Directing Acoustic Radiation using a Phased Array of Piezoelectric Transmitters

by
Daniel Michael Rodgers

B.S., Massachusetts Institute of Technology (2009)
Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the
Massachusetts Institute of Technology
August, 2010

Author _____
Department of Electrical Engineering and Computer Science
July 28, 2010

Certified by _____
Yoel Fink
Associate Professor, DMSE
M.I.T Thesis Supervisor

Accepted by _____
topher J. Terman
Chairman, Department Committee on Graduate Theses

Directing Acoustic Radiation using a Phased Array of Piezoelectric Transmitters

by
Daniel Michael Rodgers

Submitted to the
Department of Electrical Engineering and Computer Science

July 28th, 2010

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

This thesis presents an acoustic phased array system utilizing piezoelectric transducers. The system is capable of operating at arbitrary frequencies into the low megahertz range, with a trade-off between phase accuracy and operational frequency. A maximum of sixteen elements can be introduced to the array and all elements are capable of operating at arbitrary phases relative to each other. Waveform generation is done in MATLAB and LabVIEW is used to interface between a host PC issuing commands and the array itself. Results from tests run using a four element array to demonstrate beam steering and focusing at 11 kHz are included and discussed.

Thesis Supervisor: Yoel Fink
Title: Associate Professor, Department of Material Science and Engineering

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The project described in this thesis is motivated by recent advances in fiber materials. A fiber is a slender elongated material which can be spun into threads or rope, used as a component in composite materials, or even matted into sheets or meshes. Throughout their history fibers have been considered static devices which are incapable of changing their properties over a wide frequency range. Past attempts to introduce time-dependant variations to fibers have been largely limited by the inert nature of traditional glassy fiber materials (S. Egusa). However, new techniques have been discovered that allow for the inclusion of a ferroelectric polymer layer in composite fibers. This layer can be electronically contacted and encapsulated by an insulating polymer resulting in meters of piezoelectric fiber after drawing from a preform. These fibers exhibit a piezoelectric response and acoustic transduction over frequencies in the kilohertz to megahertz range(S. Egusa).

These advances in fiber technology open the door for many exciting applications. The ability to create arbitrary sized piezoelectric threads and meshes could find use in fabrics that act as microphones, in tiny filaments capable of measuring blood flow or pressure in small areas, and very large scale nets to monitor water flow. The application explored in this project involves the use of an array of piezoelectric elements in order to direct acoustic energy in arbitrary directions. This type of array, known as a phased array, has uses in ultrasonic imaging, object detection, and non-destructive testing.

The phased array designed and built for this project combines digital and analog signal processing in a novel way. A block diagram of the total system is shown in Figure 1.

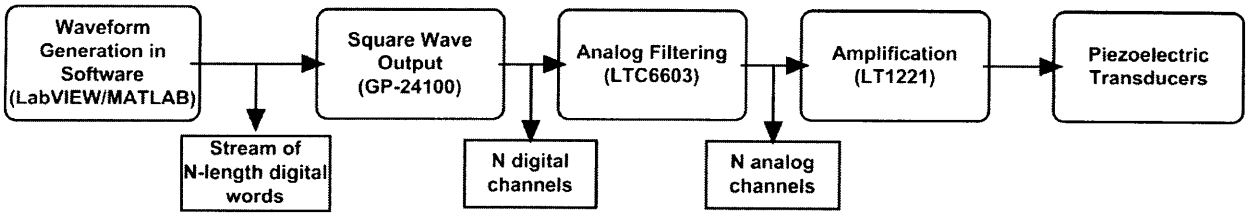


Figure 1: Block diagram of phased array system

The digital waveforms required to control up to 16 piezoelectric transducers are generated in MATLAB and, using LABVIEW, are loaded into a GP-24100. The GP-24100 is a USB device that is capable of outputting arbitrary 16-bit digital patterns at up to a 100MHz clock frequency. Each bit of this digital pattern represents an element in the array. The digital square wave generated for each element is filtered by an analog chip in order to remove unwanted higher order harmonics, leaving only the desired fundamental frequency. This filtered waveform is then amplified and applied to the proper transducer.

This thesis will begin by discussing background in chapter 2, including the piezoelectric effect and how it allows for materials to act as acoustic transducers as well as an overview of the theory behind phased arrays. Chapter 3 focuses on the software behind the demonstrated phased array system, including the user interface, generation of the waveforms, and compensation techniques applied in the digital domain. The hardware designed to filter and amplify the waveforms generated by the software algorithms is outlined in detail in chapter 4. Results from a four element array operating in the audible frequency range are presented and discussed in chapter 5. Finally, chapter 6 contains a summary of the system and presents possible improvements.

Chapter 2

Background

Piezoelectric Effect

At the heart of the operation of fibers or other materials used as acoustic transducers in this phased array is the piezoelectric effect. Piezoelectricity is the ability of some materials to generate an electric field in response to a mechanical stress, or conversely to produce a mechanical stress or strain in response to an applied electric field. This effect results from the atomic structure of the material. In a piezoelectric crystal, for example, the positive and negative electric charges inside of the material are separate but symmetrically distributed, resulting in a crystal which is electrically neutral(Walter Heywang). When a mechanical stress is applied, the symmetry is disrupted and a voltage is generated as a result of the now present charge asymmetry. When a voltage is applied to the material, the symmetry is again disrupted, and the dimensions of the material change as a result. For the purposes of this project it is not necessary to delve into a mathematical description of this effect, it will be assumed to exist and work in the transducers used in the array.

Phased Array

As mentioned earlier, a phased array is an array of transducers in which the relative phase of each element can be independently controlled. This control over the relative phase allows for the radiation pattern of the entire array to be directed and focused in a much more precise manner than would be possible for a single element. To examine how this result occurs, it is useful to start with the radiation pattern of a single element and then generalize to N elements with arbitrary phases.

Single Slit Diffraction

The radiation pattern of a single element in the array can be modeled using the diffraction pattern of a slit of the width of the element. The analysis presented here applies Huygens principle to single slit. The Huygens principle states that each point in a wavefront is in fact the center of a new disturbance and the source of a new set of waves(Bevan B. Baker). The contributions of these point sources, which have a well defined radiation pattern in all directions, can be summed to determine advancing wavefront. In the case of a line source or single slit diffraction, this means that each point in the slit acts as a point source and the wavefront produced by the slit can be found by summing the contributions of each point source. Thus, in the case of a single slit, the wave function can be written as the integral of the point source wave function over the entire slit(Ajoy K. Ghatak) :

$$\Psi(r) = \frac{i*A}{\lambda}\int \frac{e^{-ikr}}{r} \mathrm{d}slit \quad \textbf{(2.1)}$$

Where λ is the wavelength, k is the wavenumber (2π / λ), r is the distance from the slit, and A is the amplitude of the plane wave in the plane of the slit. Leaving aside intermediate steps, the following expression for the intensity as a function of θ, the angle from the center of the slit, is obtained after integrating and squaring equation 2.1:

$$I(\theta) = I_o\left[\mathrm{sinc}\left(\frac{\pi a}{\lambda}\sin\theta\right)\right]^2 \quad \textbf{(2.2)}$$

Where I_o is a constant, and a is the width of the slit. Plotting this function for different values of a/λ gives the following spatial profiles for different slits or element sizes.

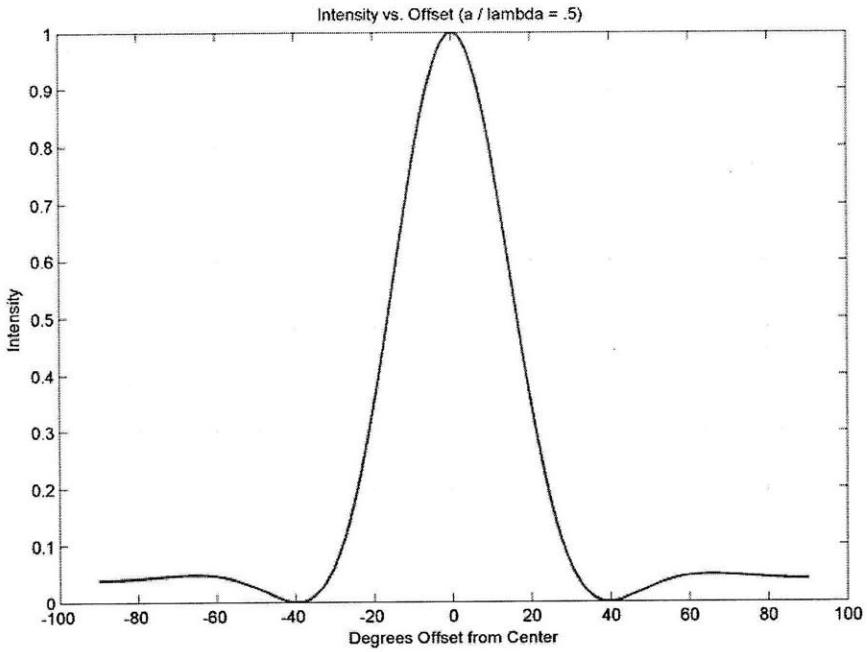


Figure 2: Intensity distribution for slit of width a = .5*λ

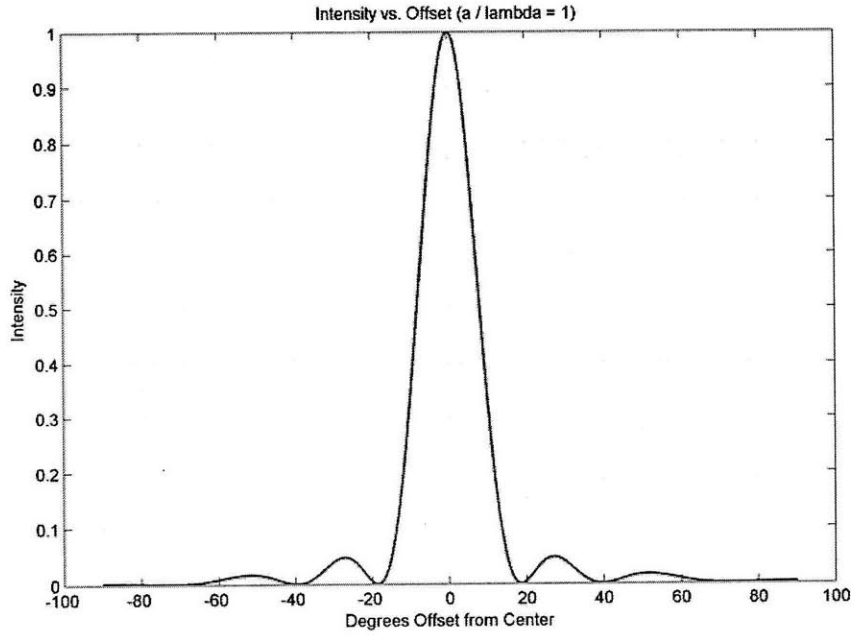


Figure 3: Intensity distribution for slit of width a = λ

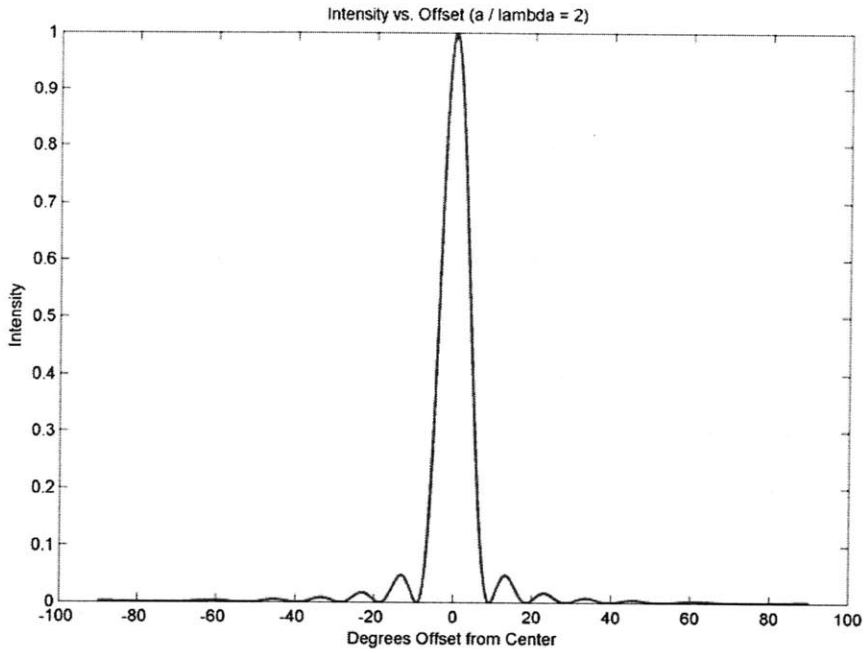


Figure 4: Intensity distribution for slit of width a = 2*λ

N slit diffraction

Forgoing the derivation, it can be shown that for an array of N equally sized slits (or elements) with a center to center separation of d, the intensity profile as a function of offset angle from the center of the array (θ), is the following (Ghatak):

$$I(\theta) = I_o \left[\text{sinc}\left(\frac{\pi a}{\lambda}\sin\theta\right)\right]^2 * \left[\frac{\sin\left(\frac{N\pi d}{\lambda}\sin\theta\right)}{\sin\left(\frac{\pi d}{\lambda}\sin\theta\right)}\right]^2 \quad \textbf{(2.3)}$$

Plots for various values of d and N with a/λ equal to 1 are shown below:

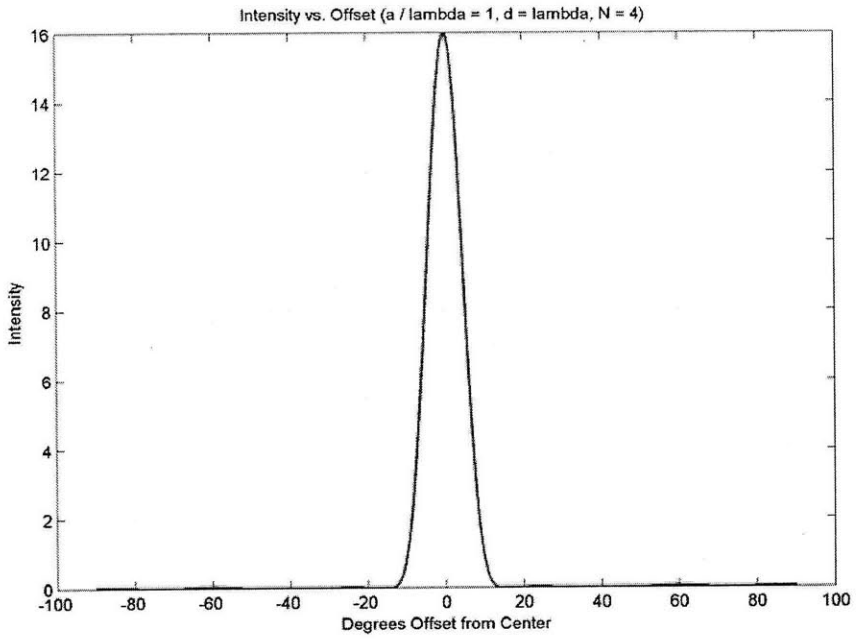


Figure 5: Intensity distribution for 4 slits of width a = λ spaced 1 wavelength apart

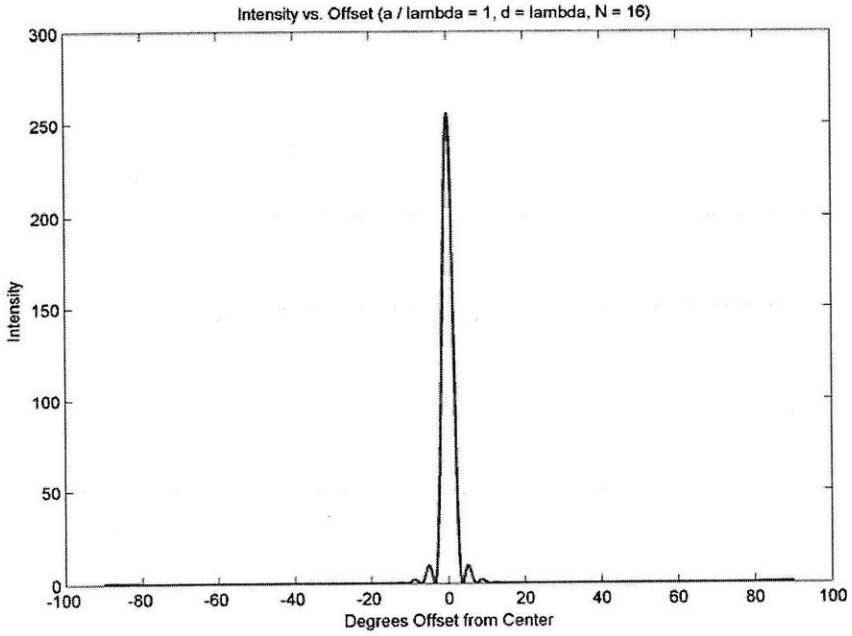


Figure 6: Intensity distribution for 16 slits of width a = λ spaced 1 wavelength apart

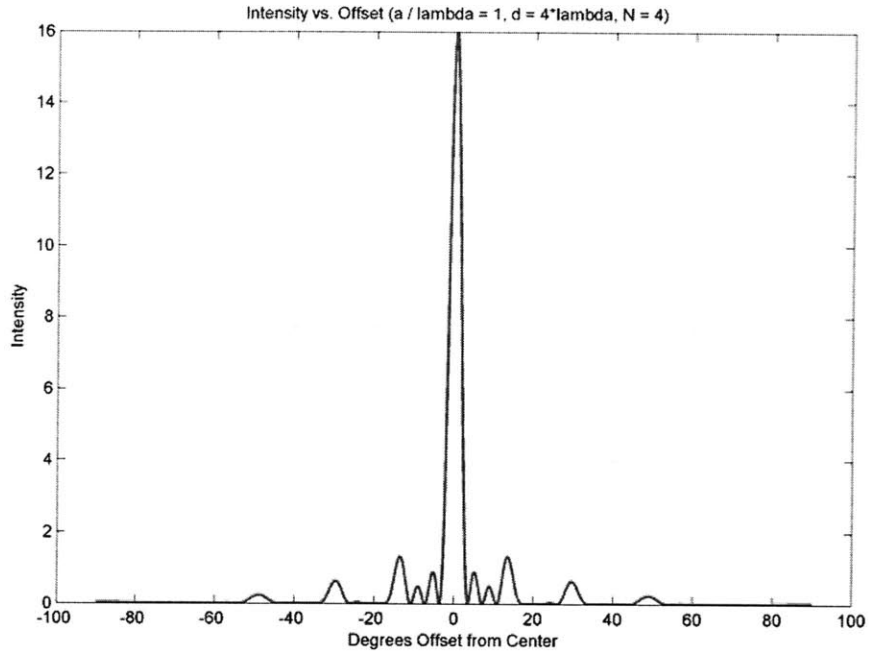


Figure 7: Intensity distribution for 4 slits of width a = λ spaced 4 wavelengths apart

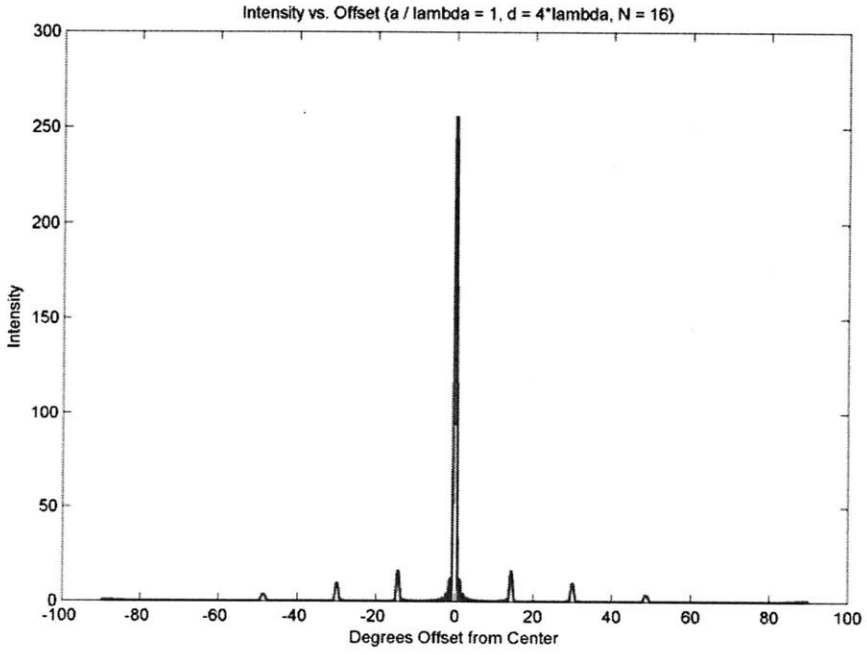


Figure 8: Intensity distribution for 16 slits of width a = λ spaced 4 wavelengths apart

It is apparent that adding more elements further focuses the intensity spatially, as this effectively increases the area of the array. Increasing element spacing also has the effect of further focusing the

main beam, but this comes at the expense of increased energy in undesirable areas, called "side-lobes". This increased focusing ability is a large reason why arrays of this nature are useful.

Steering

When a phase lag is added between successive elements, it becomes possible to adjust the intensity profile in a chosen direction. In order to determine the delay law between successive elements, and correspondingly find the phase difference between successive elements, it most useful to examine the geometry of the desired wavefront in terms of wave velocity, steering angle, and element spacing.

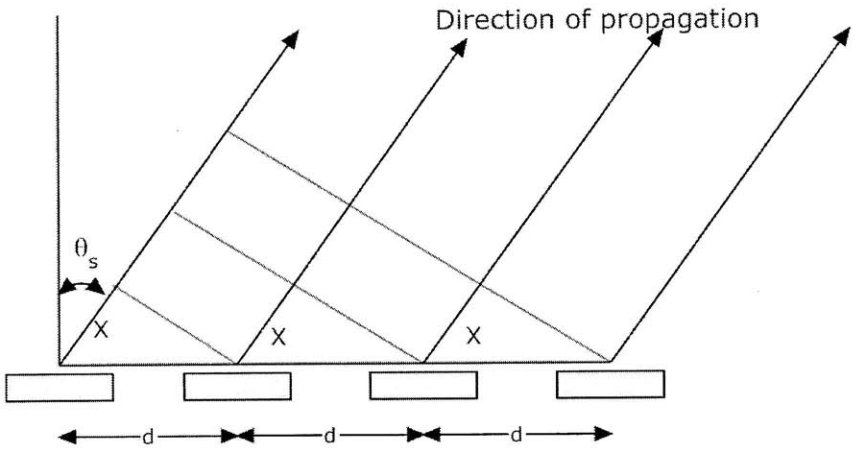


Figure 9: Geometric view of beam steering

Using the geometry shown in Figure 9, where the lines without arrows represent the location of the desired constructive interference of the wavefront, the following expression for the distance x is geometrically obtained:

$x = d \sin \theta_s$ **(2.4)**

This represents the required distance between the peaks of adjacent transducers in order to steer in the direction Θ_s. The distance x can then be written in terms of the required time delay by introducing the wave velocity and remembering that distance equals rate times time, resulting in:

$$t_{del} = \frac{d \sin \theta_s}{c} \textbf{(2.5)}$$

Finally, the time difference can be converted into a phase shift by utilizing the fact that:

$$t_{del} = \frac{\varphi}{360° * f} \textbf{(2.6)}$$

Giving the following final expression for the phase shift (in degrees) between successive elements:

$$\varphi = \frac{360° * f * d \sin \theta_s}{c} \textbf{(2.7)}$$

Thus, for a given operating frequency, speed of sound, and element spacing it is possible to determine the phase shift between elements to steer the wavefront of the array in a given direction.

Focusing

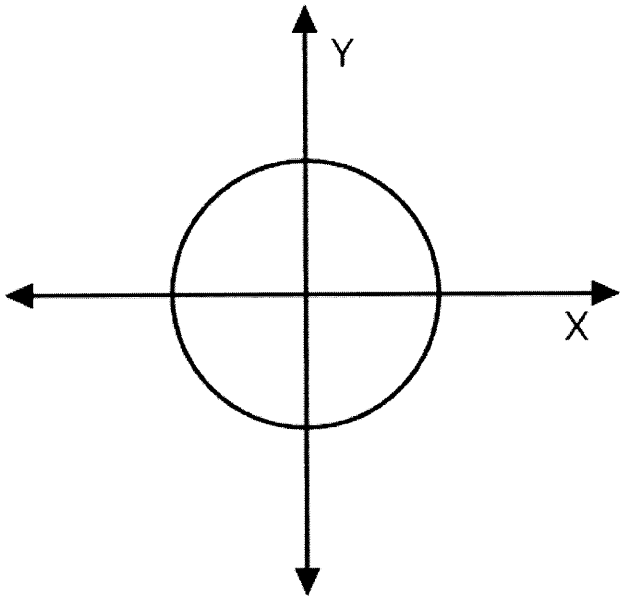


Figure 10: 2D Lens

In addition to steering the wavefront in a desired direction, it is also possible to develop delay laws, or relative phase shifts, that allow the intensity of the wavefront to peak at a desired distance from the array. This is accomplished using a parabolic phase shift akin to that of an optical lens. Observing that the phase transformation resulting from a 2 dimensional lens such as the one in Figure 10 can be written as shown in (2.8) (Goodman):

$$T(x,y) = \exp\left(-j * \left(\frac{k}{2*focal}\right) * (x^2 + y^2)\right) \textbf{(2.8)}$$

Where focal is the focal point of the lens, and x and y are the coordinates at which the incoming wavefront encounters the lens. In order to apply this law to a 1 dimensional phased array, the y coordinate is set to zero, and the phase of a given element for a desired focusing distance is calculated using the element's x coordinate as depicted in Figure 11. The resulting expression is shown in equation (2.9).

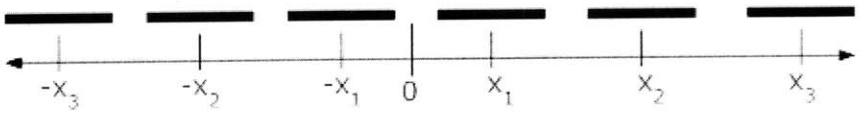


Figure 11: Array with labeled X axis

$$Phase(n) = \angle \exp\left(-j * \left(\frac{k}{2*focal}\right) * (x_n^2)\right) \quad \textbf{(2.9)}$$

(2.9) results in a parabolic phase profile whose slope is determined by the desired focal distance. Example phase profiles for varying focusing distance for a six element array with elements spaced one wavelength apart and operating in air at 11 kHz (λ=.0312cm) are shown below.

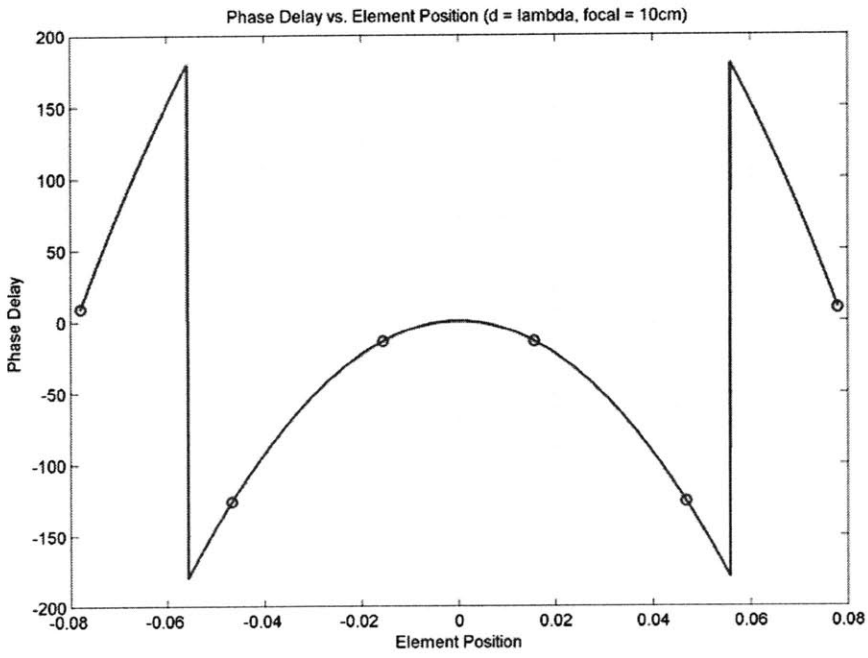


Figure 12: Phase profile for focusing at 10cm with λ=3.12cm

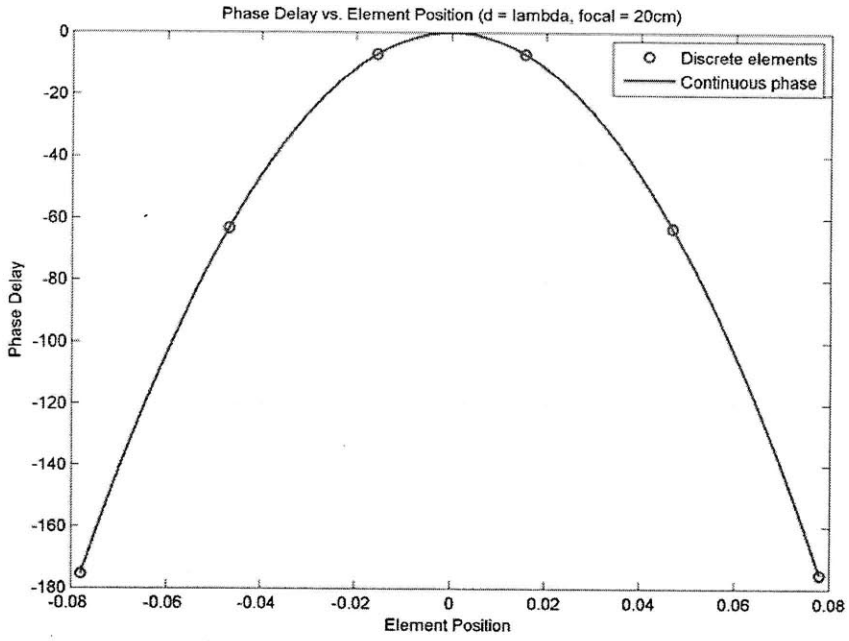


Figure 13: Phase profile for focusing at 20cm with λ=3.12cm

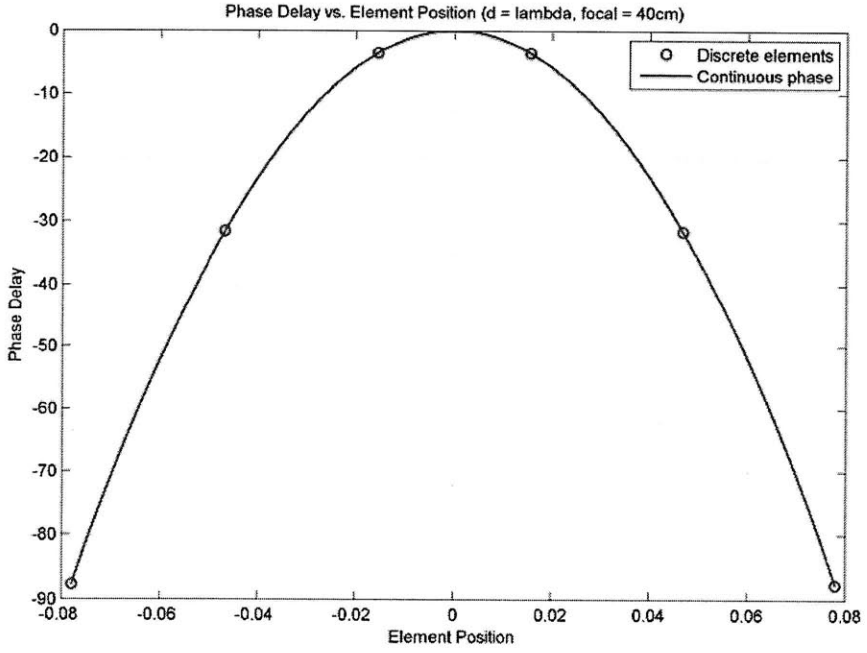


Figure 14: Phase profile for focusing at 40cm with λ=3.12

Note that the discontinuity in Figure 12 is the result of the phase delay wrapping around, i.e. a phase delay of -190° is equivalent to a phase lead of 170°. As might be expected, for further focusing

distances the phase profile flattens out. To focus the array infinitely far away, all elements will be in phase.

Filtering of Square Waves

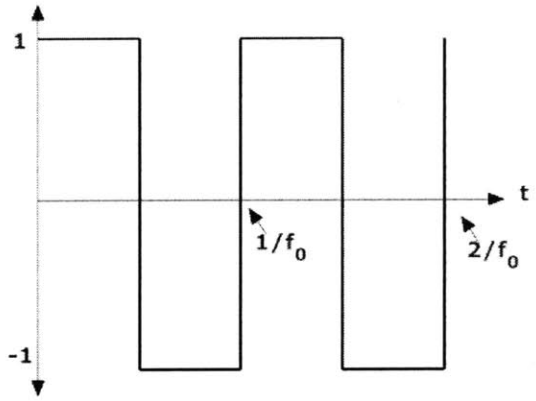


Figure 15: Example square wave

The phased array system described within this thesis accomplishes the aforementioned steering and focusing calculations in the digital domain. As a result of having only two discrete output levels available for a single channel, it is only possible to generate shifted square waves. In order to drive the transducers at only the desired frequency, these square waves are filtered to suppress the undesired frequency components as much as possible. Using Fourier analysis, the frequency spectrum of the square wave at a frequency of f_0 shown in Figure 15 can be represented as follows (Weisstein):

$$f(x) = \frac{4}{pi}\sum_{n=1,3,5\ldots}^{\infty}\frac{1}{n}\sin\left(n * 2\pi * t\right) \textbf{(2.10)}$$

All frequencies above the fundamental frequency f_0 are unwanted and must be filtered out. Luckily, only odd harmonics are present and the amplitude of the higher order harmonics is already less than the amplitude of the fundamental. Therefore, the filtering requirements for removing these

harmonics are not too strict, and the chosen low pass filter easily suppresses the effects of the higher order harmonics.

Chapter 3

Software Interface

Overview

The focus of this chapter will be the software used to control the phased array. The chapter will begin with a description of the aspects of the array which can be adjusted and controlled in software. Next, a description of the LabVIEW interface will be provided followed by an outline of the MATLAB script used by LabVIEW to generate the digital waveforms. This is followed by an overview of the capabilities of the GP-24100 arbitrary waveform generator and a summary of the advantages and limitations of generating digital waveforms as the basis for a phased array as chosen for this project.

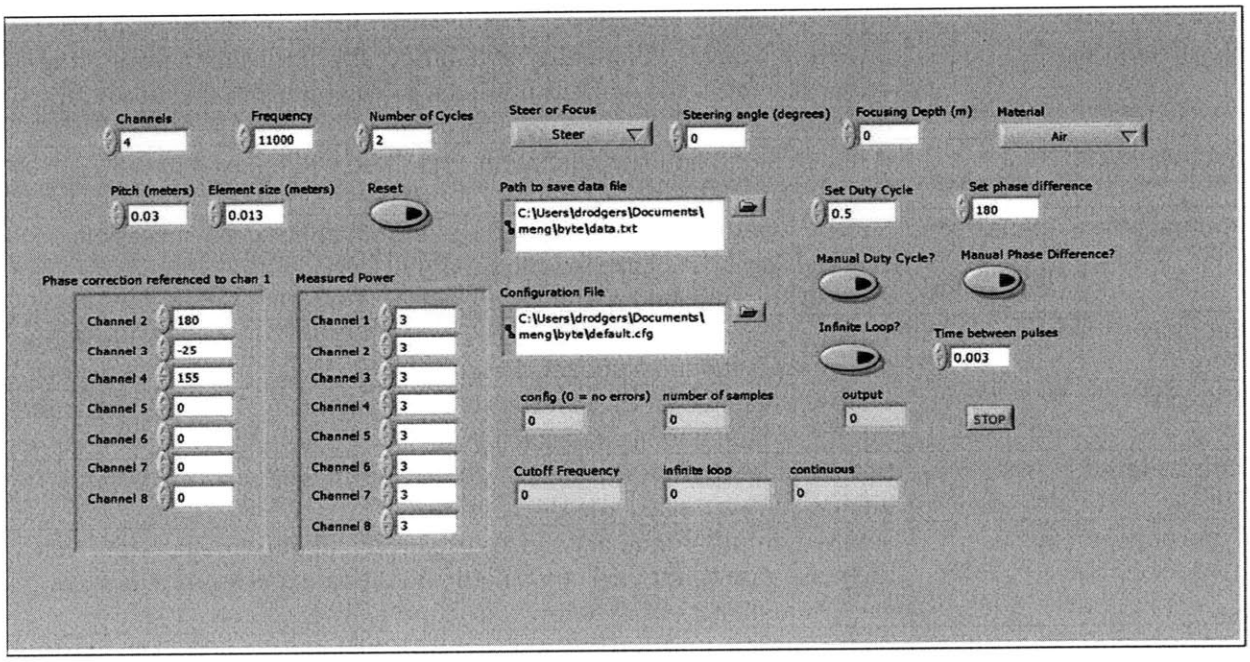


Figure 16: Front panel of LabVIEW user interface

Overview of User Controls

The following is a list of inputs available to the user of this system and a brief description of each. For reference, the user interface can be seen in Figure 16.

Channels	The number of elements in the array.
Frequency	The desired operating frequency of the array.
Cycles	The number of cycles at the desired operating frequency that will be generated. In pulse mode this is the length of the pulse, in continuous mode this determines only the length of the waveform that is looped and should be set to a small number to increase the speed of waveform generation.
Pitch	The center to center distance of the elements in the array.
Element size	The width of a single element in the array.
Phase correction	Relative to element 1, the phase shift to apply to each channel in order to compensate for differences present in the analog portion of each channel.
Measured power	The measured output power of each channel at a 50 percent duty cycle. If desired, the software can compensate for differences in the analog amplification between channels by adjusting the duty cycle of the channels.
Steering angle	The desired angle to steer the wavefront when in steering mode.
Focusing depth	The desired focal length when in focusing mode.
Steer or Focus	Choose whether to steer or focus the wavefront.
Manual duty cycle	Choose whether to manually set a duty cycle for all channels or if the duty cycle for each channel should be calculated from the measured powers.
Set duty cycle	When in manual duty cycle mode, this value will be used for all channels.
Manual phase difference	Choose whether to manually set a phase offset between successive channels or if the phase for each channel should be calculated from the desired steering angle or focusing depth.
Set phase difference	When in manual phase difference mode, this will be used as the phase difference between adjacent channels.
Material	Choose the material that the array will be operating in. This determines the speed of sound used in calculations.
Infinite loop	Determines whether the generated waveform will be looped indefinitely or pulsed at a chosen interval.
Time between pulses	Set the minimum time between pulses when in pulse mode. Note that the pulse rate is not well controlled and the time between pulses can vary, but will never fall below this value.
Configuration file path	The location on the host PC of the configuration file for the GP-24100.
Path to save data file	The location on the host PC to save the data file containing the generated waveform.
Stop	When in pulse mode, this aborts operation.
Reset	When in continuous mode, this aborts operation.

Table 1: User inputs to LabVIEW interface

LabVIEW Interface

The purpose of the LabVIEW interface is to provide the user of the array with a front end which controls MATLAB scripts and Windows DLL calls to the GP-24100. LabVIEW is responsible for passing the user inputs along to the MATLAB script which will be described in the following section. After MATLAB has processed the input and produced the desired digital pattern, LabVIEW configures the GP-24100 using the supplied configuration file and user inputs by making DLL calls to application controlling the GP-24100. After the pattern is loaded onto the GP-24100, LabVIEW commands the device to begin outputting the loaded data using the supplied settings. For a detailed view of the LabVIEW block diagram, see Appendix B.

Generation of basic digital pattern

The generation of the digital waveform to be output to the GP-24100 is done using a MATLAB script. The first step to generating the desired digital square wave is to calculate the nominal time that a channel should be at the logic levels "high" or "low" for a single cycle at the desired operating frequency and a 50 percent duty cycle. This is done using the following simple formula.

$$Ton_{nom} = Toff_{nom} = \frac{1}{frequency} * .5 \quad \textbf{(3.1)}$$

Next, this time must be converted into an equivalent number of digital samples (or clock cycles) at the GP-24100 clock frequency. The nominal samples on and off are calculated by dividing the desired time on and off by the time of a single sample and then rounding to obtain the nearest possible integer number of samples as follows:

$$Samples\_on_{nom} = round(\frac{Ton_{nom}}{1/clock_{freq}}) \quad \textbf{(3.2)}$$

While any clock frequency up to 100MHz is possible, a higher clock frequency allows for increased precision, so it does not make sense to lower the operating clock. These values are used for reference throughout pattern generation to ensure that all channels contain the same number of samples per cycle for arbitrary duty cycles.

Duty Cycle Calculation

To adjust for a difference in gain between channels in the analog portion of the signal processing chain, a technique has been developed to set the duty cycle of each channel such that the amplitude of the fundamental frequency at each channel is reduced to match the lowest measured amplitude. The amplitude of the fundamental frequency of a square wave of amplitude A and duty cycle D can be shown to be:

$$Amp = \sqrt{\frac{A^2}{\pi^2} * (1 - \cos(2\pi D))} \quad \textbf{(3.3)}$$

A plot of the amplitude at the fundamental frequency versus duty cycle for can be seen in Figure 17, as well as measured results for various duty cycles and the corresponding fundamental amplitude as reported by a digital oscilloscope. The input to filter curve represents the reported fundamental amplitude of a square wave with the corresponding duty cycle, and the output to fiber curve represents the reported fundamental amplitude after that square wave has been filtered to eliminate higher order harmonics.

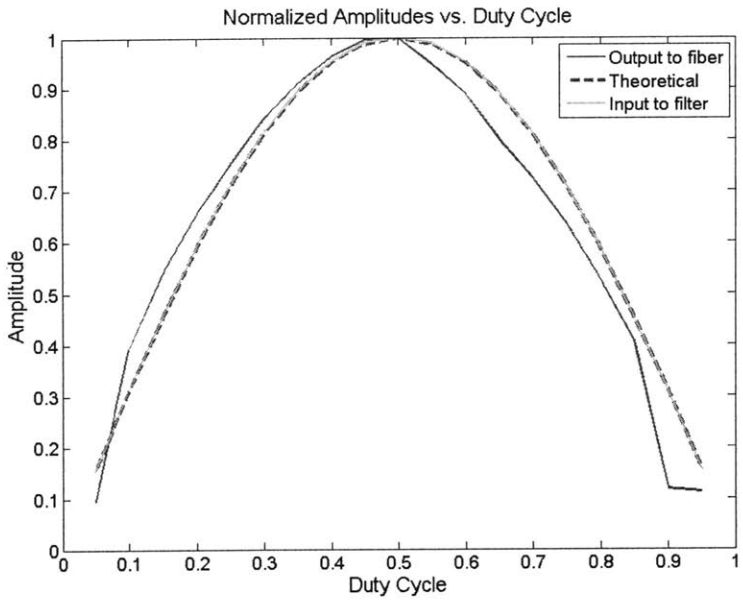


Figure 17: Fundamental amplitude vs. duty cycle of square wave

Recognizing that the amplitude at the fundamental is a maximum for D equals .5, the channel with the smallest measured amplitude is kept at a duty cycle of .5. The desired duty cycle for all other channels is calculated by assuming the gain in the channel is a constant and then calculating the value of D that will reduce the measured amplitude for that channel to the smallest measured amplitude among all channels. The result is the following expression for the duty cycle of a channel:

$$D = \frac{1}{2\pi} * \cos^{-1}(1 - \left(.2026423673 * \pi^2 * \frac{Ad}{Am}\right)) \quad \textbf{(3.4)}$$

Where Ad is the desired amplitude and Am is the measured amplitude for that channel. Using this it is possible to digitally correct for gain mismatches between channels.

Generation of Individual Channels

With the duty cycle determined for each channel, whether it was calculated from measured amplitudes or input by the user, it is now possible to calculate the proper number of samples each

channel should be kept high and low. The technique is the same as in the general case, only now the times are adjusted using the determined duty cycle:

$$Ton = \frac{1}{frequency} * D \quad \textbf{(3.5)}$$

$$Ton = \frac{1}{frequency} * (1 - D) \quad \textbf{(3.6)}$$

$$Samples\_on = round(\frac{Ton}{clock\_period}) \quad \textbf{(3.7)}$$

$$Samples\_off = round(\frac{Toff}{clock\_period}) \quad \textbf{(3.8)}$$

To create a full waveform for each channel, all that remains is to append the calculated number of ones and zeros in succession for the chosen number of cycles. With this done, the result is N channels consisting of square waves with varying duty cycles.

Generation of Full Digital Pattern

The final step is to apply a phase shift to the channels which need it, and to combine the channels into a single output file which can be read by the GP-24100. The phase shift for each channel is calculated based on the user's desired steering direction or focusing depth as described in the steering and focusing sections of Chapter 2, or taken from the manual phase shift input. Once a phase shift has been determined for each channel, the phase value is converted into a time delay, which is finally converted into a number of samples at the clock frequency using the following expression:

$$t_{shift} = \frac{phase}{2\pi frequency} \quad \textbf{(3.9)}$$

$$cycle_{shift} = round(\frac{t_{shift}}{clock_{period}}) \quad \textbf{(3.10)}$$

Hard coded phase compensation is taken into account by using (3.9) and (3.10) to again convert phase into samples and then adding the two results to get the total samples to shift. The final step is to move the start of each channels pattern the proper number of samples, effectively changing the channels phase relative to where it began. Then the shifted channels are combined into a matrix which is written to a text file.

GP-24100

The relevant capabilities of the GP-24100 for this project are its 8 megabytes of embedded memory, 16 bits of data output, 6 bits of control output, and adjustable clock frequency up to 100MHz. The embedded memory creates a limit on the length of the pattern that can be loaded onto the device. The number of cycles this limit entails depends on the chosen operating frequency of the array as well as the clock frequency of the GP-24100. The 16 data bits limit the maximum number of channels the device can control to 16, and likewise the 6 control bits limit the number of switches or settings in the analog portion of the signal path that can be modified using the GP-24100 output.

The maximum clock frequency of 100MHz results in a limit on the phase precision between channels. The phase precision that can be obtained for a desired operating frequency can be found by using the fact that the minimum time step that can be controlled with a 100MHz clock is:

$$T_{period} = {}^{1}/_{100MHz} \quad \textbf{(3.11)}$$

And the phase angle for a time delay is:

$$Phase = 360° * frequency * t_{delay} \quad \textbf{(3.12)}$$

As the frequency increases, the minimum phase step that can be commanded increases with it, reducing the precision of the array. See Figure 18 for a plot of minimum phase step compared to operating frequency.

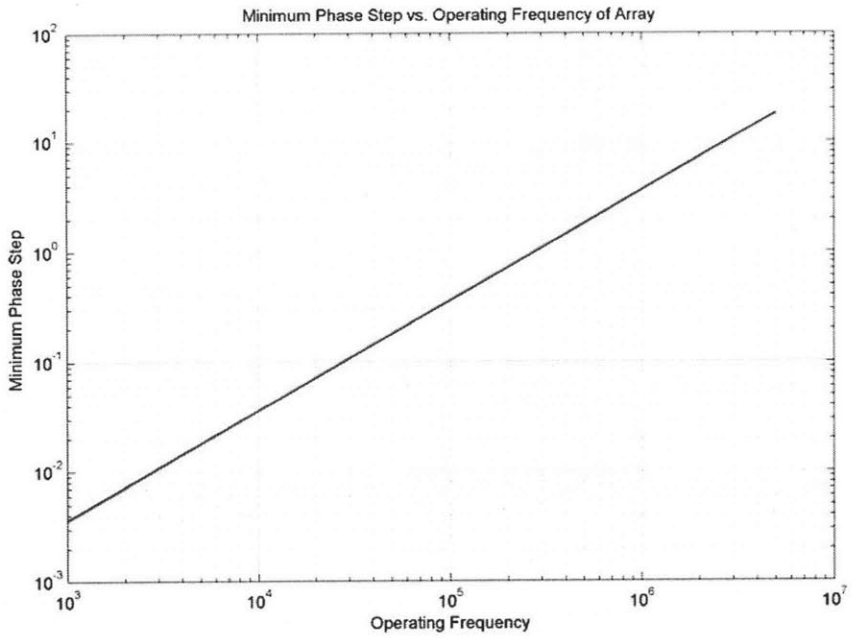


Figure 18: Minimum phase step as a function of operating frequency

Chapter 4

Hardware

Overview

The analog hardware portion of this phased array consists of three subsections. The first is the low pass filter IC responsible for eliminating all undesirable higher order harmonics from the digital square wave output by the GP-24100. The second subsection is responsible for amplifying the filter output and driving the transducers which make up the array. Finally, there are a number of power management circuits which generate the power supply voltages needed to run the filters and amplifiers.

Low Pass Filtering

The low pass filter chosen for this project is the LTC6603 from Linear Technologies. This chip provides two channels which can operate in fully differential or single ended mode. It provides a 9^{th} order low pass filter with a linear phase response along with an adjustable cut off frequency and gain. The schematic of the chip as utilized for this project is shown in Figure 19. What follows is a brief description of components and inputs which significantly alter the behavior of the IC as relevant to its use in this project. For a more detailed overview of the operation of this IC, see the datasheet provided by Linear Technology.

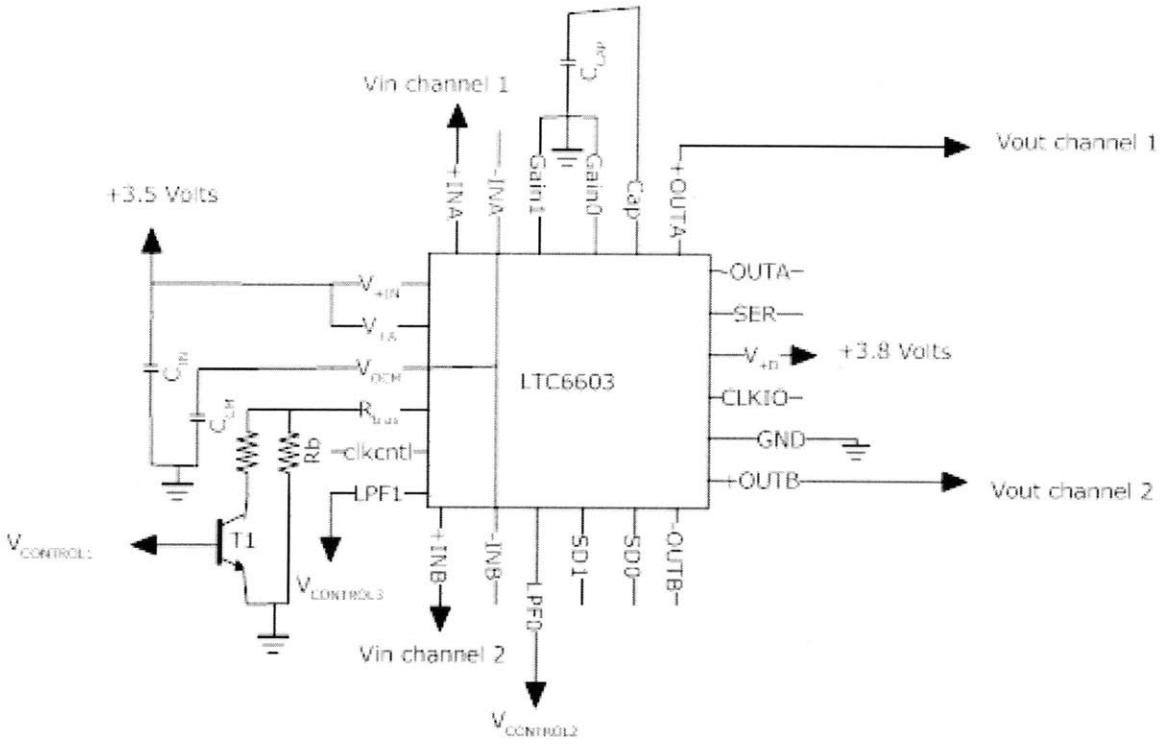


Figure 19: Schematic of low pass filter IC

Component Name	Value
Cin	.1uF
Ccm	1uF
Ccap	.1uF
Rb	100k Ohm
T1	2N2222a

Table 2: Component values for filter

The value of the capacitors in this circuit are dictated by the datasheet and do not determine the behavior of the filter. The component which largely affects the operation of the filter is the resistance attached to the R_{bias} pin. This resistance controls the frequency of the filter's internal clock according to the following relation:

$$f_{clk} = 247.2MHz * \frac{10k}{R_{bias}} \textbf{(4.1)}$$

The clock frequency is then related to the cutoff frequency of the low pass filter according to equation (4.2):

$$f_{cutoff} = \frac{f_{clk}}{512} \textbf{ (4.2)}$$

This holds true when the LPF0 and LPF1 pins are at a logic level low. If LPF0 is increased to a logic level high the cutoff frequency is increased by a factor of 4, and if LPF1 is increased to a logic level high the cutoff frequency is increased by a factor of 16. These effects are not cumulative, i.e. the largest active multiplier sets the cutoff frequency. The minimum high level voltage is 2 Volts and the maximum low level voltage is .8 Volts. Vcontrol2 and Vcontrol3 are provided by the control outputs of the GP-24100. In addition, Vcontrol1 is also supplied by the GP-24100 and provides a reduction of the resistance seen at the Rbias pin when set to high through the BJT switch, giving further control over the cutoff frequency.

Amplification

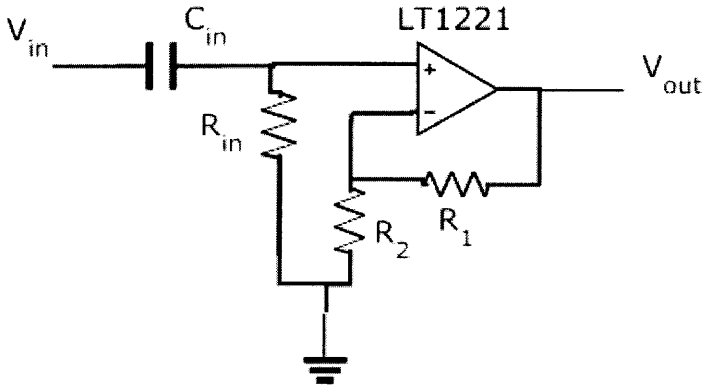


Figure 20: Schematic of transducer driving amplifier

Component Name	Value
Cin	.1uF
Rin	100k Ohms
R1	3.3k Ohms
R2	470 Ohms

Table 3: Component values for amplifier

A schematic for the amplifiers which drive the array elements is shown in **Error! Reference source not found.**. The circuit utilizes an operational amplifier (op-amp) configured as a non-inverting amplifier. The gain of this circuit depends on the values of R1 and R2 according to the following formula:

$$Gain = 1 + \frac{R_1}{R_2} \textbf{(4.3)}$$

The op-amp chosen for this circuit is the LT1221. It has a gain-bandwidth product of 150MHz. The gain in this configuration is 8 and this gain will therefore hold all the way up to 18MHz, well beyond the reasonable operating region of the digital portion of the array.

Power Circuitry

Three different supply voltages are required to power the filter and amplifiers. The amplifiers require +/- 14 Volts and the filter requires +3.5 Volts. The -14 Volts and +3.5 Volts are generated from an externally supplied +14 Volts.

The -14 Volts required to provide the amplifiers with a dual rail supply is generated using the LT1054 switched capacitor voltage converter.

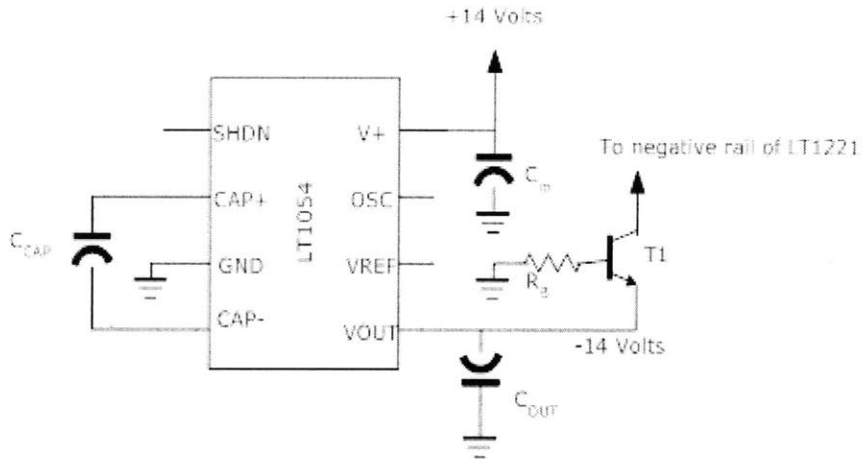


Figure 21: Schematic of voltage inverter

Component Name	Value
Cin	2.2uF
Cout	100uF
Ccap	100uF
Rb	10k Ohms
T1	2n222

Table 4: Component values for voltage inverter

This IC is capable of providing up to 100mA of output current. The BJT transistor at the output is required to prevent the voltage at the Vout pin from being pulled up to a value above ground. Since this circuit exclusively powers the operational amplifiers in the system and the supply current in the op-amps flows from the positive supply voltage to the negative supply voltage, it is very likely that Vout could be charged to a positive value if the transistor were removed from the circuit.

The +3.5 Volts required to power the LTC6603 is generated using the LT1763 low dropout linear regulator.

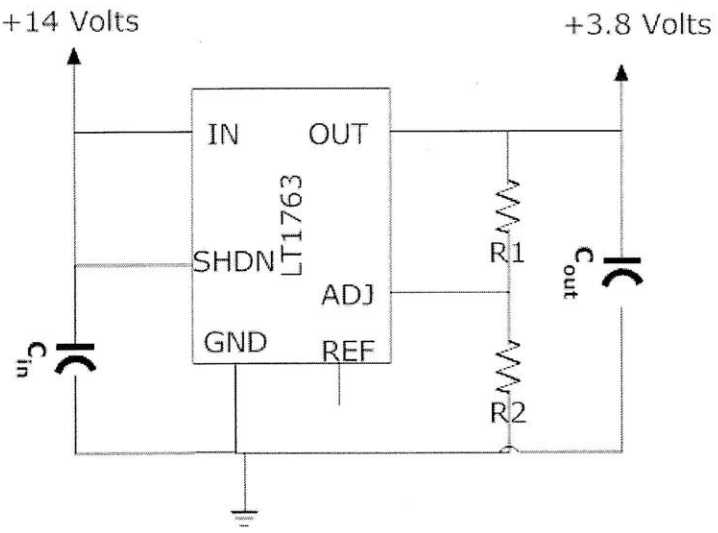


Figure 22: Schematic of filter power supply

Component Name	Value
Cin	10uF
Cout	10uF
R1	261k Ohms
R2	124k Ohms

Table 5: Component values for filter power supply

The output voltage of this circuit is controlled via feedback to the ADJ pin. The output voltage can be adjusted from 1.22 Volts to 20 Volts according to the following formula:

$$Vout = 1.22V * \left(1 + \frac{R1}{R2}\right) + I_{adj} * R1 \textbf{ (4.5)}$$

$$I_{adj} = 30nA \textbf{ (4.6)}$$

As indicated in the schematic, for the component value selected in this system Vout is equal to 3.8 Volts.

Chapter 5

Measurements

Test Setup

To examine the capabilities of the designed phased array, tests were run in an open air environment. Various steering and focusing profiles were chosen and a receiving piezoelectric transducer was used to measure the signal power at chosen spatial locations. In this way, a profile of the acoustic radiation pattern was determined for a given steering or focusing setting. In order to choose an operating frequency for the array in this environment, two frequency scans were conducted. The first scan measured the power at the receiver over a range of with no signal present in order to determine the noise background. Next, the scan was repeated with the array transmitting at max amplitude with all elements in phase at the frequency of interest in order to determine the signal level possible at that frequency. The two results can be seen in Figure 23.

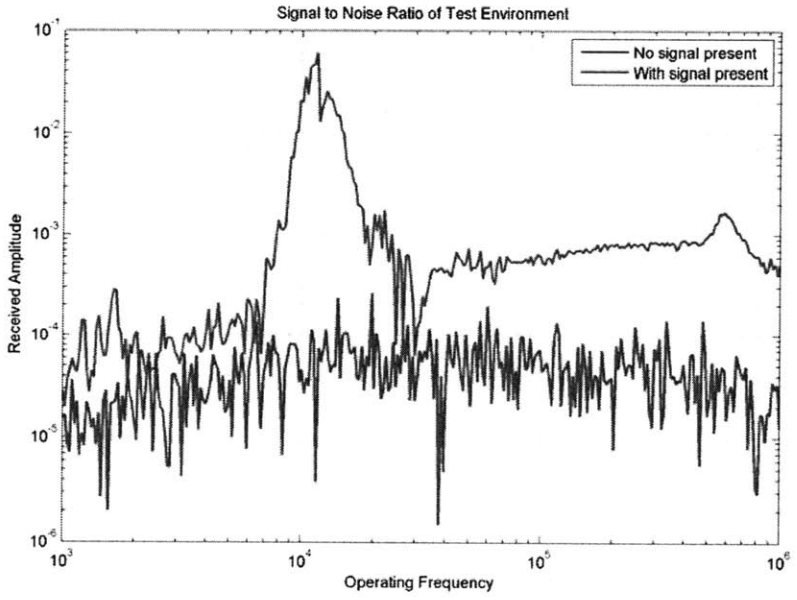


Figure 23: Signal and noise comparison of test environment

11 KHz, the frequency with the highest signal level to noise level ratio, was chosen as the operating frequency for the array in this environment. Note that at higher frequencies (above 50kHz) signal transmission is dominated by electromagnetic radiation rather than acoustic waves, making this test setup unsuitable for ultrasonic applications.

All of the following measurements were taken using an array of four elements of width 1.3cm spaced 3cm apart.

Beam steering results

The following figures depict the spatial radiation pattern at a distance of 38cm from the array. Data points were taken in the range from -22.5° to 22.5° with a step size of 2.5° between successive measurements.

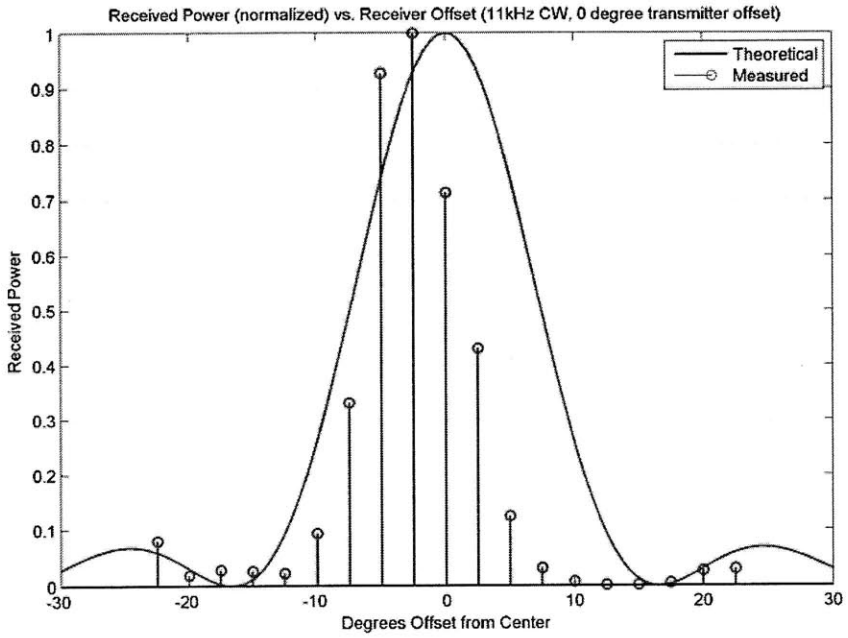


Figure 24: Measured spatial radiation pattern for a 4 element array

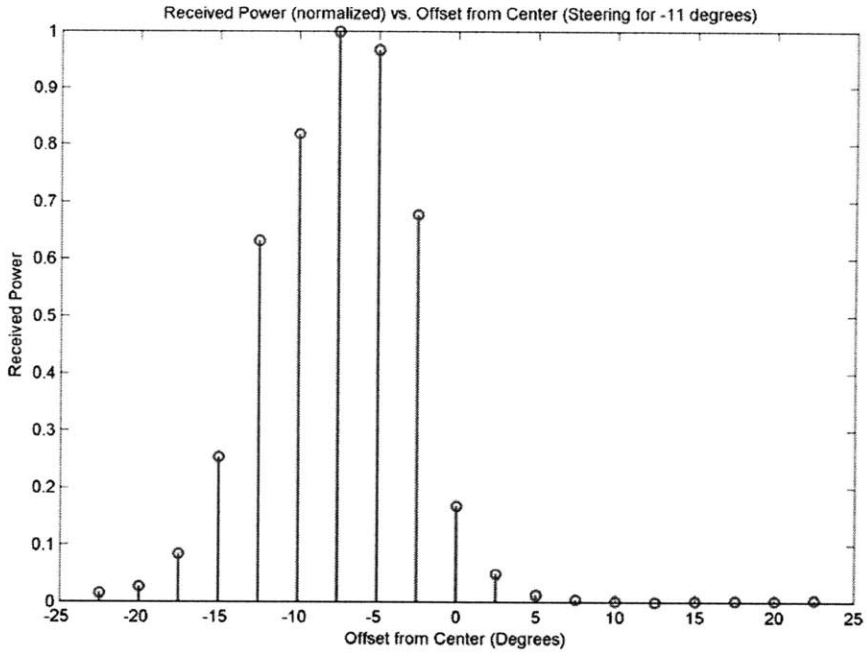


Figure 25: Measured spatial radiation pattern for a 4 element array steered for -11 degrees

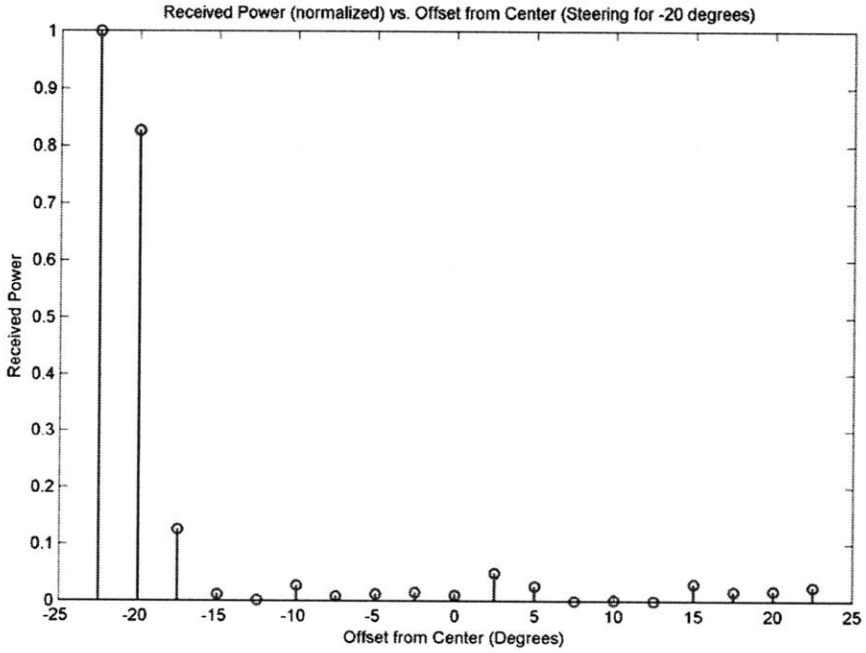


Figure 26: Measured spatial radiation pattern for a 4 element array steered for -20 degrees

These results show a clear ability to direct the radiation from the array in a preferred direction. However, there is still some imprecision in the results that could stem from a number of factors.

The first factor is that the steering calculations are done using a fixed speed of sound, when in actuality the speed of sounds varies with air temperature and humidity. A 6% variation in the speed of sound (which is the equivalent of changing the air temperature from 90° Fahrenheit to 0° Fahrenheit) changes the steered angle by just over 1° Even more significant is the spacing between elements. If the elements are actually 2.8cm apart rather than 3cm, the steered angle changes by just over 1.5°. The dependence of the steering on these factors, coupled with the difficulty in creating a perfectly regulated testing environment, help to explain the variation between the actual peaks and the desired location of the peak.

Beam focusing results

The following figures depict the received power at various distances from the array. Measurements were taken at distances in the range from 2cm to 25cm in front of the array, with a step size of 1cm between measurements.

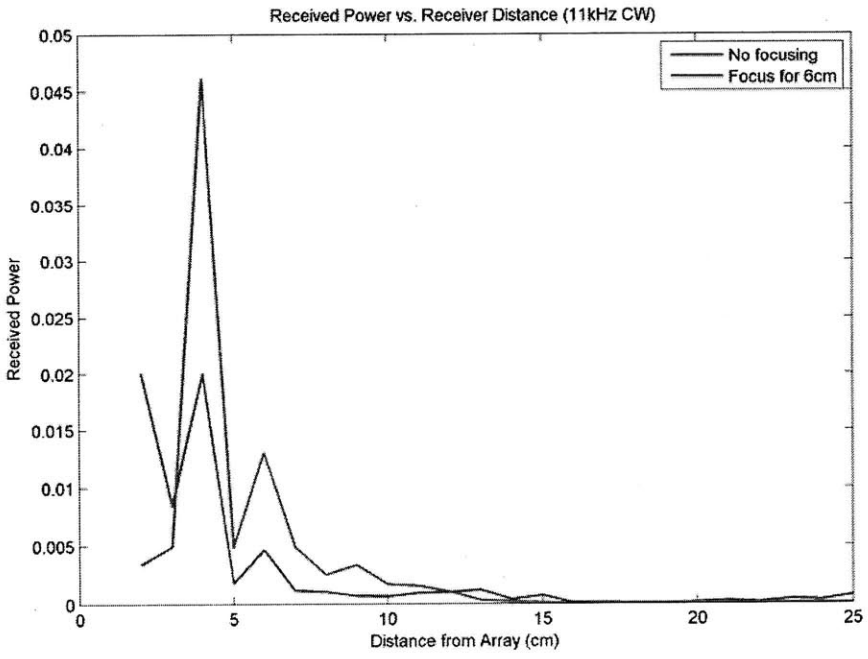


Figure 27: Received power vs. distance from array for focusing at 6cm

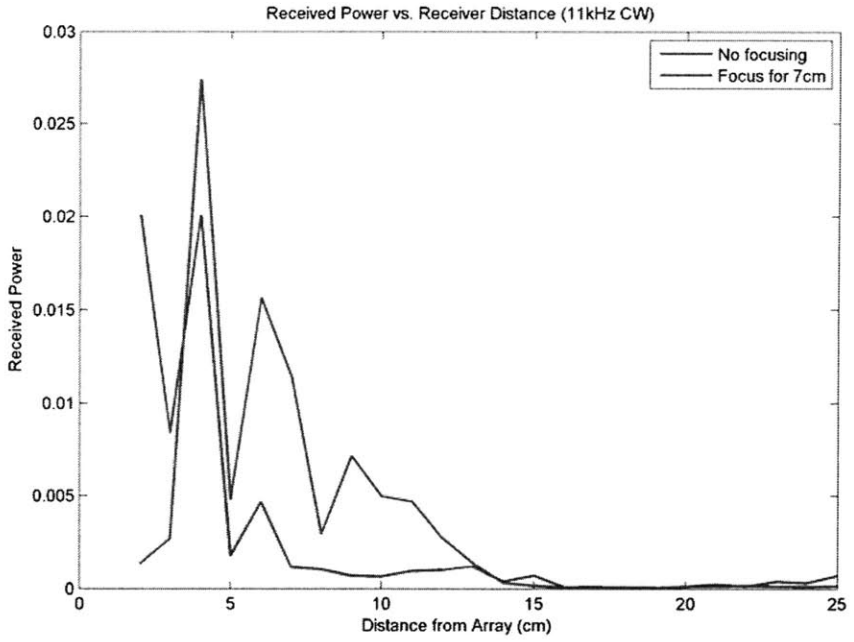


Figure 28: Received power vs. distance from array for focusing at 7cm

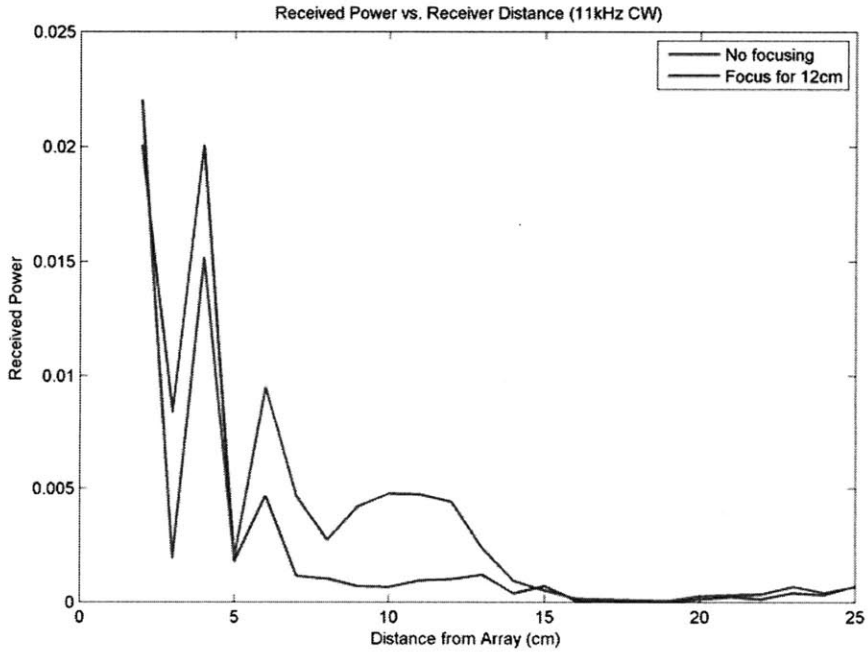


Figure 29: Received power vs. distance from array for focusing at 12cm

These results show a clear change in the intensity distribution from focusing attempts. However, they are subject to the same uncertainties as steering, namely variation in the speed of sound and

imprecision in the actual distance between elements. Furthermore, the results are affected by resonances created by adjusting the position of the receiver. When the receiver is very close to the array and is at a distance near a multiple of the wavelength, a resonant cavity is created; skewing the results from what would be the ideal case with a non-interfering receiver.

Chapter 6

Conclusion

This thesis presented a phased array system capable of arbitrarily controlling the phase of up to 16 transducers. The shifted waveforms required to direct the array are generated digitally on a personal computer using LabVIEW and MATLAB. These waveforms are then output using the GP-24100, resulting in shifted square waves. Finally, the square waves are filtered and amplified before being applied to the transducers.

Measurements were conducted on a 4 element array operating in air at 11 KHz to demonstrate both steering and focusing of the array. Results clearly show that manipulating the preferred direction and focal distance of acoustic radiation is possible. Inaccuracy is evident in the results and is likely due to uncertainty regarding the precise speed of sound and the exact pitch of the array.

Possible improvements to this system include a more carefully manufactured structure to support the transducers, providing exactness in pitch calculations. Furthermore, the measurement process could be automated to provide finer resolution in data acquisition. In software, it should be possible to increase the capabilities of the array to include such features as steering and focusing at the same time. Hardware improvements could include amplifiers with increased gain for higher signal power, as well as adjustable gain for additional compensation if required.

Bibliography

Ajoy K. Ghatak, K. Thyagarajan. Optical electronics. Cambridge: Press Syndicate of the University of Cambridge, 1989.

Bevan B. Baker, E. T. Copson. The Mathematical Theory of Huygens' Principle. New York: Chelsea Publishing Company, 1987.

Ghatak, Ajoy. Optics. New Delhi: Tata Mcgraw-Hill, 2005.

Goodman, Joseph W. Introduction to Fourier Optics. Greenwood Village: Roberts and Company Publishers, 2005.

S. Egusa, Z. Wang, N. Chocat, Z. M. Ruff, A. M. Stolyarov, D. Shemuly, F. Sorin, P. T. Rakich, J. D. Joannopoulos & Y. Fink. "Multimaterial piezoelectric fibres." Nature Materials (2010).

Walter Heywang, Karl Lubitz, Wolfram Wersing. Piezoelectricity: Evolution and Future of a Technology. Berlin: Springer, 2008.

Weisstein, Eric W. Fourier Series--Square Wave. 30 April 2010. 26 July 2010 <http://mathworld.wolfram.com/FourierSeriesSquareWave.html>.

Appendix A: MATLAB Code

```
%INPUTS FROM LABVIEW: path, steer, channels, freq, focal, cycles, Pm, p_off,
shift, dcycle, hard_dcycle, v_sound, set_phase, hard_phase, distance,
element_size

%define clock frequency, max of 100MHz
clock_freq = 100e6;
clock_period = 1 / clock_freq;

%define pattern as empty for now
pattern = [];

%nominal time on
Ton_nom = (1 / freq) *.5;
%nominal time off
Toff_nom = (1 / freq) * .5;

%number of samples to be on and off
samples_on_nom = round(Ton_nom/clock_period);
samples_off_nom = round(Toff_nom/clock_period);
%length of basic pattern
basic_length = samples_on_nom + samples_off_nom;


%generate duty cycle for each channel
%check if manual duty cycle is enable
if (dcycle > 0)
    %if yes, set duty cycles from input
    D(1:channels) = hard_dcycle;
else
    %otherwise look at lowest measured power and attempt to set all
    %channels to that power
    Pg = min(Pm);
    D = (1/(2*pi)) .* acos(1 - (.2026423673*pi^2 * Pg ./ Pm));

    D = real(D);
end


%generate all channels
for t=1:channels

    basic_pattern = [];
    %time while on for a given duty cycle
    Ton = (1 / freq) * D(t);
    %time while off for a given duty cycle
    Toff = (1 / freq) * (1-D(t));
    %number of samples to be on or off
    samples_on = round(Ton/clock_period);
    samples_off = round(Toff/clock_period);

    %check for rounding errors and pad samples off to keep all channels the
    %same length
```

```
    while( (samples_on + samples_off) < basic_length + 1)
        samples_off = samples_off + 1;
    end

    %generate pattern given calculated samples on and off per cycle
    %create seperate portions of the cycle (on and off)
    pattern_on = ones(1, samples_on);
    pattern_off = zeros(1, samples_off)

    for i=1:(2*cycles)
        if(mod(i,2) == 1)
            basic_pattern = [basic_pattern pattern_on];
        else
            basic_pattern = [basic_pattern pattern_off];
        end
    end

    %shift current channel by amount equal to half the difference between
samples_on and samples_on_nom (or samples_off and samples_off_nom)
    %this is to attempt to leave the phase unchanged after duty cylce
modification

    if(samples_on < samples_on_nom)
        dshift = (samples_on_nom - samples_on) / 2;
        shifty = basic_pattern(basic_length - abs(dshift):end);
        basic_pattern = [shifty basic_pattern];
        basic_pattern = basic_pattern(1:basic_length);
    end

    if(samples_off < samples_off_nom)
        dshift = (samples_off_nom - samples_off) / 2;
        shifty = basic_pattern(basic_length - abs(dshift):end);
        basic_pattern = [shifty basic_pattern];
        basic_pattern = basic_pattern(1:basic_length);
    end

    %append newly generated channel to total pattern
    pattern = [basic_pattern' pattern];

end

%generate phase shift for each channel
%start by converting desired steering direction to radians
shift = pi * shift / 180;

%wavelength of sound
lambda = v_sound/freq;


%choose either steering or focusing for now

if (steer == 1)
    %phase shift calculation for each channel (relative to channel 1)
    for l=1:(channels - 1)
```

```
        %copy current channel to pattern_1
        pattern_1 = pattern(:,(channels - 1))';

        %check for manually set phase offset
        if(hard_phase > 0)
            %convert hard phase offset to radians and shift according to
current channel
            p = rem( ((set_phase*1)*(pi / 180)), 2*pi);
        else
            %calculate phase to shift current channel in radians given a
            %direction to shift to and distance between elements
            p = rem( (1*2*pi*distance*sin(shift) / lambda), 2*pi);
        end

        %equivalent time delay for calculated phase shift
        ts = p / (2*pi*freq);

        %clock cycles to shift for given time delay
        cshift = round( (ts) / clock_period);

        %apply phase compensation
        %clock cycles to shift for given phase correction
        cc = round (  ((p_off(1) * ( pi / 180) * (1 / (2 * pi * freq))) /
clock_period)  );

        cshift = cc + cshift;

        %copy amount to shift from end of pattern, move to start, resize
pattern to
        %original length
        if(cshift > 0)
            shifty = pattern_1((length(basic_pattern) - abs(cshift)):end);
            pattern_1 = [shifty pattern_1];
            pattern_1 = pattern_1(1:length(basic_pattern));
        end

        %OR copy amount to shift from start of pattern, move to end, resize
        %pattern to original length
        if(cshift < 0)
            shifty = pattern_1(1:abs(cshift));
            pattern_1 = [pattern_1 shifty];
            pattern_1 = pattern_1((length(shifty) + 1):end);
        end


        %replace current channel with shifted channel
        pattern(:,(channels-1)) = pattern_1';

    end

else
    %LENS FOCUS BASED PHASE CALCULATION
```

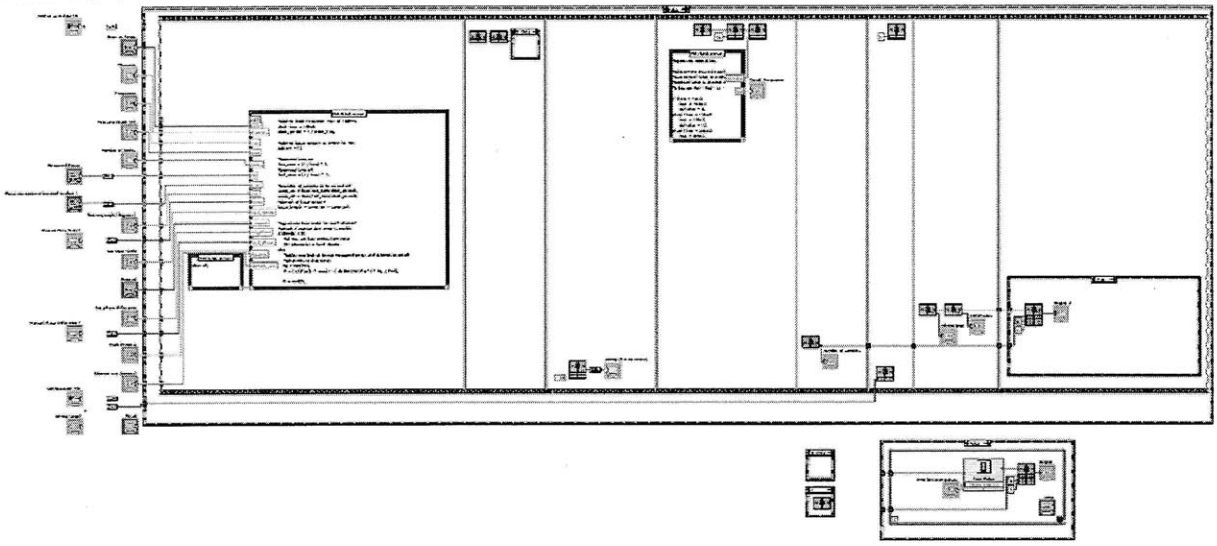
```
    %calculate wavenumber
    k = (2*pi)/lambda;

    %calculate centerpoints of each element in the array
    %total array size
    array_width = distance*(channels - 1) + element_size;

    %space centerpoints properly
    for n = 1:channels
        centerpoints(n) = element_size/2 + distance*(n-1);
    end

    %shift centerpoints to be centered at zero
    centerpoints = centerpoints - array_width/2;

    %phase shift for each calculated center point and given focal length
    phasef = angle(exp(-j*(k/(2*focal)).*centerpoints.^2));

    timef = phasef / (2*pi*freq);

    %clock cycles to shift for given time delay
    cshift = round( (timef) / clock_period);

    %shift each channel the proper amount
    for l=1:channels
        %copy current channel to pattern_l
        pattern_l = pattern(:,(channels + 1 - l))';

        if(cshift(l) > 0)
            shifty = pattern_l((length(basic_pattern) - abs(cshift(l))):end);
            pattern_l = [shifty pattern_l];
            pattern_l = pattern_l(1:length(basic_pattern));
        end

        %OR copy amount to shift from start of pattern, move to end, resize
        %pattern to original length
        if(cshift(l) < 0)
            shifty = pattern_l(1:abs(cshift(l)));
            pattern_l = [pattern_l shifty];
            pattern_l = pattern_l((length(shifty) + 1):end);
        end

        %replace current channel with shifted channel
        pattern_l = pattern_l';
        pattern(:,(channels + 1 -l)) = pattern_l;
    end

end
for y=channels:15

    pattern_fill = zeros(length(basic_pattern), 1);
    pattern = [pattern_fill pattern];

end
```

```
dlmwrite(path, pattern, 'delimiter', '', 'newline', 'pc');
```

Appendix B: LabVIEW Block Diagram

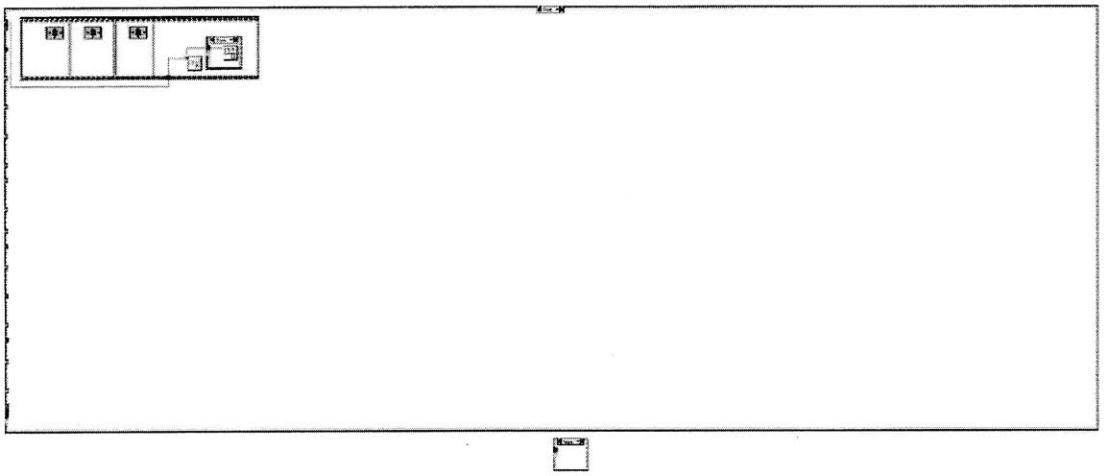
Page 2

pattern_generate.vi
C:\Users\drodgers\Documents\meng\LabView\pattern_generate.vi
Last modified on 7/26/2010 at 1:05 PM
Printed on 7/28/2010 at 2:01 PM



Page 3

pattern_generate.vi
C:\Users\drodgers\Documents\meng\LabView\pattern_generate.vi
Last modified on 7/26/2010 at 1:05 PM
Printed on 7/28/2010 at 2:01 PM



Time Delay
Time Delay
Inserts a time delay into the calling VI.

This Express VI is configured as follows:

Delay Time: 0.003 s