BME 386 Final Project - Designing an Image Detection System Using Ultrasound and Piezoelectric Ceramics

Name	*Student ID*
Hannah Cardoza	20831944
Adam Courchesne	20824794
Megan D'Silva	20828425
Kayla Jetha	20822795
Sophie Stupalo	20827311

December 2^{nd}, 2022

Abstract

The project design consisted of 4 key modules: a detecting module, a transducer pulse module, an imaging module, and an analog processing module. Two detecting module designs were considered: (1) using the HRS04 Arduino ultrasonic sensor as both the transducer and receiver and (2) using a separate piezoelectric ceramic disc as the transducer and the HRS04 Arduino ultrasonic sensor as the receiver. Two transducer pulse modules were considered: (1) using an Arduino Uno micro-controller and (2) using a function generator. Two imaging module designs were considered: (1) using a 3D printed auto-turret setup with rotational motion driven by stepper motors and (2) using a 3D printer to achieve an imaging pattern driven by translational motion. The analog processing module was achieved using Arduino code which printed the distance to the next obstruction in the serial monitor alongside a python script that processed the data to produce a 2D-pixel plot. Through system characterization exercises, the final project design was chosen. The final design used the HRS04 Arduino ultrasonic sensor as both the transducer and receiver for the detecting module, an Arduino Uno micro-controller as the transducer pulse module, a 3D printer as the imaging module, and Arduino code and a python script as the analog processing module. This design successfully scanned 3 unique images.

Contents

1 Introduction

Applications of ultrasound technology are common among many industries, especially within the medical industry. Within the medical field, ultrasound's most early and common use is external fetal scanning [1]. Over time, innovations and developments in ultrasound scanning technologies have also benefited other anatomical sites [1]. For example, within the dermatological and ophthalmic spaces, probes which are inserted into the body can be used to allow for scanning with higher spatial resolutions [1]. Within the industry, ultrasound applications are also being used for non-imaging purposes, such as investigating bone health and osteoporosis [1]. Ultrasound applications are also being used within physiotherapy treatments since sound waves can be leveraged to penetrate soft tissues [2]. This noninvasive form of treatment is used to relieve pain and improve mobility by improving blood flow and circulation to promote the healing of tissues [2]. Within clinical imaging and therapy, biomedical engineers are responsible for the design and development of ultrasound machines which are used by healthcare professionals. Therefore, it is imperative to train biomedical engineers on the mechanisms, modules and basic components which comprise ultrasonic scanners.

Ultrasound can be simply explained as a sound wave which transports mechanical energy at high frequencies to determine distances to the next physical obstruction [1]. When considering medical imaging applications of ultrasound, handheld transducers are used to send beams of ultrasound waves [3]. These waves are generated by piezoelectric crystals which are a type of ceramic material [3]. When an electric field is applied to the crystals, they generate a sound wave at a frequency between 2 and 18 MHz [3]. As the sound waves hit different tissues and liquids, they bounce back to a receiver, which is also comprised of piezoelectric crystals. The amount of time taken to receive the sound wave sent by the transducer can be leveraged to determine how far an object is and formulate images of tissues and organs [3].

To create an ultrasound machine, there are 4 important modules to consider: a detecting module, a transducer pulse module, an imaging module, and an analog processing module [4]. The detecting module is responsible for obtaining information from the outside system [4]. This module consists of the ultrasound generator and receiver, where the generator sends the sound wave and the receiver receives the reflected sound wave that has bounced back after hitting an object [4]. The transducer pulse module is responsible for providing the periodic pulses to be sent by the ultrasound generator [4]. The pulse is typically generated by a microcontroller and transmitted by a piezoelectric material [4]. The imaging module details the scanning methodology and more specifically, the pathway of movement to be completed by the ultrasound-detecting module to scan one pixel at a time [4]. The imaging module is useful for providing an array of scanned pixels or values to a processing module which forms the image. The analog processing module is responsible for converting the analog signal recorded by the receiver to a digital output which can be interpreted by the user [4].

2 Experimental Design

2.1 Design Overview

The project design consisted of 4 key modules: a detecting module, a transducer pulse module, an imaging module, and an analog processing module. The detecting module consisted of a transducer and a receiver. While designing the detecting module we explored two different options: (1) using the HRS04 Arduino ultrasonic sensor as both the transducer and receiver and (2) using a separate piezoelectric ceramic disc as the transducer and the HRS04 Arduino ultrasonic sensor as the receiver. The Arduino ultrasonic sensor operates at a different frequency than the piezoelectric ceramic, so alternative transducer pulse modules were needed. The first consisted of an Arduino Uno microcontroller and the accompanying Arduino code, whereas the second was simply a function generator. Two different imaging module designs were considered. The first used a 3D printed auto-turret setup with rotational motion driven by stepper motors. The second repurposed a 3D printer to achieve an imaging pattern driven by translational motion. Finally, the analog processing module was achieved using Arduino code which printed the distance to the next obstruction in the serial monitor alongside a python script that processed the data to produce a 2D-pixel plot.

2.2 Detecting Module

2.2.1 Alternative 1: HC-SR04 Ultrasonic Distance Sensor

As mentioned previously, the detecting module is responsible for sending and receiving the ultrasounds signal. The first design achieved this using the HC-SR04 ultrasonic distance sensor. This module calculated the distance to a target object by measuring the time taken to send and receive a sound wave [5]. The ultrasound transducer, identified by the `trig` pin, triggered the ultrasonic sound pulses and emitted an eight-cycle burst of high-frequency sound waves (40kHz) toward the target object [6]. When the sound wave detected the object, upon contact, the sound wave was reflected and received by the ultrasound receiver, identified by the `echo` pin [7]. The `trig` and `echo` pins were integrated in the ultrasonic sensor, along with a V_{CC} and GND pin. The V_{CC} pin was supplied by the +5V supply voltage pin on the Arduino Uno. The GND pin was grounded by the GND pin on the Arduino Uno. Both the `trig` and `echo` pins were connected to Digital I/O pins on the Arduino board. Figure 1 includes the circuit schematics of this module design.

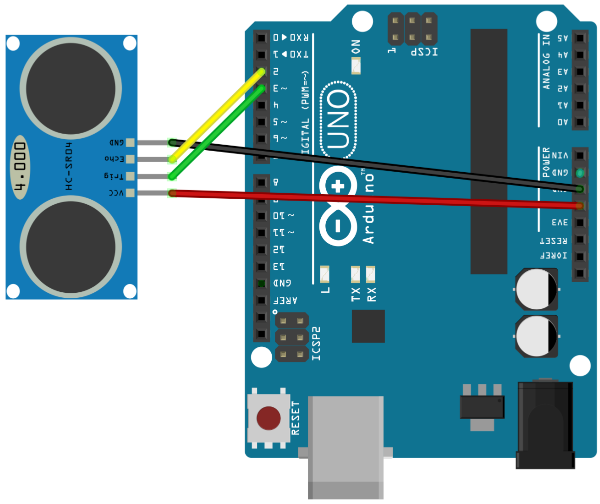


Figure 1: Schematic of the Ultrasonic Sensor Detecting Module Design.

2.2.2 Alternative 2 (Bonus): HC-SR04 Sensor and Piezoelectric Ceramic Disc

As mentioned previously, the self-built detecting module was achieved using a piezoelectric ceramic component and the HC-SR04 ultrasonic distance sensor. The design was first implemented with the `trig` pin unconnected to the Arduino Uno, however, it was found that the ultrasonic sensor requires the circuit to be complete to receive a signal. This meant that to record a signal sent using the piezoelectric ceramic, the `trig` pin needed to remain functional. The transmitter on the ultrasonic sensor was then covered with a bottle cap and replaced by the piezoelectric ceramic. This can be seen in Figure 2 alongside the accompanying transducer pulse module.

2.3 Transducer Pulse Module

2.3.1 Alternative 1: Arduino Uno Digital Output Code

The transducer pulse module is responsible for sending the periodic pulse necessary to produce an the ultrasound signal. This was implemented using Arduino code which sent a signal through the Arduino Uno digital pins at a controlled frequency. The HC-SR04 sensor data-sheet states that the minimum required input signal to produce an ultrasound wave should have a period of 10 microseconds [8]. However, any frequency above this threshold is able to produce a signal at the working frequency of 40kHz. For this reason, the transducer pulse module for Design 1 was implemented with the secondary function of controlling the frequency at which distance samples were taken. The final signal sent using the transducer pulse module had a period of 17 milliseconds with a roughly 12% duty cycle.

2.3.2 Alternative 2 (Bonus): Function Generator

The piezoelectric ceramics that were purchased for this project operated at a resonant frequency of 3.6 kHz. However, the transducer on the ultrasonic sensor has a working frequency of 40 kHz. For this reason, a function generator was necessary to induce a higher frequency wave from the piezoelectric disc that fell within the bandwidth of the receiver. This was done by sending a 20 V peak-to-peak sine wave with a frequency of 16 kHz. After the ultrasound wave was transmitted by the piezoelectric material, it successfully reflected off of the target object and was identified by the `echo` pin [7]. Figure 2 includes the circuit schematics of this module design alongside the detecting module described in Section 2.2.2.

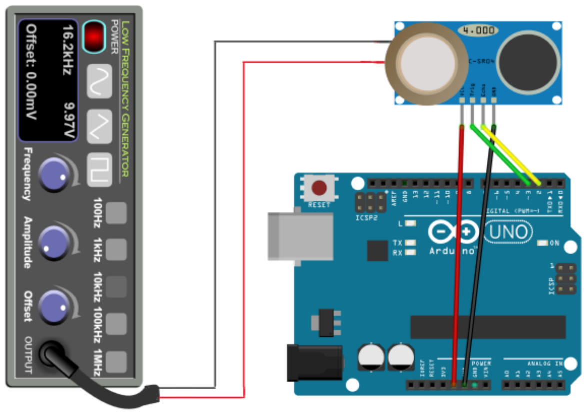


Figure 2: Schematic of the Piezoelectric and Ultrasonic Sensor Transducer Pulse Module Design.

2.4 Imaging Module

2.4.1 Alternative 1: Auto-Turret

To generate an image of the target object, this design used a single-pixel imaging technique following an expanding spiral motion which is shown in Figure 3 below. This technique allowed image acquisition to be provided by a device only equipped with a single-point detector [9]. In this case, the single-point detector corresponds to the detecting module described earlier in Section 2.2.1.

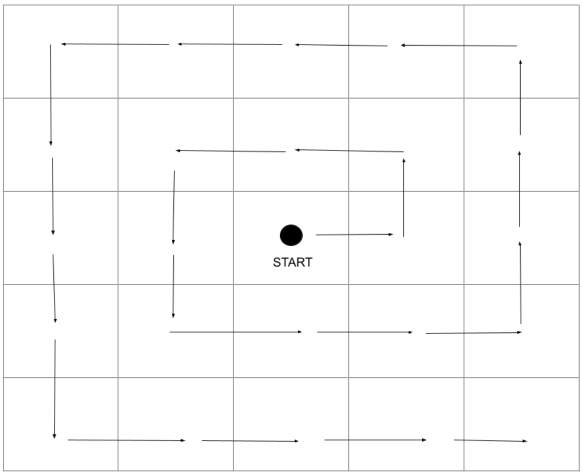
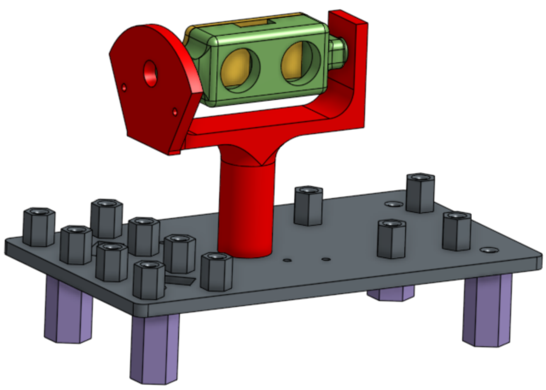
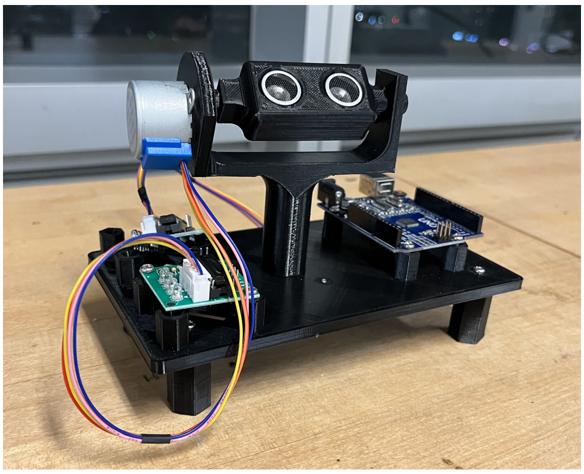


Figure 3: Turret Scanning - Expanding Spiral Pattern.

This single-pixel scanning method was implemented using an "auto-turret" design consisting of two 28BYJ-48 stepper motors and 3D printed custom components that held the Arduino Uno, motor drivers, ultrasonic sensor, and stepper motors. Figure 4 (a) shows the 3D CAD rendering built using OnShape, and Figure 4 (b) shows the assembled fixture. The fixture was designed with a base to hold the Arduino Uno and its respective drivers, and an "auto-turret" part to hold the ultrasonic sensor and stepper motors. The stepper motor fixed on the side of the ultrasonic sensor was used to control vertical movements, and the motor fixed underneath the base was used to control horizontal movements. Arduino code was written to step the motors in the necessary order to achieve the single pixel scanning method shown in Figure 3.



(a)



(b)

Figure 4: 3D Components of Design 1 shown as (a) a 3D CAD rendering built using Onshape and (b) an assembled fixture, including the Arduino Uno, motor drivers, ultrasonic sensor, and stepper motors.

The schematic that was used to implement this design can be seen in Figure 5.

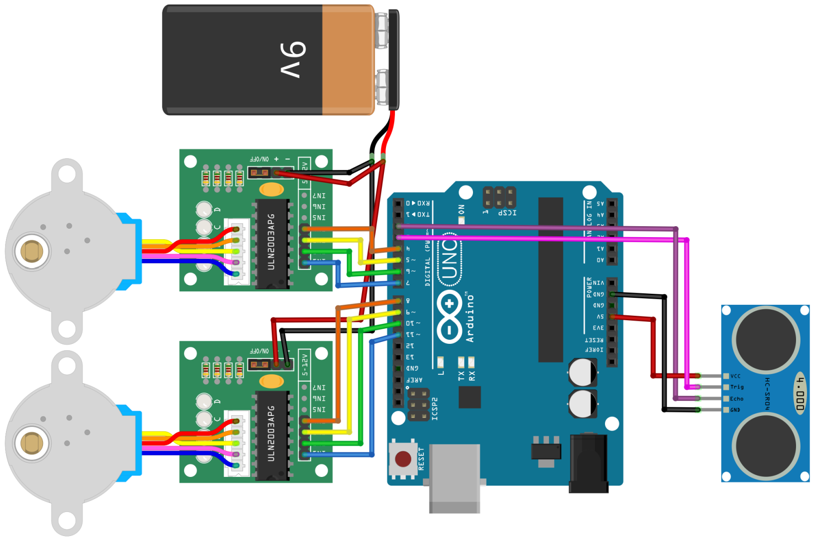


Figure 5: Schematic of the Auto-Turret Design.

2.4.2 Alternative 2: 3D Printer Gantry System

To generate an image from the target object, this design used a single-pixel imaging technique following a translational scanning pattern shown in Figure 6 below. In this case, the single-point detector also corresponded to the detecting module described earlier in Section 2.2.1.

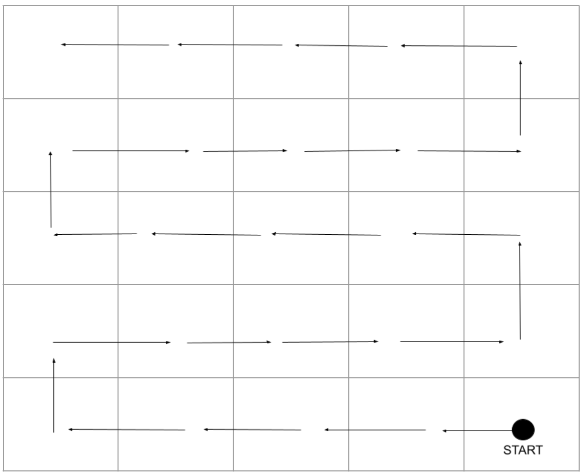
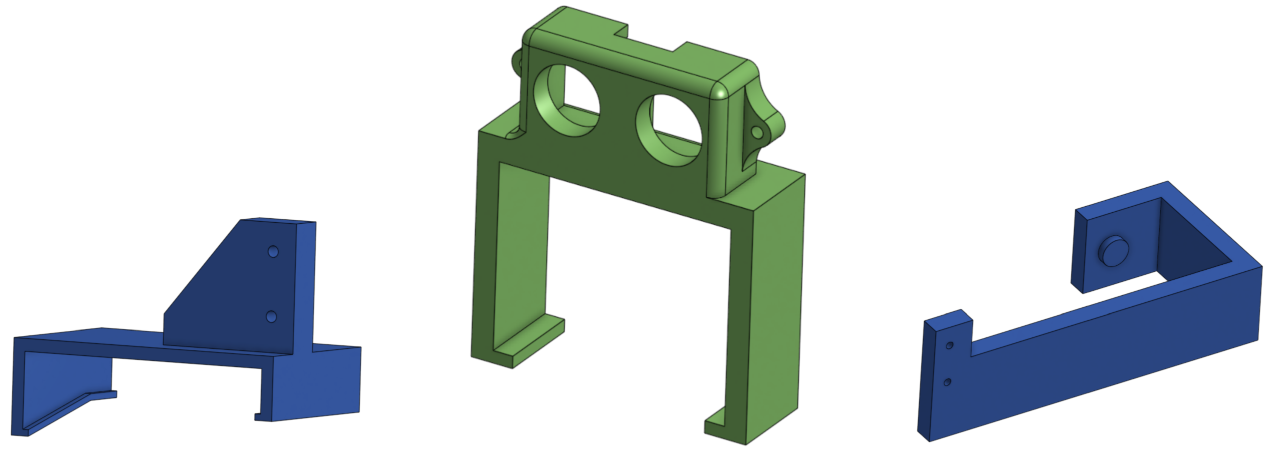


Figure 6: Translational Scanning Pattern.

Rather than purchasing and implementing additional components, such as linear actuators or gear systems, the 2D gantry system in the Creality Ender 3 3D printer that was used to produce the components described above was adapted to facilitate the chosen single-pixel scanning method. Microswitches placed at each end of the 3D printer's x-axis were used to notify the Arduino when the ultrasonic sensor finished scanning a complete row of pixels.

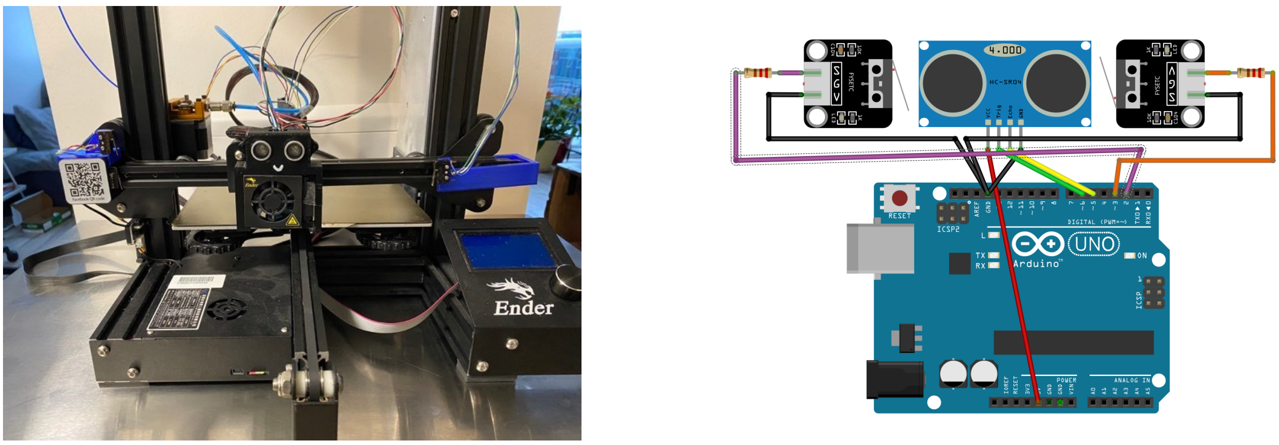
The 3D printing process was controlled using a set of instructions written in a computer numerical control (CNC) programming language called G-code [10]. A custom G-code script was written which disabled printer components like the extruder and heated bed, and moved the hot end along the desired path at the fastest possible speed.

To assemble the imaging module, CAD fixtures designed using OnShape were used to fasten the ultrasonic sensor and microswitches to the 3D printer. Figure 7 (a), (b), and (c) show the CAD renderings of the left microswitch fixture, ultrasonic sensor fixture, and right microswitch fixture, respectively. Figure 8 shows the completed (a) imaging module assembly, and (b) circuit schematic. The ultrasonic sensor fixture was designed to fit securely around on the 3D printer's hot end. Two holes were made for the transmitting and receiving transducers. The left and right microswitch fixtures were uniquely designed to mount on the left and right sides of the 3D printer's x-axis.



(a) (b) (c)

Figure 7: 3D CAD Renderings of Mounts for the (a) Left Microswitch, (b) Ultrasonic Sensor, and (c) Right Microswitch.



(a) Physical Components (b) Schematic

Figure 8: Completed Imaging Module Assembly.

2.5 Analog Processing Module

The analog processing module is used to process the ultrasound signals and convert them to a digital output for the user to read [4]. Using the `pulseIn()` function, the echo pin read the travel time and calculated the distance, which was shown on the output console [11]. By setting the echo pin to HIGH, the `pulseIn()` function waited for the sound wave to reflect off of the object, depicted when the pin was HIGH, and started timing until the wave was received by the receiver, depicted when the pin was LOW [11]. The `pulseIn()` function returned the time of the wave signal, which was multiplied by 0.034, representing the speed of sound, and divided by 2, since the sound travels there and back, to obtain the distance measurement [11]. After each sample was taken using the scanning patterns described in Section 2.4, each recorded distance value was then mapped to a greyscale value from 0 to 255, 0 being black and 255 being white. Note that any value above 18cm was mapped to 255 to negate any background noise. The greater the distance of the pixel, the lighter the colour assigned to the pixel. Thus, closer obstructions were assigned lower greyscale values (dark grey and black), and further obstructions were assigned higher greyscale values (light grey and white). Upon notification, the Arduino would execute a `Serial.println()` command in the serial monitor. This sequentially printed an array of greyscale values for each row. This process is shown in just a few lines of code below.

```
digitalWrite(trigPin, LOW);
delay(15);
digitalWrite(trigPin, HIGH);
delay(2);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
grayScaleValue = map(distance, 2, 18, 0, 255);
if( distance > 18) {
  grayScaleValue = 255;
} else if( distance < 2 ) {
  grayScaleValue = 0;
}
Serial.print(grayScaleValue);
Serial.print(" ");
```

Once a row of greyscale values was printed in the Arduino serial monitor, a Python script was used to read those values, process the data, and produce an image. The data was read using the `PySerial` package. After each row was imported, the string of text was processed. Processing this data included removing unexpected characters added to the string during the import process and separating resulting greyscale values into an array of floats. Since the scanning direction alternated for each row, the resulting array needed to be inverted for every other row. The resulting array of numbers was then appended to a greater 2D array. The Python `Numpy` and `MatPlotLib` were then used to convert the image into a 2D-pixel plot. Specifically, the `MatPlotLib imShow()` function was used to produce the pixel plot out of the 2D array which was cast to fit the `Numpy` float data type as required by `imShow()`. Note that the `imShow()` function automatically incorporates Nearest Neighbour Interpolation to produce a smoother image.

2.6 Conclusion

Two design alternatives were produced for the detecting module, transducer pulse module, and imaging module. Only one design was made for the analog processing module. For the detecting module, the HC-SRO4 setup described in Section 2.2.1 was implemented as part of the final design. Since the piezoelectric ceramic was not used, the function generator was no longer necessary so the transducer pulse module described in Section 2.3.1 was adopted. Finally, the imaging module which recruited the 3D printer gantry system as described in Section 2.4.1 was also implemented. Justification for the use of these components will be addressed in the next section.

2.7 Bonus

Although the designs mentioned above which recruited the function generator and piezoelectric ceramic were not used in the final design, they were still implemented to satisfy the requirements for the bonus. The final circuit was shown above in Figure 2. Note that there was no imaging module used for this design since it was only necessary to measure distances using the piezoelectric component as a transducer and the ultrasonic receiver integrated in the ultrasonic sensor rather than create an image. If this detecting module configuration was used in the scanning machine, the module would be assembled with the 3D Printer similar to what is mentioned in Section 2.4.1, although the piezoelectric material would be connected to a function generator.

Here is a link to the video that demonstrates it working: `https://youtu.be/349mDD896vE`

3 System Characterization

3.1 HC-SR04 Ultrasonic Sensor vs Piezoelectric Ceramics (Bonus)

In order to determine the best detection module, the ultrasonic sensor transmitter was compared against the piezoelectric ceramic transmitter. The HC-SRO4 ultrasonic sensor was tested using its built-in transmitter and receiver, to ensure the correct distances can be accurately measured. Next, the piezoelectric ceramic was tested by using the piezoelectric ceramic as the transmitter and the ultrasonic sensor's built-in receiver as the receiver. The piezoelectric ceramic was placed in front of the covered ultrasonic sensor's built-in transmitter, to test if the correct distances could be accurately detected after a 20V peak-to-peak 16 kHz sine wave was applied. These modules detection limits were evaluated to determine the module that should be used in the final design.

3.1.0.1 Detection Limit Testing

Various rounds of testing were performed to evaluate the functionality and precision of the ultrasonic and piezoelectric ceramic detection modules. Two tests were performed in order to characterize the systems, determine their detection limits, and their optimal operating conditions to establish the most accurate module. The first test identified the optimal longitudinal range the system could best detect the distance to the target object. The second test established the lateral range of placement so that the system could detect that a target object was present. These parameters were evaluated by comparing the distance to the next obstruction that was printed in the Serial Monitor to the distance the target object was placed at.

Longitudinal Limit Testing

Both detecting modules were tested to determine the farthest distance the system could reliably detect a 25.5 cm by 7.5 cm object. The object was placed at distances which ranged from 5 cm to 100 cm away from the detecting module for the ultrasonic sensor and 5 cm to 30 cm for the piezoelectric ceramic. These distances were compared to the detected distances shown in the Serial Monitor. The setup and results are shown in Figure 9 and Table 1.

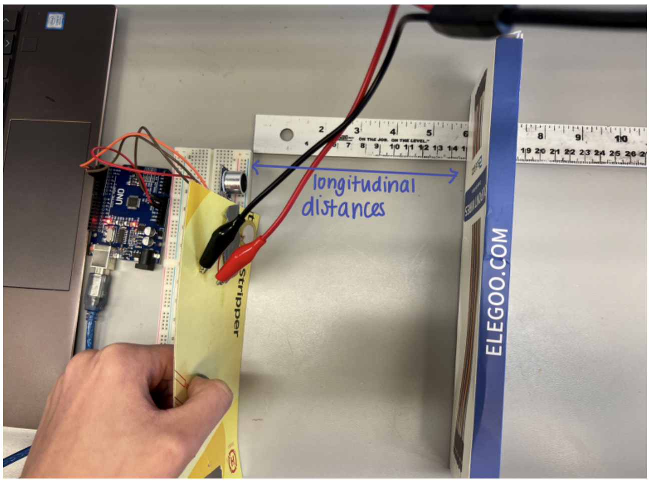


Figure 9: Longitudinal distance limit testing of ultrasonic sensor with piezoelectric ceramic component

Table 1: Longitudinal Limit Results for the Ultrasonic and Piezoelectric Detecting Modules

Piezoelectric Ceramic		**HC-SR04 Ultrasonic Sensor**	
Object Distance (cm)	**Reported Distance (cm)**	**Object Distance (cm)**	**Reported Distance (cm)**
5	7	5	4
7.5	9	10	10
10	11	20	19
12.5	15	30	30
15	17	40	39
17.5	20	50	49
20	24	60	58
22.5	26	70	67
25	29	80	76
27.5	34	90	81
30	38	100	81

Through the longitudinal detection limit testing it was established that the piezoelectric ceramic detecting module sensed the distance to the target object most accurately when the object was placed 7.5 cm to 12.5 cm away. The distances reported gradually worsened in precision when the object was placed more than 25 cm away from the detecting module. Meanwhile, for the ultrasonic sensor detecting module, the reported distance was the most accurate when the object was placed at a distance ranging from 5 cm to 60 cm. Likewise, as the object distance increased, the reported distances gradually worsened in precision.

Lateral Limit Testing

Based on the results from the longitudinal testing, it was determined that the most accurate distance for both modules was at a longitudinal distance of 10 cm. Therefore, a constant longitudinal distance of 10 cm was controlled, while a range of lateral distances were tested. The 7.5 cm by 22.5 cm object was placed at distances ranging from ±10 cm from the centre of each detection module, and were compared to the detected distance shown in the Serial Monitor. This aided in understanding at what off-centre distance could the detecting module detect a target object was present. The setup and results are shown in Figure 10 and Table 2.

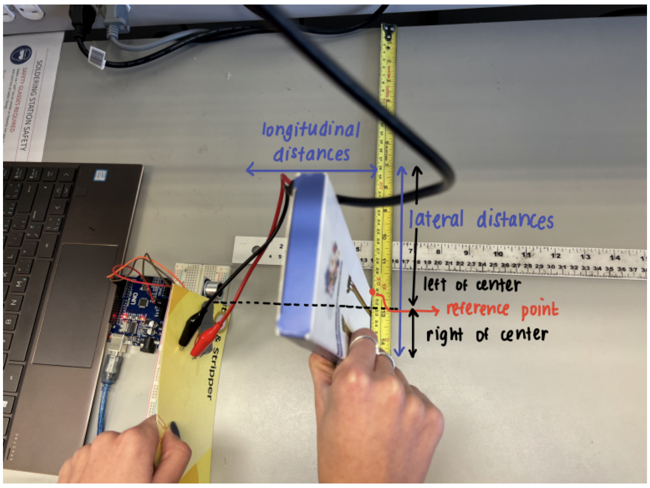


Figure 10: Lateral distance limit testing of ultrasonic sensor with the piezoelectric ceramic component.

Table 2: Lateral Limit Results for the Ultrasonic and Piezoelectric Detecting Modules

	Piezoelectric Ceramic		**HC-SR04 Ultrasonic Sensor**	
Direction	**Distance from Reference Point to Center (cm)**	**Reported Distance**	**Distance from Reference Point to Center (cm)**	**Reported Distance**
right	4	92	4	79
right	3	95	3	79
right	2	12	2	86
center	0	11	0	10
left	2	11	2	9
left	4	11	4	10
left	6	11	6	9
left	7	12	7	10
left	9	12	9	10
left	10	84	10	74
left	11	83	11	74

As seen in Figure 10, the bottom corner of the box acted as the reference point and the box was displaced at increment distances from right to left of the centre of the detection module. With a greater amount of the ultrasonic sensor covered by the object, the precision of the detecting module was more accurate. Therefore, the lateral distance was the most accurate when the reference point of the object was placed at the centre until 9 cm to the left for both the ultrasonic and piezoelectric ceramic modules.

3.1.0.2 Conclusion

Both detection modules were able to successfully generate and receive ultrasound waves and determine the distance to an obstructing object. Optimal longitudinal and lateral detection ranges were detected at which an object can be identified. Although, the piezoelectric detecting module setup was not used for the design since the accuracy did not compare to using the ultrasonic sensor on its own. As discussed in the longitudinal testing, the ultrasonic sensor on its own was able to detect a larger range of distances at better precision.

3.2 Optimal Distance Testing

Once the ultrasonic sensor proved to be more accurate, additional testing was performed to determine at what distance the object should be placed form the sensor to achieve the best resolution. For these tests, the ultrasonic sensor was connected to the Arduino the same way as described in Section 2.2.1. The distance to next obstruction was being printed every 500 ms in the serial monitor. A measuring tape was setup adjacent to the ultrasonic sensor and a 1 inch wide 9 V battery was placed at varying distances from the sensor. This setup can be seen below in Figure 11.

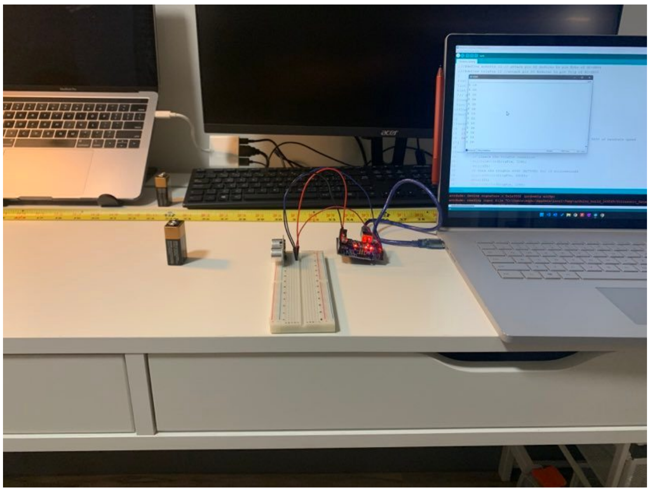


Figure 11: Setup for Optimal Distance Testing.

Four total trials were performed where the battery was placed 8", 6", 4", and 2" away from the sensor. The sensor was then shifted laterally until the right side of the battery became the next obstruction. At this point a measurement from the top of the breadboard to the bottom of the desk was taken (for example, this measurement was 10" for the 8" trial). Next the sensor was moved laterally just to the point where the left side of the battery was no longer the next obstruction. At this point a measurement from the top of the breadboard to the bottom of the desk was taken (for example, this measurement was $4\frac{1}{2}$" for the 8" trial). Finally, the difference between the two measurements was calculated (for example, this calculations was 10" – $4\frac{1}{2}$" = $5\frac{1}{2}$" for the 8" trial). This method was repeated for all of the other trials and a summary of the results can be seen in Table 3. 4" away from the sensor was determined to be the optimal operating distance as the 1" battery was it perceived most accurately at that distance.

Table 3: Summary of the Optimal Distance Trial Results.

Distance from Sensor	Perceived Width
8"	$5\frac{1}{2}$"
6"	$4\frac{1}{8}$"
4"	$1\frac{3}{4}$"
2"	$2\frac{5}{8}$"

3.3 Stepper Motors and the Auto-Turret System

Given the rotational single-pixel scanning method, a 28BYJ-48 stepper motor was equipped in the design to rotate the ultrasonic sensor and scan each pixel of the target object grid. Operating at a 5 DC voltage, the stepper motor made 64 steps per 360° rotation, which translated to a fixed 5.625° angle increment per step [12]. As regulated in the Arduino code, the spinning direction was controlled depending on the sequence of pulses. This let the motor spin both in the clockwise and counterclockwise directions and allowed the ultrasonic sensor to scan consecutive pixels of the grid. Additionally, the frequency and number of the pulses determined the speed and how far the motor turned, respectively [12]. The motors were set to spin at 5 rotations per minute, equal to 1 rotation every 12 seconds, and therefore 5.33 steps per second. Although these design specifications were controlled and altered within the Arduino code, an encountered challenge was the high degree increment per step. Due to the chosen stepper motor, each step corresponded to a 5.625 degree increment which forced the ultrasonic sensor to group together a greater section of the grid per scan. A limitation of the ultrasonic sensor is that the maximum angle of measurement is 15°. The measuring angle of the ultrasonic sensor would then be exceeded after 3 steps from the origin in the x or y directions. When the measuring angle exceeds 15°, the transmitted sound wave reaches the target, however, the receiving wave is reflected at the same angle of incidence and it does not follow the same trajectory as the transmitted wave. As a result, the wave would not reach the ultrasonic receiver and the echo pin would be unable to register the wave. This phenomenon is shown below in Figure 12. Ultimately, the design would only be able to complete 5 steps in the x and y directions which would result in a 5 by 5 grid, which would not attain an appropriate resolution to detect hand gestures and objects. For this reason, the imaging module alternative outlined in Section 2.4.1 was used in the final design.

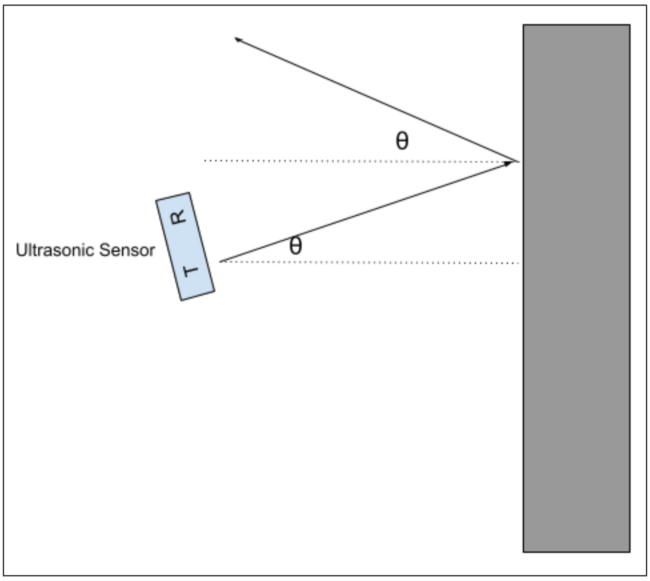


Figure 12: Reflection of Sound Wave at an Angle.

3.4 3D Printer Gantry System

3.4.1 Size

The 3D printer gantry system has an equal fixed distance in which the hot end can move in the horizontal and vertical direction. This aligns well with the goal of producing a square image. The maximum distance that can be covered in the horizontal and vertical directions is roughly 20 cm, which means the maximum area which can be imaged is about 400 cm^2.

3.4.2 Scan Time

The 3D printer utilizes a jack screw connected to a stepper motor to move the detector module vertically, and a belt systems connected to a stepper motor on one side to move it in the horizontal direction. The jack screw mechanism moves at a much slower pace than the belt mechanism, so the vertical translation of detecting module was then the limiting factor. Additionally, the horizontal movement was a continuous movement, whereas the vertical movement was done in steps as outlined in Figure 6. The translation speed was set to be as fast as the printer could support (about 250 mm/s), so the only remaining parameter to change was the size of the vertical steps. Changing the size of the vertical steps introduced a trade-off in which image resolution was sacrificed for the scan speed. Table 4 shows the resulting scan time and total number of vertical steps for various step sizes.

Table 4: Sample Time and Image Resolution for Various Vertical Step Sizes.

Step Size (mm)	Scan Time (Minutes)	Number of Steps
1	15	200
2	8	100
4	2	50

Since 8 and 15 minutes is an unreasonable length of time for a single scan, it was decided that a vertical step size of 4mm would produce optimal results in terms of scanning speed.

3.5 Image Resolution

According to the results in Table 4, the resolution corresponding to the 4 mm step size is only 50 samples by 50 samples for a square image (since the height limit is 20 cm). Since the number of vertical steps was confirmed to be 50, it was then necessary to select a sample frequency which would result in 50 samples for each horizontal translation. This was done through a process of trial and error. It was found during this process that the speed at which the detecting module was translated from side to side by the 3D printer was inconsistent. This meant that even though the sample rate remained constant, the number of samples for each row would vary. The solution to this was to sequentially add or remove samples from alternating ends of each array of samples until the desired size was achieved. This resulted in image distortion.

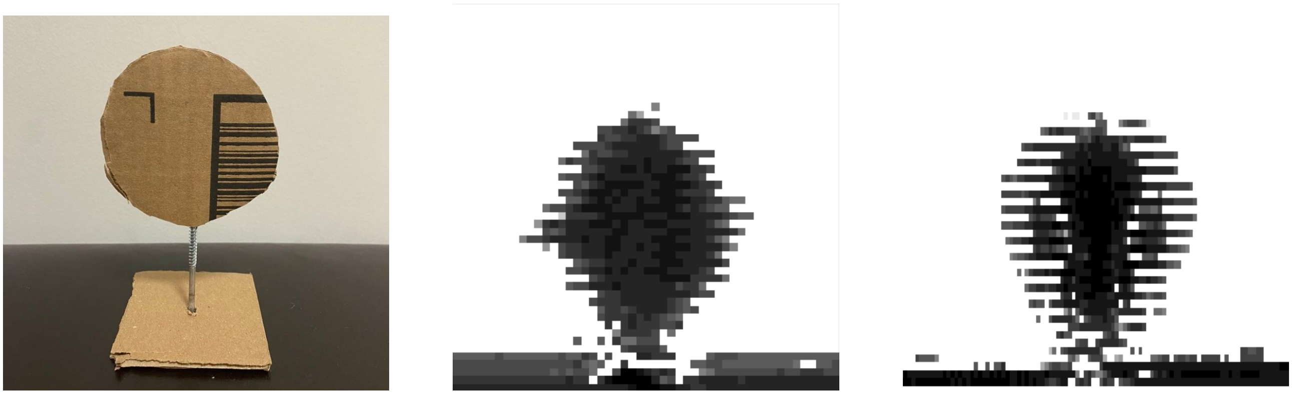
In an attempt to mitigate this, the effect of the inconsistent track speed on different sample frequencies was evaluated. Table 5 shows the maximum and minimum number of samples found in a row after a completed scan for two different sample frequencies.

Table 5: Variation in Number of Samples Per Row for Different Sample Frequencies

Sample Period (ms)	Desired Samples	Max Samples	Min Samples	Variation
17	50	66	49	17
12	75	88	58	30
7	100	132	87	45

From these results, it is seen that the variation is greater as the sample frequency increases. In fact, doubling the number of desired samples more than doubles the variation seen in the actual number of samples taken for each row. This indicates that increasing the resolution of the image exacerbates the severity of the image distortion.

To measure the actual effect on the quality of images taken, images were produced with the sample frequencies corresponding to 50 and 100 desired samples. Note that in the case where 100 samples were taken in the horizontal direction, the number of vertical samples needed to be doubled to maintain a square output image. Instead of decreasing the vertical step size to 2mm and drastically increasing the scan time, an alternative data manipulation method was employed. After processing the serial data using the method described in Section 2.5, each horizontal row was duplicated before being appended to the two dimensional `Numpy` array. This allowed for a greater image resolution while still producing a square image without drastically increasing scan time. An example of the resulting image can be shown in Figure 13



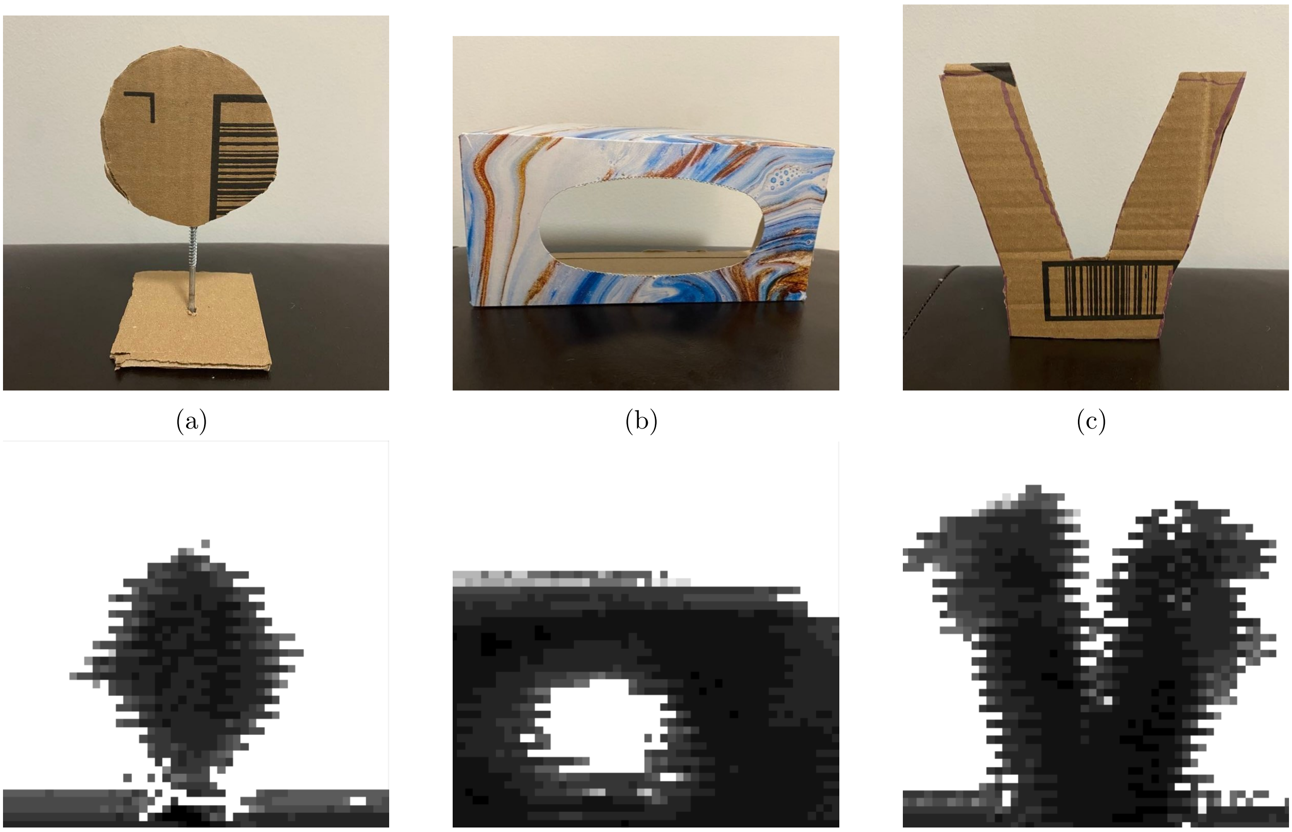
(a) Original (b) 50 Samples (c) 100 Samples

Figure 13: Image of the Circle Shape Taken with Varying Numbers of Horizontal Samples.

This shows that although we increased the resolution of our image, the resulting image distortion created a lower quality image. Trials were also completed for the other two shapes used to demonstrate the final design, however they showed the same trend and were omitted for brevity. From this, we concluded an image resolution of 50 by 50 samples best balances resolution and distortion. This justifies the use of a 17ms sample period as described in Section 2.3.1.

4 Results & Discussion

Three objects with distinct shapes were found to test the final design. The first was a floating circle, the second was a hollow tissue box, and the third was a V-shape. The resulting images produced of these shapes can be seen in Figure 14. Note that the dark line along the bottom of each image is the stand used to level the objects with the bottom of the scanning area.



(a) (b) (c)

(d) (e) (f)

Figure 14: Images of the Real Objects Next to Images Taken with the Final Design of the Ultrasound Machine.

Two key limitations were found while designing the system; the 15 degree rotational limit on the HC-SRO4 module, and the configuration of the 3D printer's gantry system.

The initial imaging module design facilitated a single-pixel imaging technique shown in Figure 3. The design and process used to realize this technique are described in Section 2.4.2. It was found that the HC-SR04's 15 degree rotational limit coupled with the minimum stepper motor rotation of 5.625 degrees meant that the resulting image resolution would be too small, and thus another imaging module design was used. It is important to note that methods of reducing the minimum step motor rotation were considered. For example, a gear system could have been implemented to translate the 5.625 degree step motor rotation to a lesser rotation of the detecting module. That being said, the resulting image size would still be too small. If we consider the optimal imaging distance of 4" found in Section 3.2, the final image area can be determined using simple trigonometry as shown in Figure 15. Even if ideal motors existed which could produce an unlimited number of steps before the 15 degree threshold is met, the resulting image would still only be about 2 inches in height and width if the optimal measuring distance is employed. This further supports that the auto-turret design is sub-optimal for this purpose.

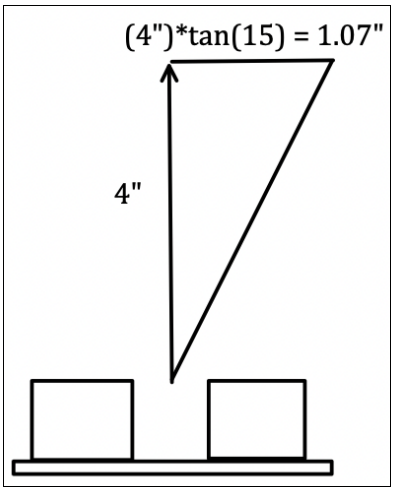


Figure 15: Calculation for maximum image area.

Another challenge was encountered when configuring the 3D printer's gantry system. Since the stepper motors in the 3D printer could not be manipulated directly, the code that moved the ultrasonic sensor could not be integrated with the code that took the distance samples. This meant that the ultrasonic sensor sampling could not be synchronized with the movement of the ultrasonic sensor. As outlined in Section 3.5, this resulted in an inconsistent number of samples in each row and image distortion.

5 Conclusion

The detecting, transducer pulse, imaging, and analog processing modules that were designed for this project successfully produced 3 distinct images. This project also successfully accomplished the bonus challenge of measuring distances using the piezoelectric material as the transducer and the ultrasonic sensor as the receiver. If this project were to be undertaken again in the future, it is recommended that a custom gantry system be made with stepper motors which can be controlled by the same script that controls the collection of samples. This would allow the samples to be synchronized with the steps of the motors and would avoid the inconsistency issue that was found with the 3D printer gantry setup. It is also recommended that a grid of piezoelectric materials which operate at the same frequencies be used in addition to the translational motion to capture samples of the same point multiple times. This setup would then require very careful data processing to construct the resulting image by using an average of each sample and to avoid mixing up data points. Additionally, the 2D gantry system uses only one stepper motor to move the hot end vertically. If a stepper motor was implemented on either end of the frame to assist with moving the detecting module vertically, the jack screw mechanism could be accelerated and the scan time would then be greatly reduced. Although there is certainly room for improvement, this project successfully reinforced the medical imaging course concepts.

References

[1] T. G. Leighton, "What is ultrasound?" *Progress in biophysics and molecular biology*, vol. 93, no. 1-3, pp. 3–83, 2007.

[2] E. Jacques, "Ultrasound therapy treatment for chronic pain," November 2022. [Online]. Available: https://www.verywellhealth.com/what-is-ultrasound-therapy-2564506

[3] K. Pottinger, "Ultrasound explained," April 2021. [Online]. Available: https://facty.com/anatomy/general-interest/ultrasound-explained/1/

[4] J. Yeow, "Physics of Medical Imaging Project," pp. 1–12, October 2022. [Online]. Available: https://learn.uwaterloo.ca/d2l/le/content/829206/viewContent/4622438/View

[5] R. Burnett, "Understanding how ultrasonic sensors work," March 2020. [Online]. Available: https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm

[6] Last Minute ENGINEERS, "How HC-SR04 Ultrasonic Sensor Works & Interface It With Arduino." [Online]. Available: https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/

[7] R. N. Tutorial, "Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino." [Online]. Available: https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/

[8] E. Freaks, "Ultrasonic Ranging Module HC - SR04." [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[9] Z. Qiu, Z. Zhang, J. Zhong *et al.*, "Comprehensive comparison of single-pixel imaging methods," *Optics and Lasers in Engineering*, vol. 134, p. 106301, 2020.

[10] Dejan, "G-code explained — list of most important g-code commands." [Online]. Available: https://howtomechatronics.com/tutorials/g-code-explained-list-of-most-important-g-code-commands/

[11] P. Hub, "Ultrasonic Sensor HC-SR04 with Arduino Tutorial." [Online]. Available: https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6

[12] MikroElektronika, "28BYJ-48 – 5V Stepper Motor." [Online]. Available: https://download.mikroe.com/documents/datasheets/step-motor-5v-28byj48-datasheet.pdf