calculated.

$$R = \frac{V_{in}R_{Buzzer}}{V_{out}} - R_{Buzzer}$$

	Column 1	C2	C3	C4	C5
Row 1	116.0Ω	42.3Ω	29.5Ω	42.3Ω	116.0Ω
R2	42.3Ω	10.0Ω	4.4Ω	10.0Ω	42.3Ω
R3	29.5Ω	4.4Ω	0Ω	4.4Ω	29.5Ω
R4	42.3Ω	10.0Ω	4.4Ω	10.0Ω	42.3Ω
R5	116.0Ω	42.3Ω	29.5Ω	42.3Ω	116.0Ω

Figure 16: *Ideal voltage divider resistors*

Since these values of resistors are not standard, the following table is the actual resistance value used in this project.

	Column 1	C2	C3	C4	C5
Row 1	115.0Ω (47+68)	42.0 Ω	29.5Ω	42.0 Ω	115.0Ω
R2	42.0Ω (22+10+10)	10.0Ω (10)	4.07Ω	10.0Ω	42.0 Ω
R3	29.5Ω (22+7.5)	4.07Ω (22‖15‖7.5)	0Ω	4.07Ω	29.5Ω
R4	42.0 Ω	10.0Ω	4.07Ω	10.0Ω	42.0 Ω
R5	115.0Ω	42.0 Ω	29.5Ω	42.0 Ω	115.0Ω

Figure 16: *Experimental voltage divider resistors*

Fortunately, the values of the experimental voltage divider resistors were able fairly accurately match the values of the ideal resistors. Because of this, the weighting error is negligible and the ideal vs. experimental weights are not visibly differentiable when plotted in MATLAB. It can thus be assumed that the element weights are working at almost 100% efficiency.

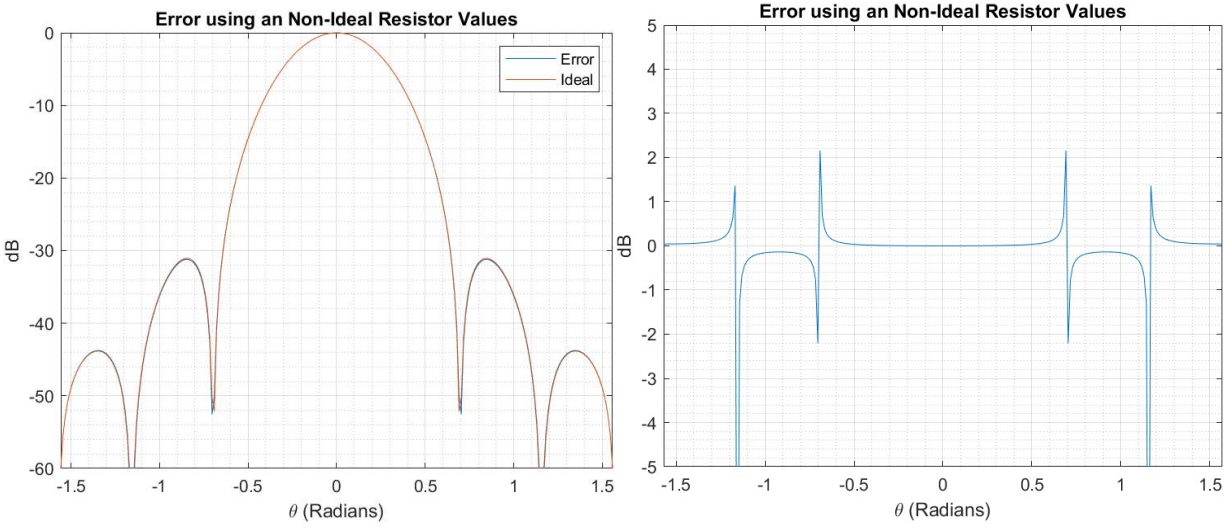


Figure 17: *Experimental vs ideal voltage divider resistors*

Adding a Phase Delay to the Speaker Array

To change the angle of the main beam or the phase angle, small time delays are added in between the Arduino functions that transmit each array columns audio signals. From before, since the only phasing is in the θ_o direction, the phase angle can be written as a function of time.

$$\theta_o = sin^{-1}\left(\frac{c}{2\pi d}\delta t\right)$$

Solving for the time delay between each array element (array column in this case), the equation can be written in terms of the expected phase angle θ_o.

$$\delta t = \sin\theta_o \frac{2\pi d}{c}$$

Adding this time delay to an Arduino IDE program is slightly more complicated. The time between two pins starting the square wave voltage pulse train required to power the piezo-electric buzzers depends on the number of clock cycles the function tone.play() takes to execute. This can be estimated by running the same function through a loop several hundred times and recording the elapsed time. Running the function tone.play() 500 times results in an overall delay of 5.835ms. Thus, the execution time between two Arduino pins starting the square wave voltage pulse train is approximately 11.67µs. The required time delay between array elements in order to steer the main beam of the array can be written as δt minus the execution time.

$$\delta t = \sin\theta_o \frac{2\pi d}{c} - 11.67\mu s$$

In Arduino IDE, this can be applied as a simple delayMicroseconds() function between initiating the square wave voltage pulse trains with a slight caveat. Because of the Arduino Mega clock speed (16MHz), delayMicroseconds() is only accurate to multiples of 4µs and can even vary up to 8µs. Giving δt a random error from between -8µs and +8µs of the ideal time delay, the expected error to occur can be plotted in MATLAB. This error is quite negligible and thus can be discarded from the analysis.

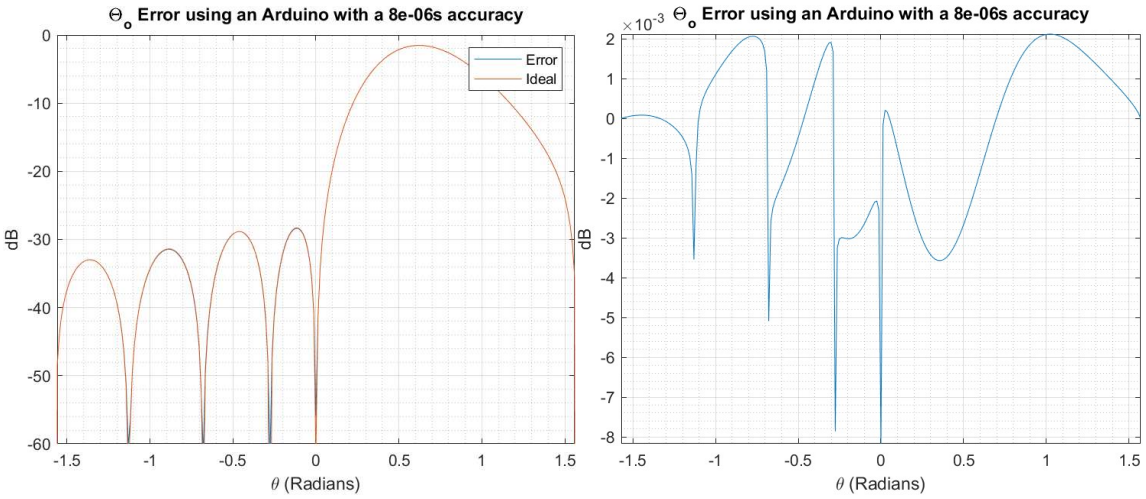


Figure 18: *Experimental vs ideal time delay*

Building the Phased Speaker Array

Given the constraints stated in the design process (number of Arduino Clocks, number of array elements, and the element spacing) the building process of the phased speaker array was fairly straightforward. All 25 piezo-electric buzzers were tested before construction using an Arduino Mega, necessary electrical components, and a breadboard. The buzzers seemed to have an extremely 'tinny' sound that was fixed by placing a piece of electrical tape over the sound emission hole to increase the resonance of the buzzer.

Figure 19: *Passive piezo-electric buzzers with and without an emission hole cover*

To mount the piezo-electric speakers, 25 holes with a spacing of 5cm were drilled into a 30cmx30cm piece of plywood. All the piezo-electric buzzer grounds were then soldered together.

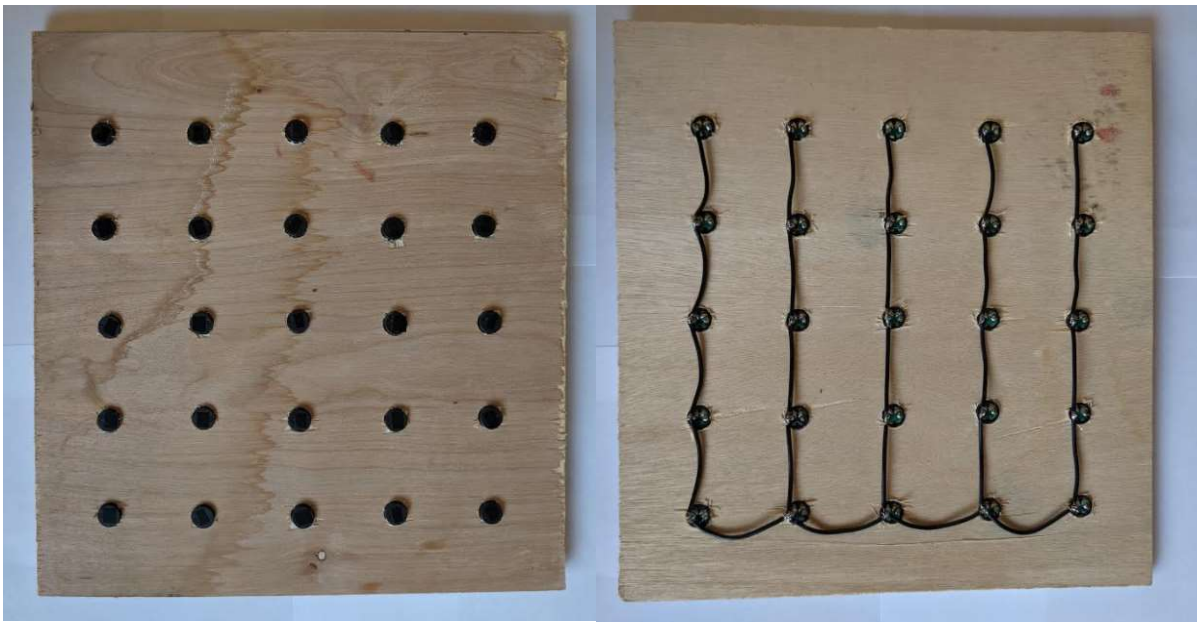


Figure 20: *Phased speaker array "antenna" fixture*

The voltage dividers and BJT inverting voltage amplifiers were then soldered on generic PTH PCB boards with each output soldered to the positive leads on the piezo-electric buzzers. Five 9V battery connectors were attached to power the amplifiers.

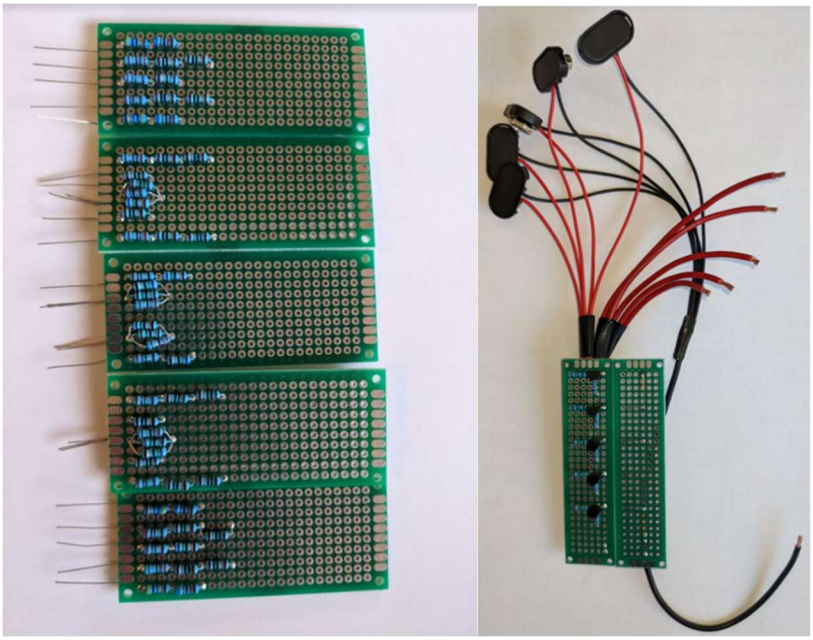


Figure 21: *Voltage dividers and BJT inverting voltage amplifiers*

Each of the five inputs for the amplifiers can be connected to the Arduino Mega through a screw/terminal block shield. The grounds for both the amplifier and piezo-electric buzzers were then soldered together and connected to the Arduino Mega ground pin.

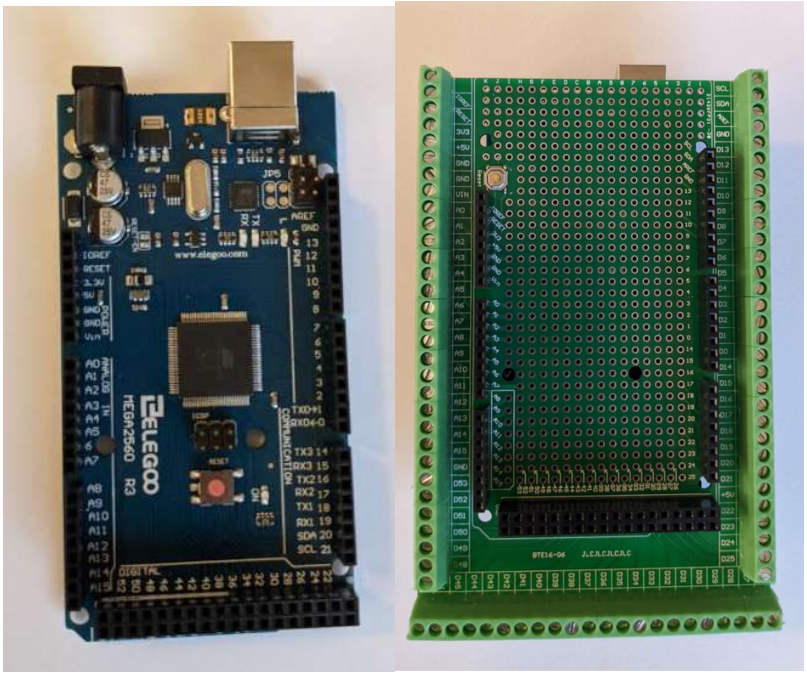


Figure 22: *Arduino Mega and screw/terminal block shield*

Phased Speaker Array Results

To test the performance of the phased speaker array, measurements were taken and plotted alongside the expected results for three different phase angles: $\theta_o = 0,\ \pi/4,$ and - $\pi/3$ (0°, 45°, and -60°). By comparing the theoretical and experimental results, the functionality of the array can be quantitatively analyzed. Using the decibel meter on a smartphone (Google Pixel 4a) and measuring every 5° at a distance of 20 meters from the array, adequate data can be collected for our final analysis. With a time delay of 11.67µs (estimated time to execute the tone.play() function in Arduino IDE) between the initialization of each of the five square wave voltage pulse trains powering the columns of the array, the main beam of audio signals at a frequency of 3136Hz will be created and centered at $\theta_o = 0$. At the time of data collection, the measured ambient background noise was found to be approximately 36dB.

Individual time delay	Total time delay between each column				
11.67µs	C1	C2	C3	C4	C5
	0	11.67µs	23.34µs	35.01µs	46.68µs

Figure 23: *Time delay between each array column when* $\theta_o = 0$

θ	dB at $\theta_o = 0$	θ	dB	θ	dB	θ	dB	θ	dB
-90°	36 dB	-50°	35 dB	-10°	47 dB	30°	41 dB	70°	34 dB
-85°	35 dB	-45°	36 dB	-5°	50 dB	35°	38 dB	75°	35 dB
-80°	35 dB	-40°	38 dB	0°	53 dB	40°	36 dB	80°	34 dB
-75°	34 dB	-35°	40 dB	5°	52 dB	45°	35 dB	85°	34 dB
-70°	36 dB	-30°	39 dB	10°	49 dB	50°	34 dB	90°	36 dB
-65°	37 dB	-25°	41 dB	15°	45 dB	55°	36 dB		
-60°	34 dB	-20°	43 dB	20°	46 dB	60°	34 dB		
-55	36 dB	-15°	46 dB	25°	43 dB	65°	37 dB		

Figure 24: *Data collected from the phase speaker array at* $\theta_o = 0$

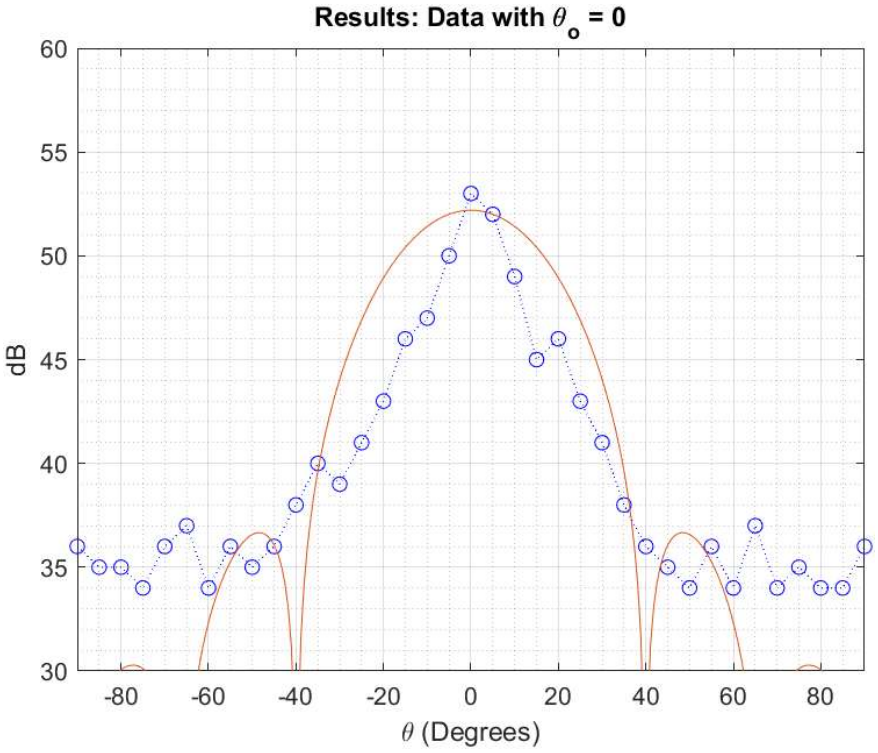


Figure 25: *Data collected from the phase speaker array at $\theta_o = 0$ plotted against the theoretical values.*

The main beam has a gain of 17dB above the ambient background noise threshold level of 36dB and follows the theoretical value of the complete pattern representation of the phased array extremely well. In-person, the effect of the phased array was very significant. When between angles of -20° and 20°, the sound was noticeably louder than that of larger values of θ.

The same method of data collection was used when measuring the performance of the array at a phase angle of $\theta_o = \pi/4$. When adding the correct time delay between the initialization of the square wave voltage pulse trains connected to each array column, the maximum value should be measured at 45°

Individual time delay	Total time delay between each column				
0.648ms	C1	C2	C3	C4	C5
	0ms	0.648ms	1.295ms	1.943ms	2.591ms

Figure 26: *Time delay between each array column when $\theta_o = \pi/4$*

θ	dB at $\theta_o = \pi/4$	θ	dB	θ	dB	θ	dB	θ	dB
-90°	34 dB	-50°	35 dB	-10°	37 dB	30°	47 dB	70°	44 dB
-85°	35 dB	-45°	36 dB	-5°	36 dB	35°	49 dB	75°	42 dB
-80°	37 dB	-40°	37 dB	0°	38 dB	40°	51 dB	80°	41 dB
-75°	36 dB	-35°	36 dB	5°	39 dB	45°	50 dB	85°	40 dB
-70°	34 dB	-30°	37 dB	10°	41 dB	50°	49 dB	90°	36 dB
-65°	36 dB	-25°	34 dB	15°	42 dB	55°	47 dB		
-60°	34 dB	-20	36 dB	20°	46 dB	60°	47 dB		
-55	34 dB	-15	35 dB	25°	48 dB	65°	45 dB		

Figure 27: *Data collected from the phase speaker array at* $\theta_o = \pi/4$

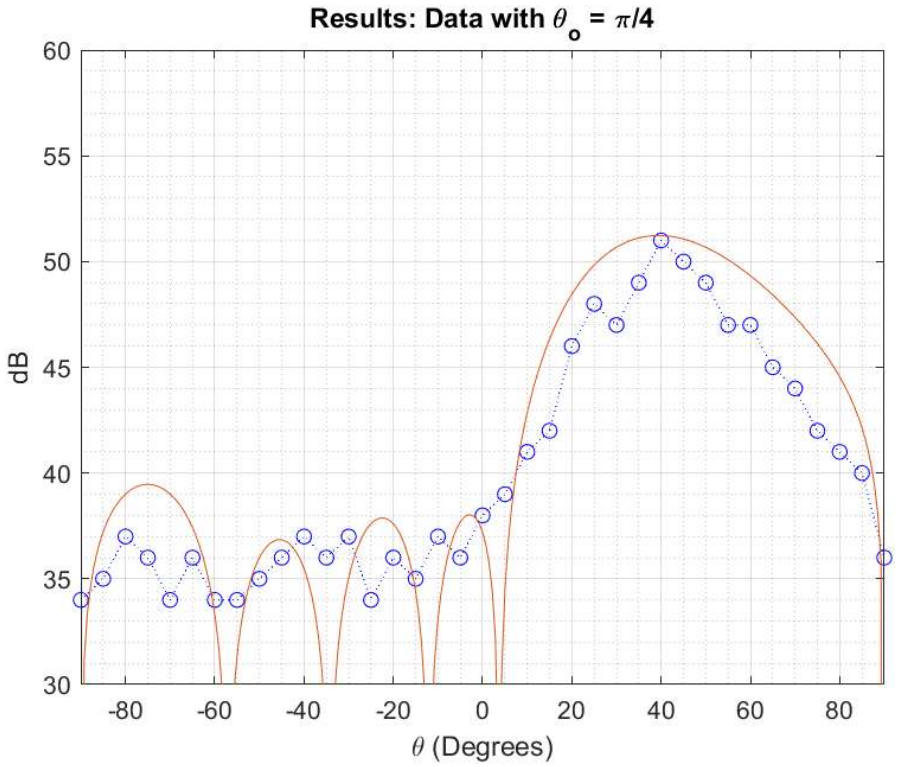


Figure 28: *Data collected from the phase speaker array at* $\theta_o = \pi/4$ *plotted against the theoretical values.*

The maximum value of the main beam at 40° (slightly lower than the predicted 45°) was measured to be 51dB, 15dB higher than the ambient background noise threshold. The maximum gain of the main beam was slightly lower than the unphased array of when $\theta_o = 0$. This is caused by the roll-off attenuation caused by the element pattern. As the phase angle increases, the maximum value of the main beam gain should decrease.

Finally, the last phase angle, $\theta_o = -\pi/3$, can be analyzed. To switch the phase angle to a negative value, the time delays applied to the array columns will be added in reverse. This is accomplished by setting column 5 as the base time delay of 0 and increasing the time delay for columns 4-1 respectively.

Individual time delay	Total time delay between each column				
0.793ms	C1	C2	C3	C4	C5
	3.173ms	2.380ms	1.586ms	0.793ms	0

Figure 29: *Time delay between each array column when* $\theta_o = -\pi/3$

θ	dB at $\theta_o = -\pi/3$	θ	dB	θ	dB	θ	dB	θ	dB
-90°	38 dB	-50°	50 dB	-10°	37 dB	30°	34 dB	70°	43 dB
-85°	40 dB	-45°	48 dB	-5°	36 dB	35°	37 dB	75°	42 dB
-80°	41 dB	-40°	49 dB	0°	33 dB	40°	33 dB	80°	42 dB
-75°	44 dB	-35°	47 dB	5°	34 dB	45°	38 dB	85°	39 dB
-70°	45 dB	-30°	46 dB	10°	34 dB	50°	40 dB	90°	36 dB
-65°	47 dB	-25°	44 dB	15°	36 dB	55°	41 dB		
-60°	49 dB	-20	43 dB	20°	36 dB	60°	42 dB		
-55	50 dB	-15	39 dB	25°	35 dB	65°	44 dB		

Figure 30: *Data collected from the phase speaker 34array at* $\theta_o = -\pi/3$

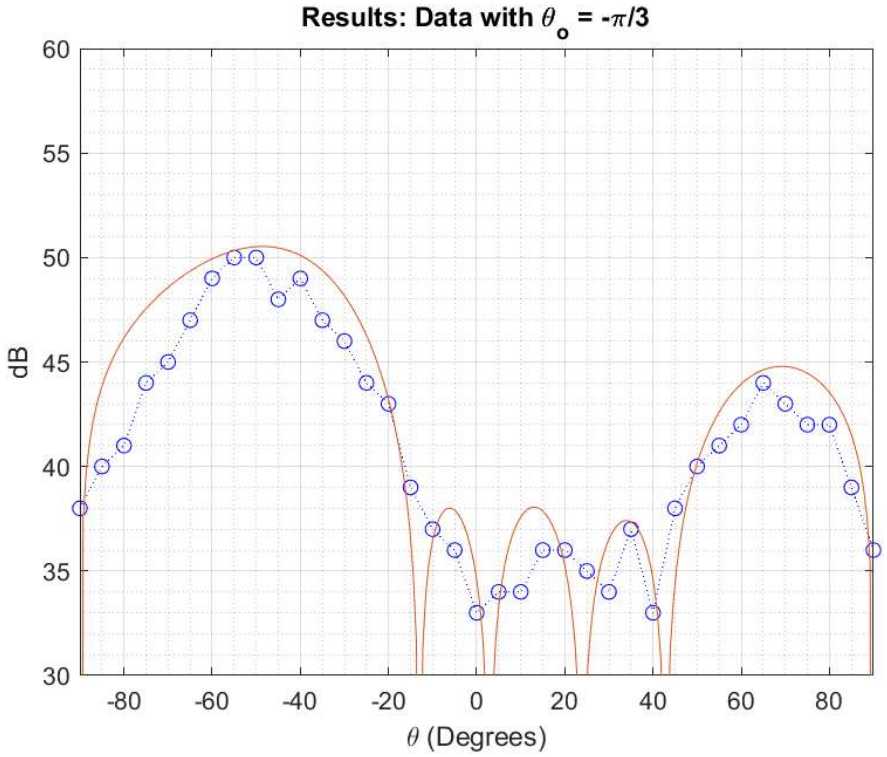


Figure 31: *Data collected from the phase speaker array at* $\theta_o = -\pi/3$ *plotted against the theoretical values.*

As shown in the plot, the phase angle of $\theta_o = -\pi/3$ is too extreme for the element spacing of 0.05m to eliminate grating lobes. In both the theoretical and experimental data, the main beam and grating lobe occurs around θ = -55° and θ = -65° respectively. Again, the effect of the roll-off attenuation caused by the element pattern is obvious as the gain of the main beam and grating lobe are both lower than the main beam of smaller phase delays. The gain of the main beam of 50dB (14dB above the ambient background noise level of 36dB) and the grating lobe had a maximum of 44dB (8dB above the ambient background noise level of 36dB).

Conclusion

Overall, the phased speaker array built was a complete success. The measured data followed the theoretical simulations extremely closely while capturing nuances like grating lobes occurring at large values of θ_o and slight directionality error caused by the element weighting. Creating a phased speaker array gave many insights to how active electronically scanned arrays function and, although not as useful as other transmitting array such as 5G antennas or radar, it gave a simplified and easily analyzed application of this beamforming technique.

Appendix I

Equations:

Linear Array Factor:

$$\mathrm{AF} = \sum_{n=1}^{N} A_n e^{j\frac{2\pi}{\lambda}x_n \sin\theta}$$

Linear Array Factor with a phase delay:

$$\mathrm{AF} = \sum_{n=1}^{N} A_n e^{j\left(\frac{2\pi}{\lambda}x_n \sin\theta - \frac{2\pi}{\lambda}x_n \sin\theta_o\right)}$$

Phase delay as a function of time:

$$\theta_o = \sin^{-1}\left(\frac{c}{2\pi d}\delta t\right)$$

Rectangular Array Factor:

$$\mathrm{AF} = \sum_{l=1}^{N*M} C_l\, e^{j\left(\frac{2\pi}{\lambda}x_l \sin\theta\cos\phi - \frac{2\pi}{\lambda}x_l \sin\theta_o \cos\phi_o\right)} e^{j\left(\frac{2\pi}{\lambda}y_l \sin\theta\sin\phi + \frac{2\pi}{\lambda}y_l \sin\theta_o \sin\phi_o\right)}$$

Element Pattern:

$$\mathrm{EP} = \cos^{\frac{\mathrm{EF}}{2}}\theta$$

Complete Pattern Representation:

$$F(\theta) = \mathrm{EP} * \mathrm{AF} = \cos^{\frac{\mathrm{EF}}{2}}\theta * \sum_{l=1}^{N*M} C_l\, e^{j\left(\frac{2\pi}{\lambda}x_l \sin\theta\cos\phi - \frac{2\pi}{\lambda}x_l \sin\theta_o \cos\phi_o\right)} e^{j\left(\frac{2\pi}{\lambda}y_l \sin\theta\sin\phi + \frac{2\pi}{\lambda}y_l \sin\theta_o \sin\phi_o\right)}$$

dB conversion:

$$F_{\mathrm{dB}} = 10\log_{10}(\mathrm{EP} * \mathrm{AF})^2$$

Time delay as a function of expected phase angle:

$$\delta t = \sin\theta_o \frac{2\pi d}{c}$$

Time delay as a function of expected phase angle with tone.play() execution time:

$$\delta t = \sin\theta_o \frac{2\pi d}{c} - 11.67\mu s$$

Appendix II

MATLAB Functions

find_array_signal_linear:

```
 function [AF, EP, theta] = find_array_signal_linear(x, weights, lambda,
theta_phase, EF)
 i = 1;
 resolution = pi/250;
     for theta = -pi/2:resolution:pi/2
         cosx = sin(theta);
         cosx_phase = sin(theta_phase);
         Tx = 2*pi/lambda.*x.*cosx;
         Tx_phase = 2*pi/lambda.*x.*cosx_phase;
         AF(i) = abs((sum(weights.*exp(1j*((Tx-Tx_phase))))));
         EP(i) = abs(cos(theta).^(EF/2));
         i = i + 1;
     end
 theta = [-pi/2:resolution:pi/2];
 end
```

find_array_signal:

```
 function [AF, EP, phi, theta] = find_array_signal(x, y, weights, lambda,
phi_phase, theta_phase, EF)
 i = 1;
 j = 1;
 resolution = pi/250;
 for phi = 0:resolution:2*pi
     for theta = -pi/2:resolution:pi/2
         cosx = cos(phi).*sin(theta);
         cosy = sin(phi).*sin(theta);
         cosx_phase = cos(phi_phase).*sin(theta_phase);
         cosy_phase = sin(phi_phase).*sin(theta_phase);
         Tx = 2*pi/lambda.*x.*cosx;
         Ty = 2*pi/lambda.*y.*cosy;
         Tx_phase = 2*pi/lambda.*x.*cosx_phase;
         Ty_phase = 2*pi/lambda.*y.*cosy_phase;
         AF(i,j) = abs((sum(weights.*exp(1j*((Tx-Tx_phase))).*exp(1j*(Ty-
Ty_phase)))));
         EP(i,j) = abs(cos(theta).^(EF/2));
         i = i+1;
     end
     j = j+1;
     i = 1;
 end
```

```
phi = [0:resolution:2*pi];
theta = [-pi/2:resolution:pi/2];
end
```

find_taylor_weights:

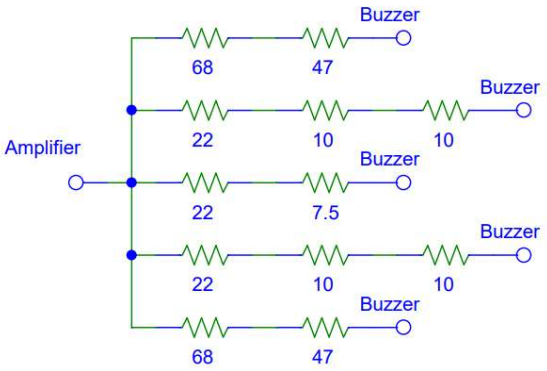
```
function [ weights ] = find_taylor_weights(n_rect,sll,nbar)
points = length(n_rect);
r = 10^(abs(sll)/20);
a = log(r+(r*r-1)^0.5) / pi;
sigma2 = nbar^2/(a*a+(nbar-0.5)^2);
%Compute Fm, the Fourier coefficients of the weight set
for m=1:(nbar-1)
 for n=1:(nbar-1)
 f(n,1)=1-m*m/sigma2/(a*a+(n-0.5)*(n-0.5));
 if n ~= m
 f(n,2)=1/(1-m*m/n/n);
 end
 if n==m
 f(n,2)=1;
 end
 end
 g(1,1)=f(1,1);
 g(1,2)=f(1,2);
 for n=2:(nbar-1)
 g(n,1)=g(n-1,1)*f(n,1);
 g(n,2)=g(n-1,2)*f(n,2);
 end
 F(m)=((-1)^(m+1))/2*g(nbar-1,1)*g(nbar-1,2);
end
jj = [1:points]';
xx = (jj-1+0.5)/points - 1/2;
W = ones(size(jj));
mm = [1:nbar-1];
W = W + 2*cos(2*pi*xx*mm)*F';
WPK = 1 + 2*sum(F);
wgt = (W / WPK)';
weights = wgt;
end
```

Appendix III

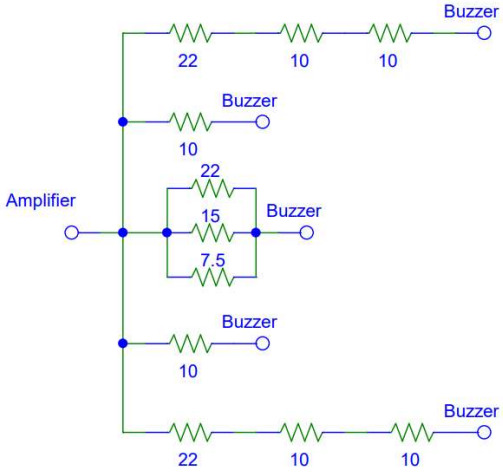
Resistor Values for Voltage Dividers:

	Column 1	C2	C3	C4	C5
Row 1	115.0Ω (47+68)	42.0 Ω	29.5Ω	42.0 Ω	115.0Ω
R2	42.0Ω (22+10+10)	10.0Ω (10)	4.07Ω	10.0Ω	42.0 Ω
R3	29.5Ω (22+7.5)	4.07Ω (22‖15‖7.5)	0Ω	4.07Ω	29.5Ω
R4	42.0 Ω	10.0Ω	4.07Ω	10.0Ω	42.0 Ω
R5	115.0Ω	42.0 Ω	29.5Ω	42.0 Ω	115.0Ω

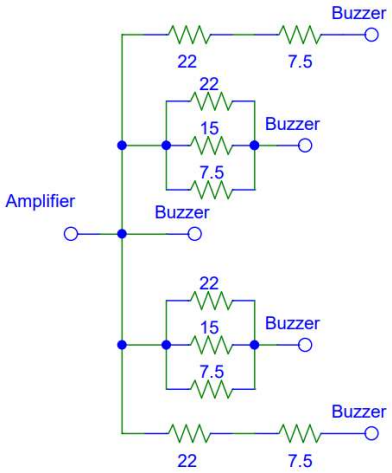
Columns 1 & 5:



Columns 2 & 4:



Columns 3:



Appendix IV

Arduino Mega Code

```
#include <Tone.h>
Tone tone1;
Tone tone2;
Tone tone3;
Tone tone4;
Tone tone5;
float pi = 3.14159;
float theta = 0*pi/180;
float toneTime = 0.0000117;
float distance = .05;
float Time = 0;
float c = 343;
int i;
void setup()
{
  Serial.begin(9600);
  tone1.begin(26);
  tone2.begin(28);
  tone3.begin(30);
  tone4.begin(32);
  tone5.begin(34);
  if(theta > 0){
  Time = sin(theta)*2*pi*distance/c - toneTime;
  tone1.begin(26);
  tone2.begin(28);
  tone3.begin(30);
  tone4.begin(32);
  tone5.begin(34);
  }
  if(theta == 0){
  Time = 0;
  tone1.begin(26);
  tone2.begin(28);
  tone3.begin(30);
  tone4.begin(32);
  tone5.begin(34);
  }
  if(theta < 0){
  theta = -1 * theta;
  Time = sin(theta)*2*pi*distance/c - toneTime;
  tone1.begin(34);
  tone2.begin(32);
  tone3.begin(30);
  tone4.begin(28);
  tone5.begin(26);
  }

  tone1.play(NOTE_G7);
  delayMicroseconds(Time);
  tone2.play(NOTE_G7);
  delayMicroseconds(Time);
  tone3.play(NOTE_G7);
  delayMicroseconds(Time);
  tone4.play(NOTE_G7);
  delayMicroseconds(Time);
  tone5.play(NOTE_G7);
}
void loop(){}
```

References

Brock, Billy C. “The Application of Taylor Weighting, Digital Phase Shifters, and Digital Attenuators to Phased-Array Antennas.” *SANDIA REPORT*, 2008, doi:10.2172/932884.

Stutzman, Warren L., and Gary A. Thiele. *Antenna Theory and Design*. Wiley, 2013.

Engda, Tewelgn Kebede, et al. “Massive MIMO, MmWave and MmWave-Massive MIMO Communications: Performance Assessment with Beamforming Techniques.” 2020, doi:10.21203/rs.3.rs-69959/v1.

Brown, Arik D. *Electronically Scanned Arrays: Matlab Modeling and Simulation*. CRC Press, 2017.

Björnson, Emil, et al. “Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency.” *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, 2017, pp. 154–655., doi:10.1561/2000000093.