ACOUSTIC MANIPULATION OF SOUND

WITH SOFT MATERIAL-BASED ACTUATORS

By

YANJUN WANG

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Mechanical and Aerospace Engineering

Written under the direction of

Dr. Aaron D. Mazzeo

And approved by

_____

_____

_____

New Brunswick, New Jersey

October, 2015

ABSTRACT OF THE THESIS

ACOUSTIC MANIPULATION OF SOUND WITH SOFT MATERIAL-BASED ACTUATORS

By YANJUN WANG

Thesis Director:

Dr. Aaron D. Mazzeo

This thesis describes a unique parabolic acoustic manipulator with an inflatable structure, which has high gain and directivity. We created a morphable elastomeric reflecting surface with a diameter of 14 cm (6 in). Applying vacuum deforms the device into a concave structure, which provides directional amplification of incoming acoustic waves. In addition, the author characterized the impedance of the soft material employed in the acoustic reflector, Ecoflex 00-10, in an impedance tube. Ecoflex 00-10 has a measured reflection coefficient of approximately 0.9 at frequencies ranging from 500 Hz to 5000 Hz. This new characterization suggests this class of silicone-based elastomers is capable of advanced morphable devices to manipulate sound. Simulations also demonstrate that the soft reflecting surface is capable of transformation into a set of desired parabolic shapes between an initial planar geometry (neutral position) and a configuration with maximum curvature. With an applied vacuum, the membrane reaches its maximum deformation limited by the aluminum housing. At this stage of actuation, experimental results show the deformed membrane has similar gain and directionality (polar response) as rigid parabolic reflectors. This type of system might find future uses for adjustable parabolic microphones and long-range communication devices.

ACKNOWLEDGMENT

First I would like to express my sincere gratitude to my advisor Dr. Aaron D. Mazzeo for his guidance, patience, knowledge and his passion in soft robotics and related research. Special thanks to Professor Baruh, for the time we spent together creating the figures for the book Applied Dynamics, those Inkscape skills I learned from it benefited me a lot in my later research.

Many thanks to my fellow lab mates, especially Jingjin Xie, Xiangyu Gong, Chen Yang, and Ke Yang for all the discussion, help, and advices. I would also like to thank the undergraduate students that contributed in this project, especially Robert Froster, Harry Song, Adam Burrous, and Billy Teagarden.

Finally, I would also like to thank my family, and my friends. I had to run all my experiments after midnight to have less noise and interruption, and I got really tired and frustrated by the results sometimes. It was the understanding, support, and encouragement from my family and friends that helped me through all the hard times.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

1.1 Motivation

Conventional hard material-based parabolic microphones and speakers for acoustic enhancement normally fix their focal point at a given location. Such systems employ a rigid, concave surface with non-adjustable acoustic properties, such as gain and directionality. Interestingly, similar biological systems for directional hearing and broadcasting are not always in a perfect parabolic shape, but still provide some ability of adjustment. A concave collection of feathers, or a facial disc, helps owls direct sound waves toward their ears [1]. Humans and some primates are also capable of putting hands in front of their mouth while vocalizing to increase the directivity of their voice.

One key parameter to characterize a parabolic microphone is the half-power beamwidth, which is the angle between the half-power (-3 dB) points on the main lobe of the polar response. This half-power beamwidth generally decreases dramatically as the input frequency increases. The MKH8020 from Wildtronics (a professional parabolic microphone manufacturer) has a half-power beamwidth of approximately 15 degrees of at 3150 Hz and less than 5 degree at 6300 Hz [According to the manufacturer's data sheet]. Although the gain is remarkable at those high frequencies, for animal-based vocalization recording, such as birds chirping which can go up to 5k Hz [2], the beamwidth might actually be too narrow and pose difficulty in tracking a moving sound source.

A Long Range Acoustic Device (LRAD), a type of directional acoustic hailing device that employs the similar principle in reverse (i.e., sound source instead of a microphone),

has various applications in law enforcement and military to send warnings at a distance or as a determent of wildlife from airport runways and other facilities. The New York Police Department is equipped with LRAD (sound cannon) and has used it against protesters since 2004 [3]. The deployment of LRAD and its power have been controversial as injuries have been reported by individuals exposed to the LRAD at unsafe distances.

Given the limitations of current acoustic enhancement systems with fixed focal points, a mechanical system with adjustable focus, gain, and beamwidth might be desirable. Recent developments in flexible electronics and biomimetics have demonstrated the ability of soft material-based thin membranes to transform from an initially planar geometry into desired curved shapes [4]. In 2011, Inhwa Jung et al. developed a dynamically tunable hemispherical electronic eye camera system with adjustable zoom capability achieved by the hydraulic actuation of a thin elastomeric membrane.

1.2 Objective

The objective of this work is to demonstrate how elastomeric actuators with large deformations can manipulate the reflection, directionality, and focusing of sound for acoustic sensors. Recent advances in soft materials with various applications have demonstrated the potential of developing soft material-based mechanical system to generate sound [5], but there have yet to be demonstrations of their ability to manipulate sound. The new soft material-based reflectors described in this work are lightweight, inexpensive, and simple to fabricate. By controlling the vacuum pressure applied to the device, the deformation of the surface renders different levels of amplification or attenuation. The reflecting surface is capable of changing its curvature going from a flat,

neutral position without applied vacuum, to a maximum curvature with an applied vacuum under 1 atm.

1.3 Background Information

A parabolic microphone is one type of acoustic sensor that has directional sensitivity to sound as described by Olsen (1957) [6]. A reflector shaped in a parabola collects and focuses the acoustic energy to a microphone to pick up the sound at a distance [7]. Since the dish-like microphone is effective in enhancing sounds from distant objects, common uses include recording localized noise in nature, animal-based vocalization and close-up action at sporting events or in movie production.

An expression for the geometry of a parabolic dish with a focal length F in x, y coordinates follows

$$x^2 = 4F(y - F), \tag{1}$$

where $|x| \leq \frac{D}{2}$ and D is the diameter of the dish as shown in **Figure 1**. In practice, the focal length can be expressed as $F = \frac{D^2}{16H}$ where H is the depth of the dish. Common ratios of $\frac{F}{H}$ range from 0.25 to 0.65 and larger $\frac{F}{H}$ correspond to higher efficiencies and less sensitivity to geometrical variations.

The theoretical gain of a parabolic reflector is

$$G_{dB} = 10log_{10}\left(\frac{4\pi A}{\lambda^2}\, e_A\right) = 10log_{10}\left(\left(\frac{\pi d}{\lambda}\right)^2 e_A\right), \tag{2}$$

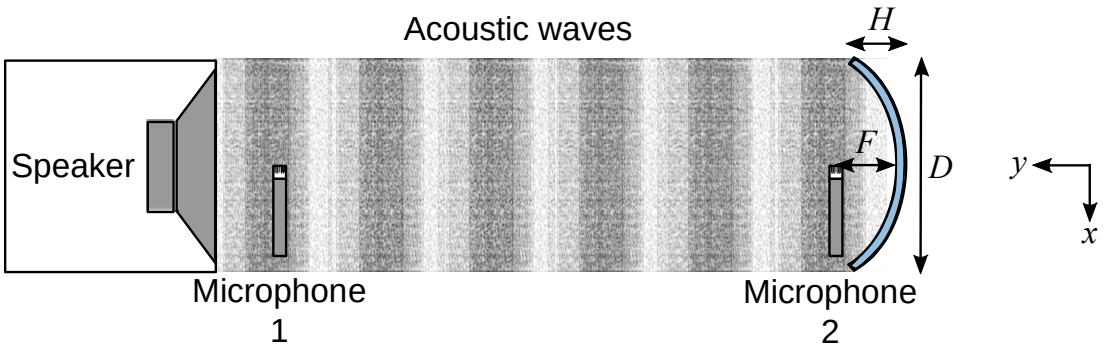

Figure 1. Schematic of the parabolic dish and measurement setup. The sound pressure level is measured both at the speaker and the focal point of the elastomeric parabolic dish (marked in blue color). We placed the sound source along the normal angle of incidence in front of the parabolic reflector.

where *A* is the aperture, project area of the parabolic reflector; *D* is the diameter of the parabolic reflector; λ is the wavelength of the acoustic waves; e_A is called aperture efficiency, a dimensionless parameter between 0 and 1. As an example, a dish with a diameter of 15.2 cm (6 in) and efficiency of 50% can produce gains greater than 5 dB for frequencies greater than 5000 Hz.

2. Experimental Design

2.1 Impedance Tube measurement

To demonstrate silicone-based elastomers common to many current pneumatic soft robots [3-5] would be capable of reflecting acoustic waves, we include data for acoustic reflection and absorption of Ecoflex 00-10. We constructed an impedance tube using two microphones and a digital analysis system (transfer-function method) in order to characterize the acoustic reflection and absorption of the elastomer used in manufacturing the soft reflecting dish. We designed the tube to measure the absorption and reflection coefficients from 300 Hz to 4500 Hz, according to criteria from ASTM E1050-12 [11] and ISO 10534-2:1998 [12]. The tube body is two pieces of Schedule-40 PVC tube, with an inner diameter of 38 mm (1.5 in); the length of the testing area is 324 mm (12.75 in). The distance of the closest microphone position to the test sample is 83 mm (3.25 in). A retractable solid backing plate located in the other segment of the tube provides a terminating sound reflector.

When taking measurements, we supplied a random noise at one side of the tube with a speaker connected to a laptop running MATLAB©, and we measured the sound pressure simultaneously at two spaced locations along the tube with the distance of 12 cm. By decomposing the data into forward- and backward-traveling components, calculation of the normal-incidence absorption and reflection coefficients were possible [11]. To improve resolution at high frequencies ($> 1000\,Hz$), we decreased the spacing between microphones to 3 cm and also employed a metal insert as a dummy microphone to seal the remaining port. The acquisition of data was through two Brüel & Kjaer (B&K) 4961 multi-

field microphones, a NI-9234 module, a NI cDAQ-9171 USB chassis, and the laptop running MATLAB©.

To calculate acoustic absorption and reflection coefficients, we need to calculate the complex transfer function between the two measurement points, which is defined by

$$H_{12} = \frac{S_{12}}{S_{11}} = \frac{p_2 \cdot {p_1}^*}{p_1 \cdot {p_1}^*} = |H_{12}| e^{j\phi} = H_r + j\, H_i\,, \tag{3}$$

where H_r is the real part of H_{12}; H_i is the imaginary part of H_{12}; S_{12} is the cross spectrum; S_{11} is the auto spectrum; p_1, p_2 are the complex sound pressures at tow microphone positions; ${p_1}^*$ is the complex conjugate of p_1. In practice, we found the use of the MATLAB© function tfestimate to be extremely helpful in producing a smooth transfer function [13].

Using H_{12}, we can then calculate the reflection coefficient r as

$$r = |r| e^{j\phi_r} = r_r + j\, r_i = \frac{H_{12} - e^{-jk_0 s}}{e^{jk_0 s} - H_{12}}\; e^{2jk_0 x_1}, \tag{4}$$

where r_r is the real part of r; r_i is the imaginary part of r; s is the separation between the two microphones; x_1 is the distance between the test sample and the further microphone position; k_0 is the complex wave number defined by

$$k_0 = \frac{\omega}{c_0} = \frac{2\pi f}{c_0}\,, \tag{5}$$

where ω is the angular frequency; f is the frequency; c_0 is the speed of sound.

Using reflection coefficient r, the absorption coefficient α can be expressed as

$$\alpha = 1 - |r|^2 = \; 1 - r_r^2 - r_i^2\;, \tag{6}$$

where r_r is the real part of r; r_i is the imaginary part of r.

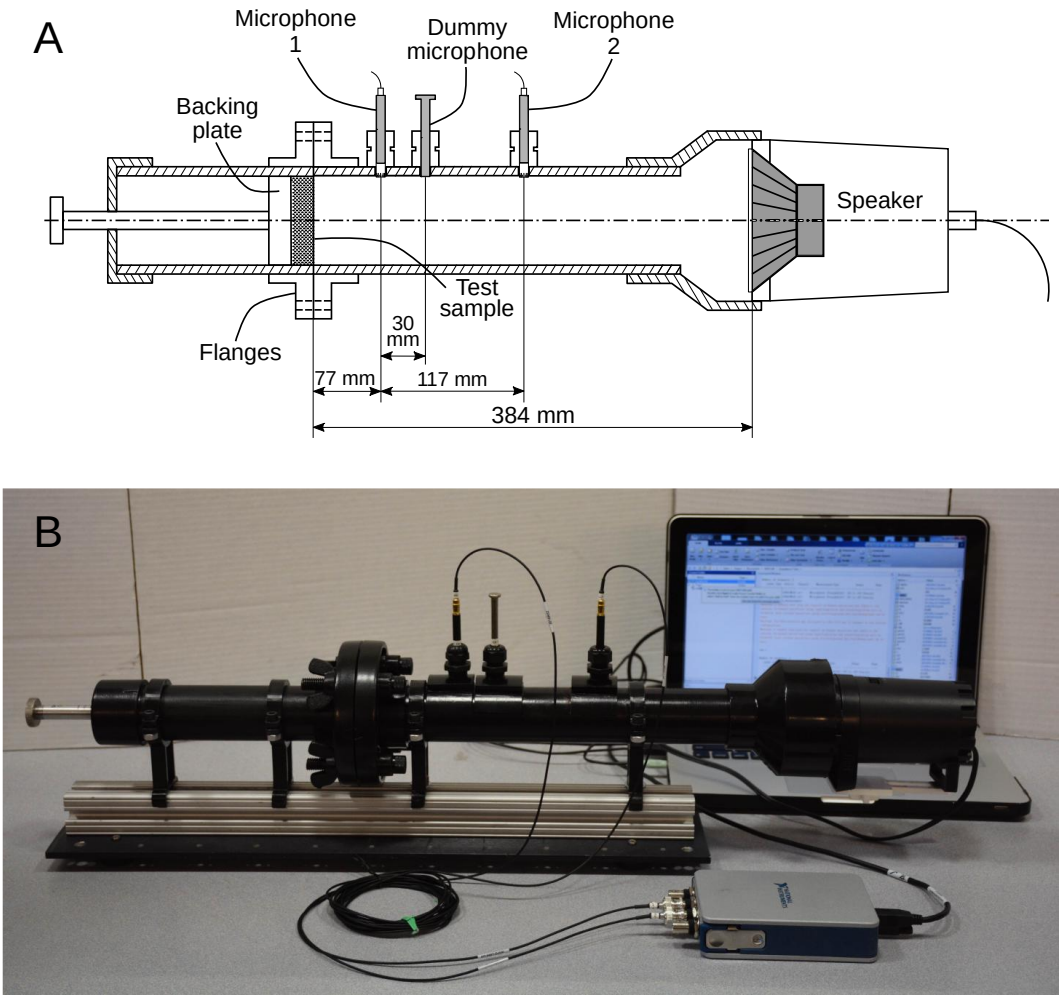

Figure 2. (A) Schematic illustration of the impedance tube and major measurements. The sample holder (left) held the sample (a piece of 20 mm thick Ecoflex 00-10 that fit snugly inside the tube) with a retrievable backing-plate; the actual testing area (right) held the microphones and sound source. Two flanges joined two sections of the tube together and provided an airtight environment. (B) Photograph of the impedance tube and the analysis system. White noise signal comes from the pre-generated audio file on the computer through the audio jack at 44100 Hz sampling rate. The data acquired by two microphones goes into MATLAB© through NI-9234 at 51200 Hz sampling rate simultaneously.

2.2 Elastomeric Parabolic Dish

Elastomer-based pneumatic actuators often employ branch-like channels, but this system uses an open cavity to provide axisymmetrically uniform deflection of a soft membrane. The membrane deflects under applied vacuum provided in an aluminum-based housing. The applied vacuum was very modest going from atmospheric pressure to -3684 psi, which we applied by pulling a plunger on a syringe (140 cc). By controlling the vacuum pressure applied to the device, the reflecting surface was capable of changing its curvature going from a flat, neutral position without applied vacuum to a maximum curvature with an applied vacuum under 1 atm.

Ecoflex 00-10, the material of the membrane, is a soft silicone-based rubber with a Shore 00 Hardness of 00-10. We prepared different molds for the soft membrane, which included a flat mold, a mold with the center 2 mm lower than its outer edge, and a mold with the center 3 mm lower than its outer edge. By changing the thickness of the membrane linearly in the radial direction, different curvatures resulted with applied vacuum.

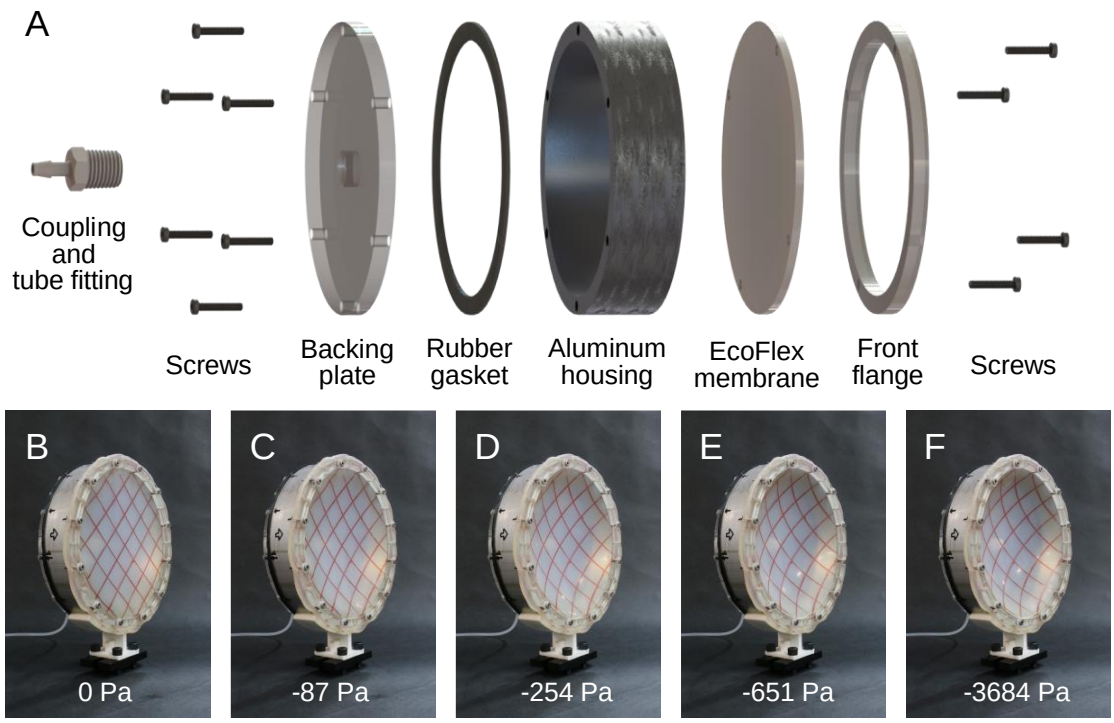

Figure 3. (A) Exploded-view illustration of the components of the vacuum cylinder: Laser cut transparent acrylic glass (backing plate and front flange), neoprene rubber gasket, and aluminum cylinder with 6.35-mm (0.250 in) wall thickness, 3-mm thick membrane of Ecoflex 00-10, tube fitting and screws. (B-F) Photographs of the deformed system under five different vacuum levels from no vacuum (left) to maximum deformation (right).

2.3 On-axis and Directional Acoustic measurements

In order to characterize the acoustic gain of our elastomeric dishes, we conducted on-axis, multiple-point measurements with two B & K multi-field, omni-directional microphones in a lecture hall. Two tripods held the system and the sound source 1.8 m above the ground; the distance between the reflector and the sound source was approximately 1 m. We swept through frequencies from 2 kHz to 10 kHz on a linear scale with a 3-second pause between each frequency. We used the one microphone closest to the sound source to detect the start and end point of the signal for its higher signal-to-noise ratio (SNR), and we took the measurement at this point close to the sound source and another point near the focal point of the soft reflecting dish. We processed the sound pressure data acquired by the microphone at the focal point of the dish in MATLAB© and fit the sinusoidal waves to the measured data. Using these fits, we determined the amplitudes of the pressure-based acoustic waves.

To characterize the directionality of system, we kept the sound source and the microphone close to the sound source stationary. We then continuously output a frequency of sound (1 kHz, 2 kHz, 3 kHz, 4 kHz, 5 kHz, 6 kHz, 7 kHz, 8 kHz, 9 kHz,10 kHz), while the reflector and the other measurement point driven by a step motor rotated by 360 degrees. We fit the sinusoidal waves to 360 points in a full revolution to produce polar plots for the measured directionality of the system.

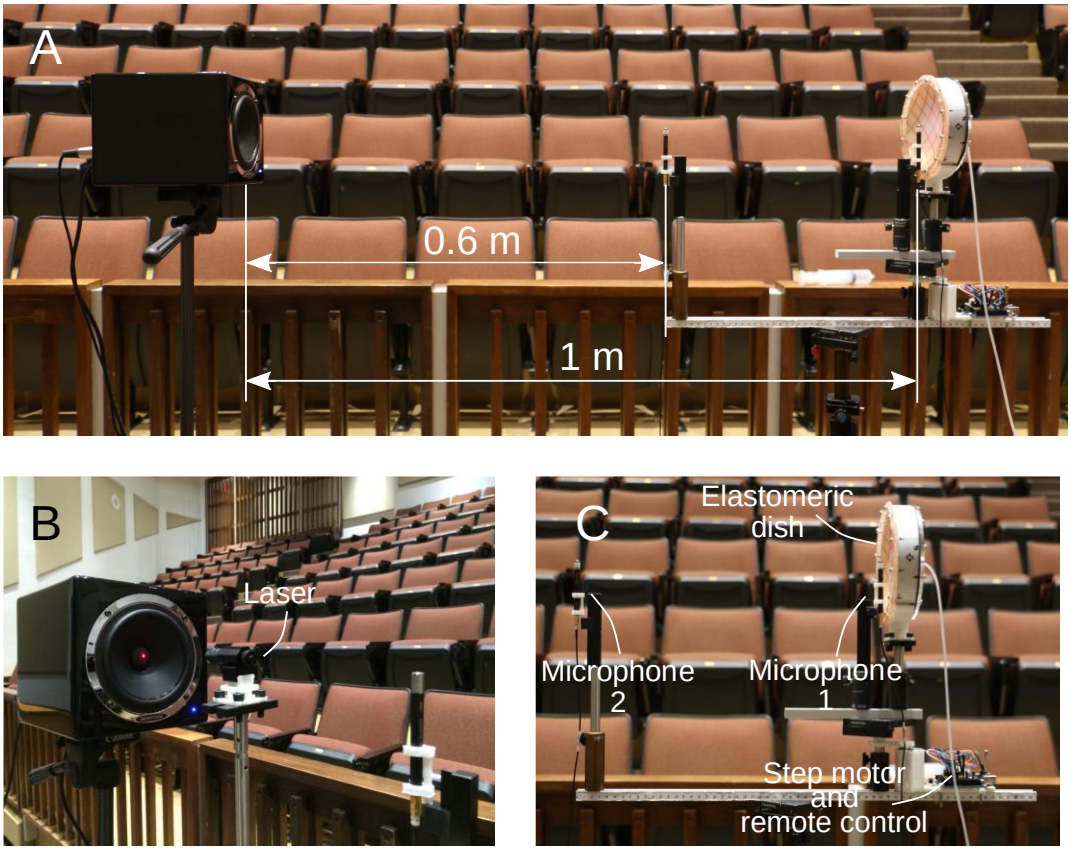

Figure 4. (A) Photograph of the measurement setup. (B) We used a laser beam to align the sound source, the parabolic reflector and two microphones. (C) Close up photograph of the parabolic reflector system with Arduino-controlled step motor system (on the bottom).

3. Fabrication Techniques

3.1 Laser Cutting

We laser cut acrylic sheets to fabricate the front flange and backing plate of the vacuum cylinder. The acrylic sheets are transparent and allowed us to view the deformation of the membrane. To cut the acrylic board in a fast and accurate manner, we used our laser cutter VLS 2.30 Universal Laser System with 2D sketches. As the laser vaporized the material with minimal lateral forces, we did not have to clamp the work pieces. The whole run took about 20 minutes, the cut was smooth and uniform along the thickness. We then tapped the center hole on the backing plate in order to fit the adapter and tube fitting. We also laser cut the mold for the elastomeric membrane, and the acyrlic sheet provided a smooth surface for molding process, which helped create a membrane with uniform mechainical properties.

3.2 3D Printing

Since our laser cutter is operated by moving the laser beam on a 2D plane, it is great for making a through cut or in other words, fabrication of accurate 2D structures. Nonetheless, for fabrication of more complex 3D structures, its application is limited, and sometimes requires the part be broken into several 2D pieces for later assembly. In order to simplify the fabrication process, we used our 3D printer, Flashforge Creator, to make 3D components. We used Sainsmart 1.75 mm PLA filament as the material, and chose 10% to 25% as the infill percentage depending on the stiffness requirement of each part.

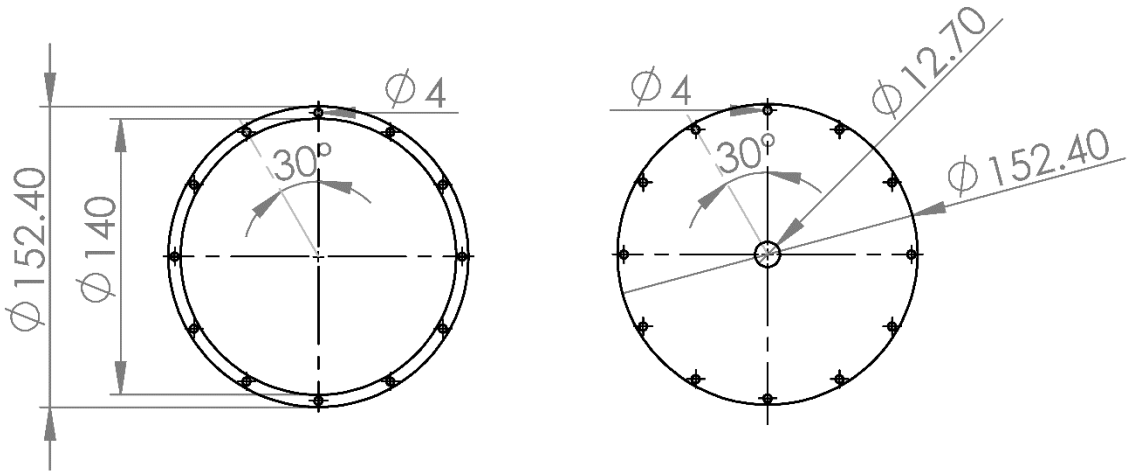

Figure 5. Laser cutting layout of the acrylic front flange and backing plate of the vacuum cylinder (unit: mm)

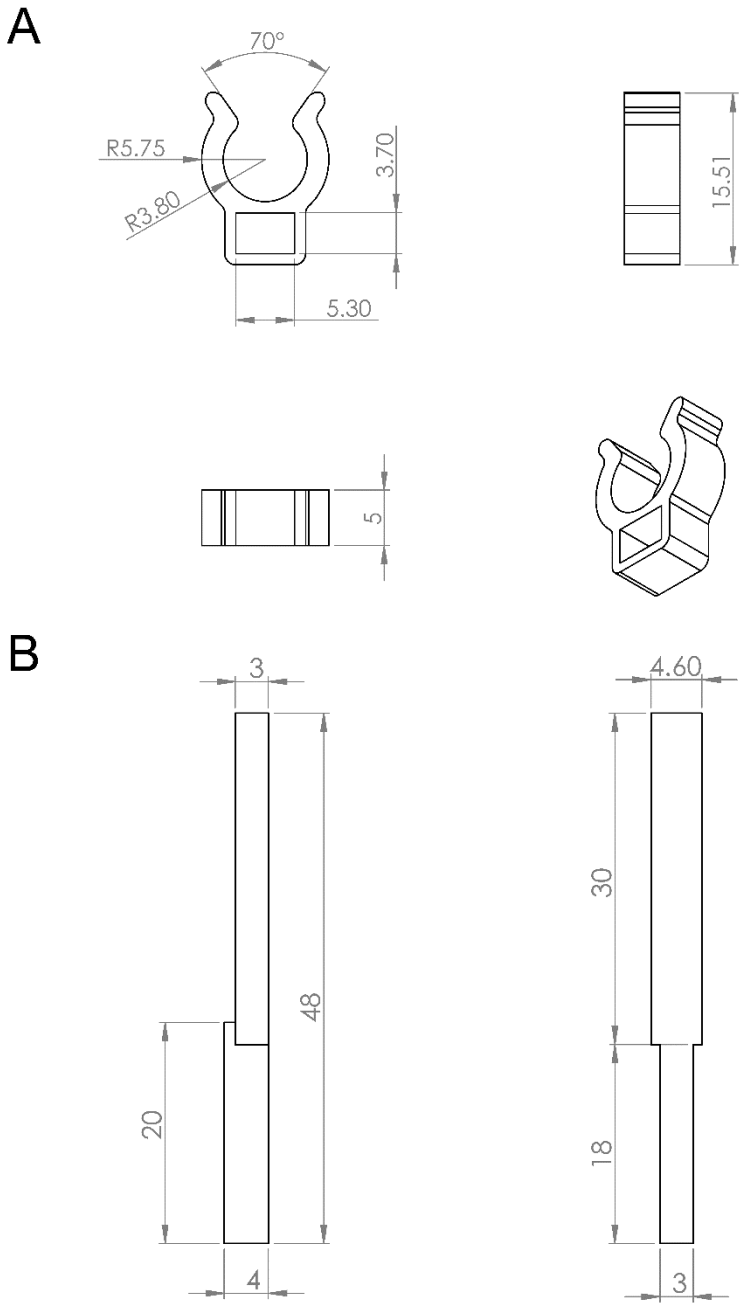

Figure 6. 3D-printed microphone clip and clip mount to hold the quarter inch B&K 4961 multi-field microphones (unit: mm)

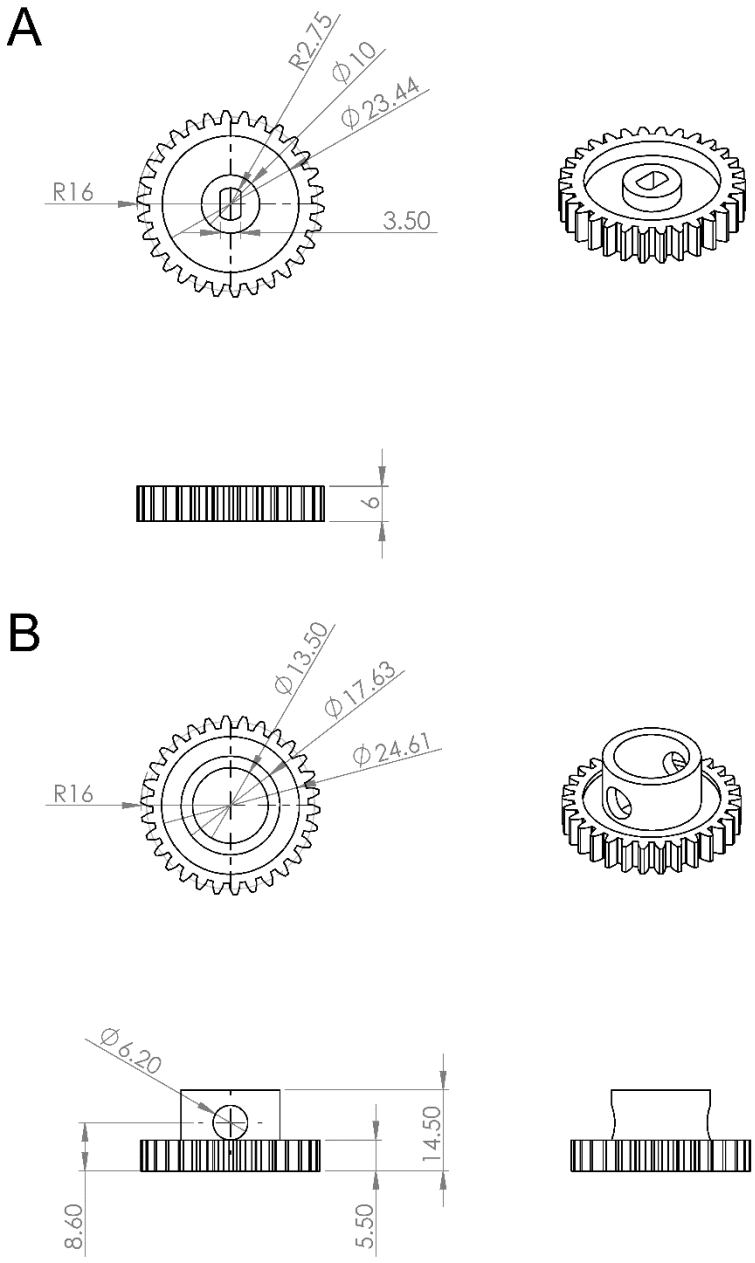

Figure 7. (A) 3D-printed gear that fits the step motor shaft in the directionality test (B) 3D printed gear that fits the mounting base of the dish system in the directionality test (unit: mm)

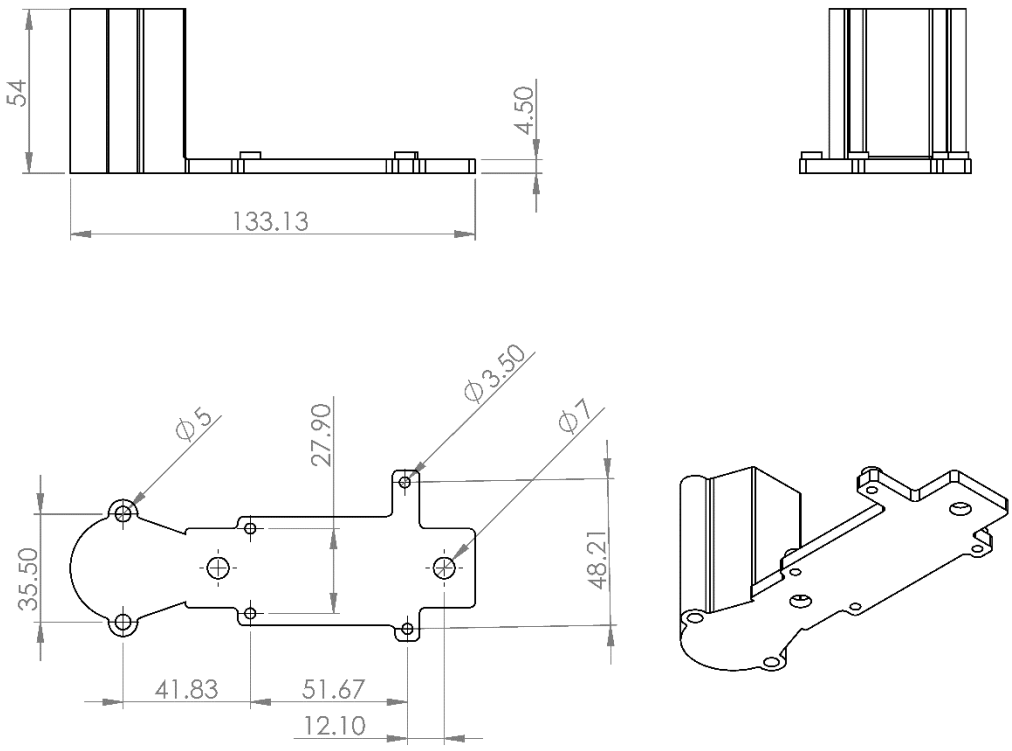

Figure 8. 3D-printed holder for Arduino and step motor base (unit: mm)

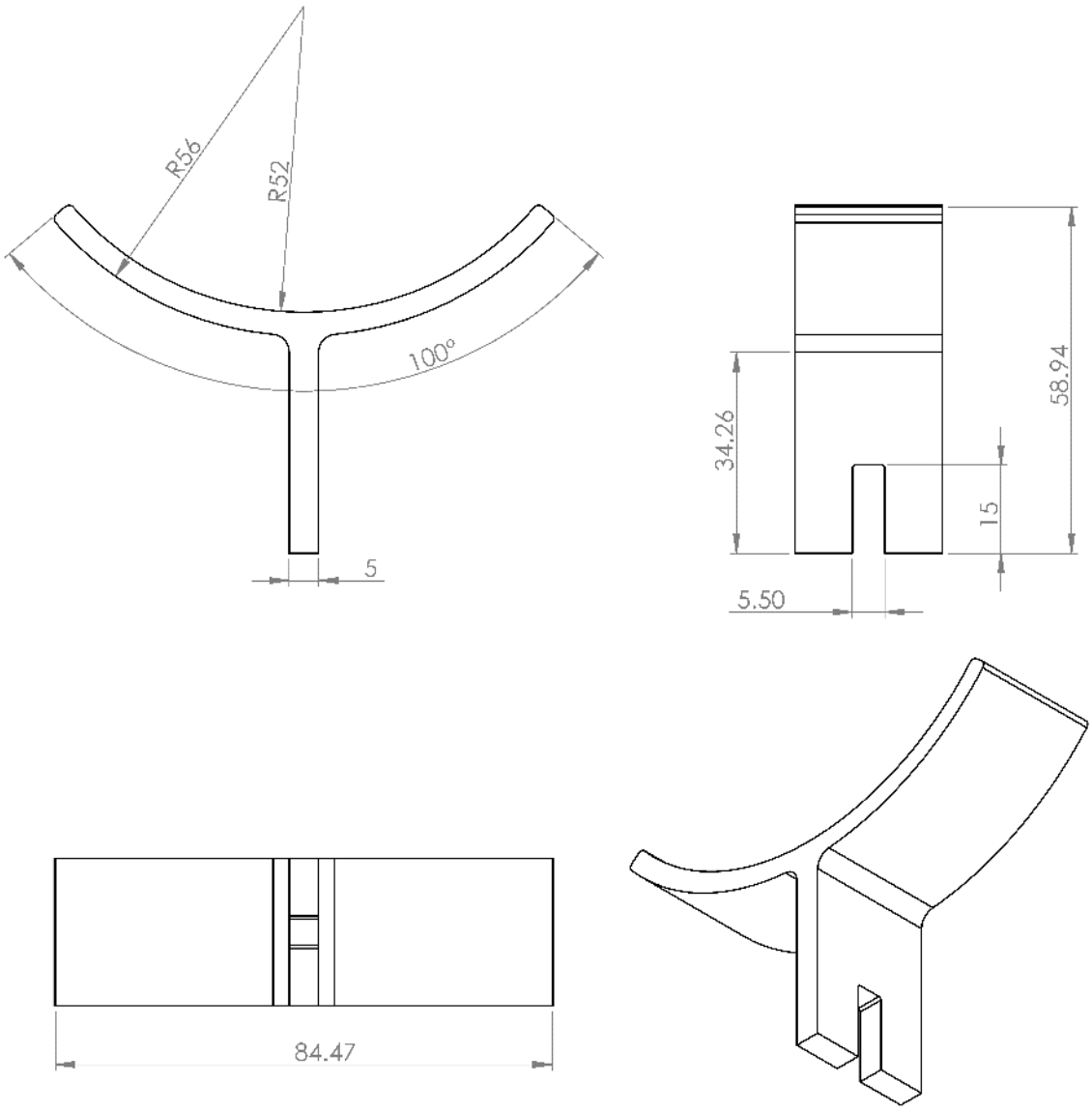

Figure 9. 3D-printed vacuum cylinder holder, the slot on the bottom fits inside the mount and key shown in Figure 10 (unit: mm)

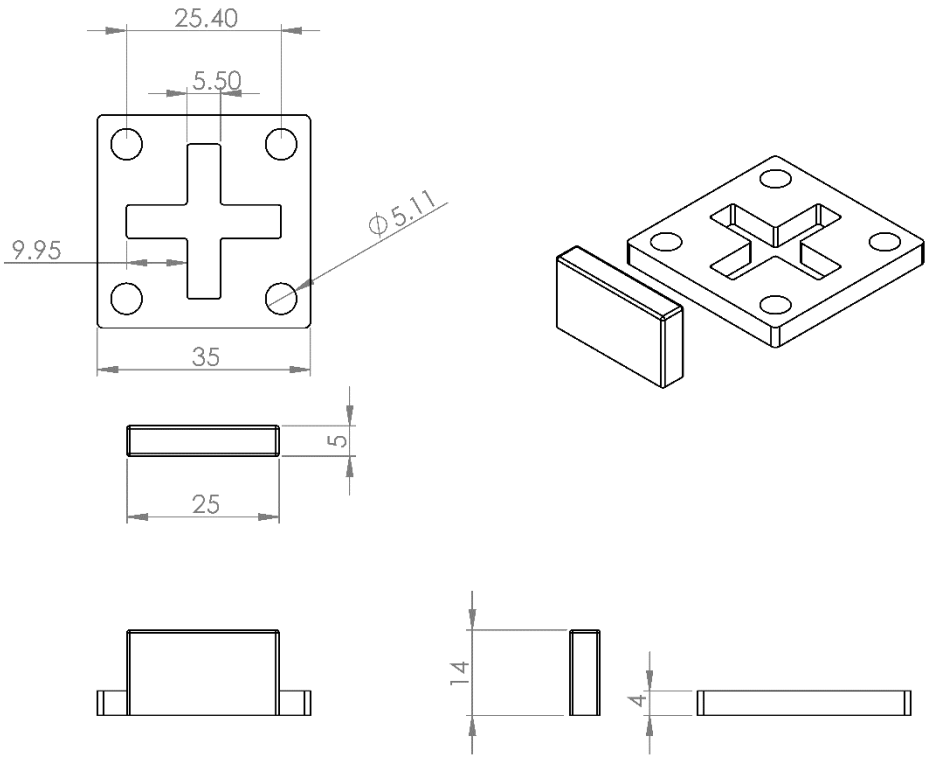

Figure 10. 3D-printed mounting base and key for the vacuum cylinder holder, and laser pointer (unit: mm)

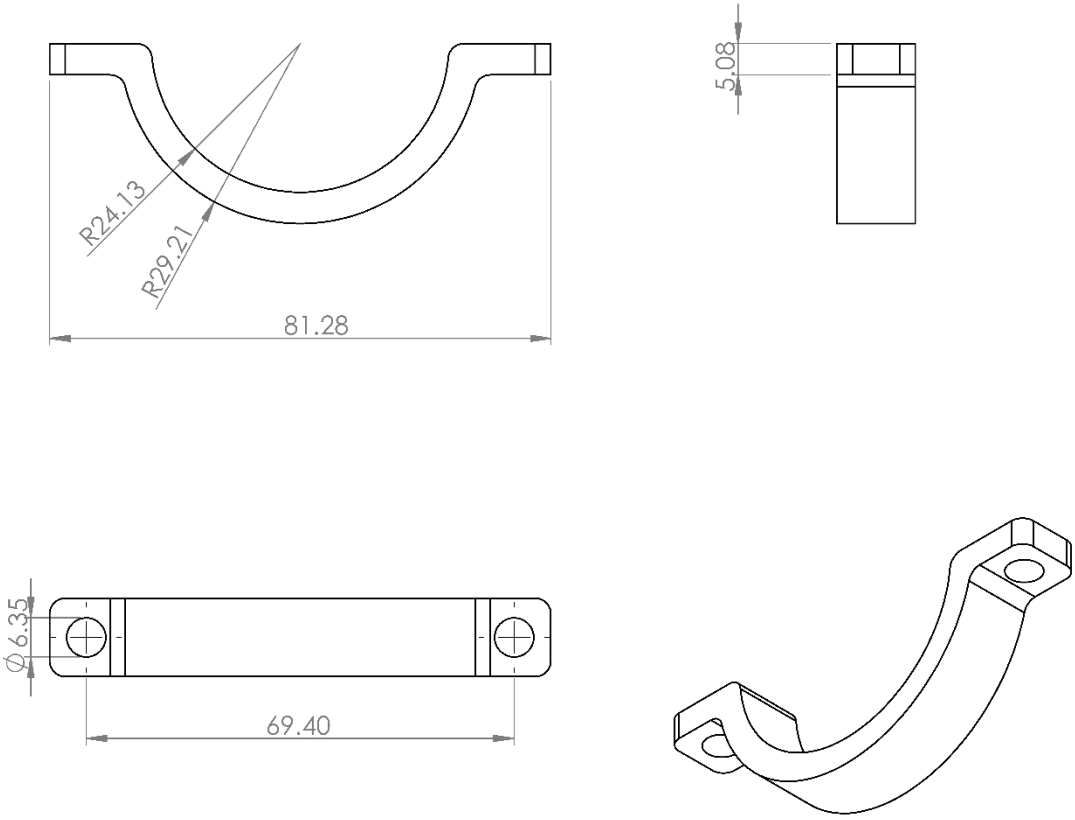

Figure 11. 3D-printed tube collar (top) for the impedance tube (unit: mm)

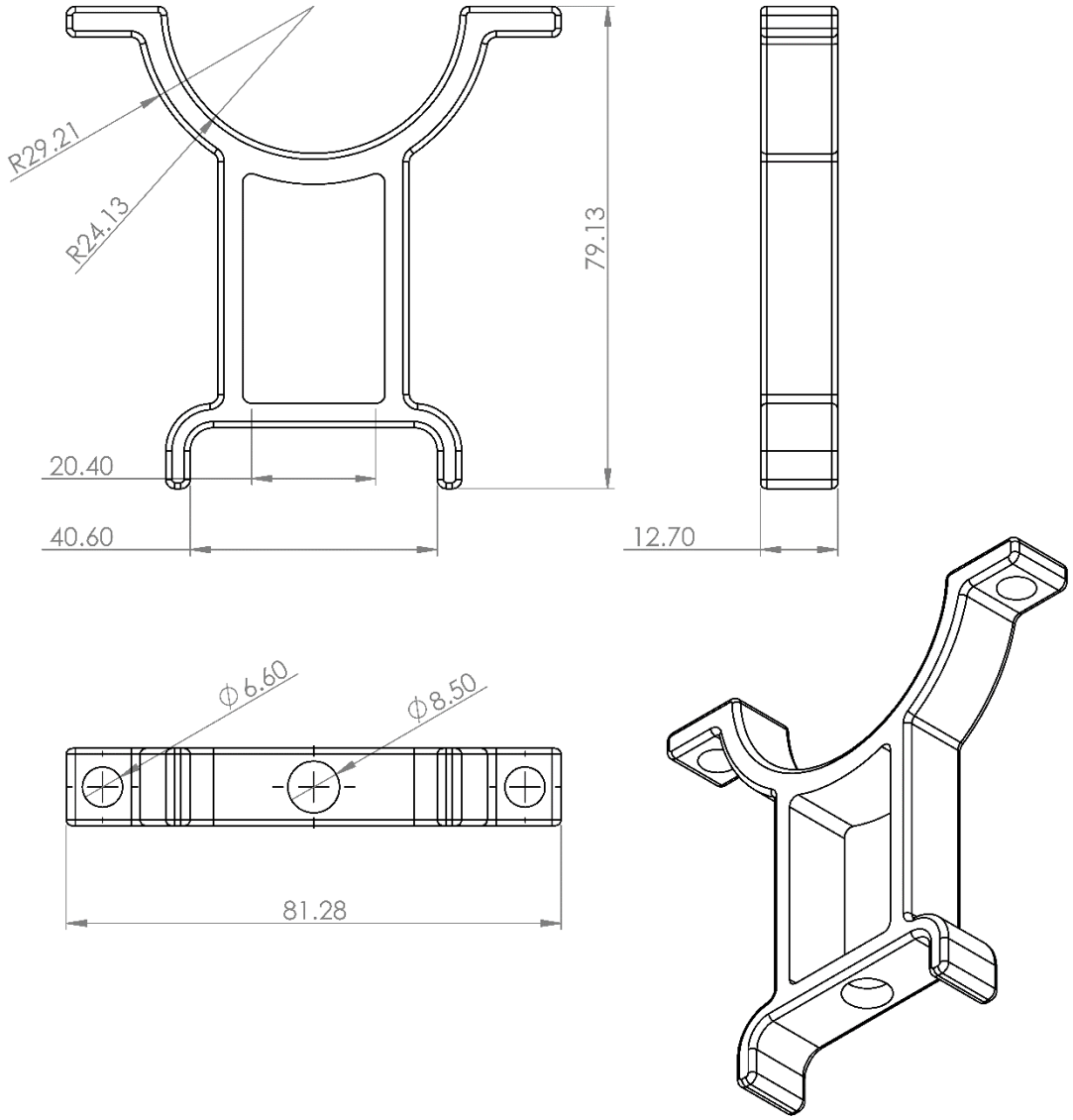

Figure 12. 3D-printed tube collar (bottom) and mounting base for the impedance tube, the bottom of it sits on top of the slotted aluminum rail (unit: mm)

3.3 Molding

In order to create our elastomeric reflector, we first poured two reagents of Ecoflex 00-10 1:1 (weight ratio) together in a cup and mixed it well. Then, we degassed the mixture in a vacuum chamber to eliminate the air bubbles inside the mixture. After approximately 10 minutes, there were no major air bubbles escaping from the liquid anymore, we took the mixture out and poured the liquid into our mold and let it sit on top of a leveled surface for a few hours to cure. The mold employed 12 screws that allow us to make 12 holes on the elastomeric reflector, those holes helped in mounting and alignment of the reflector for assembly of the vacuum cylinder. In practice, we also marked the elastomeric reflector with red grid, for image recognition and 3D re-construction.

4. Result and Discussion

4.1 Mechanics of the elastomeric membrane

In order to get a parabolic-shaped and more efficient elastomeric dish under different level of vacuum, we studied and simulated the mechanics of the elastomeric membrane in COMSOL. We considered the material, Ecoflex 00-10, as hyperelastic materials and fitted the material testing data to a 5-term Mooney-Rivlin hyperelastic material model; the strain energy density function is defined by:

$$W = C_{10}(\bar{I_1} - 3) + C_{01}(\bar{I_2} - 3) + C_{20}(\bar{I_1} - 3)^2 + C_{11}(\bar{I_1} - 3)(\bar{I_2} - 3) + C_{02}(\bar{I_2} - 3)^2 + \sum_{i=1}^{N} \frac{K_i}{2}(J_3 - 1)^{2i}, \quad \textbf{(7)}$$

where $C_{01}, C_{10}, C_{02}, C_{20}, C_{11}$ are material parameters, for uniaxial tensions, $\lambda_1 = 1 + \varepsilon = \lambda, \lambda_2 = \lambda_3 = \frac{1}{\sqrt{\lambda}}$, and I_1, I_2 are the stretch invariants defined by

$$I_1 = {\lambda_1}^2 + {\lambda_2}^2 + {\lambda_3}^2; \quad \textbf{(8)}$$

$$I_2 = {\lambda_1}^2{\lambda_2}^2 + {\lambda_2}^2{\lambda_3}^2 + {\lambda_1}^2{\lambda_3}^2, \quad \textbf{(9)}$$

and for incompressible materials $\bar{I_1} = I_1$, $\bar{I_2} = I_2$, $J_3 = 1$. For Ecoflex 00-10, stress strain data give us

$$C_{10} = -7.48 \times 10^{-3}\ MPa, C_{01} = 1.96 \times 10^{-2}\ MPa,$$
$$C_{20} = -1.98 \times 10^{-4}\ MPa, C_{02} = -5.39 \times 10^{-4}\ MPa, C_{11} = 3.21 \times 10^{-3}\ MPa. \quad \textbf{(10)}$$

We then used he coefficients above to simulate the deformation of a 3 mm-thick elastomeric membrane. **Figure 5C** shows that by varying the applied pressure on one side of the membrane from 0 pa to 900 Pascal, a sets of deformations from unactuated to the maximum deformation limited by the aluminum housing was achieved. We fit 2nd-degree

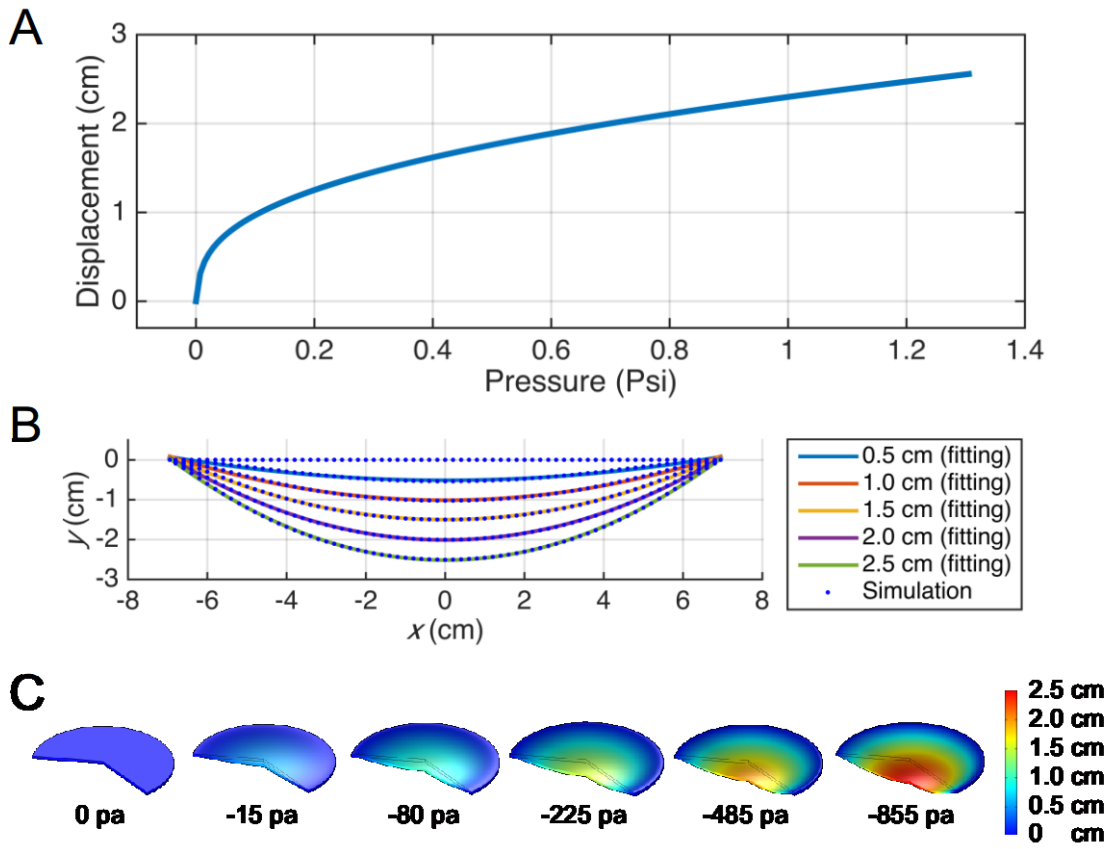

Figure 13. (A) COMSOL simulated relationship between the vacuum pressure (gauge pressure) and the displacement of the center of a 14 cm (6 in) elastomeric dish, the maximum deformation is 2.5 cm which is constrained by the thickness of the aluminum housing (B) 6 COMSOL simulated sets of profiles from 0 deformation to maximum deformation, and the 2nd-degree polynomial fit in MATLAB© for the 5 sets of deformed shape. (C) 6 COMSOL simulated deformed shapes and pressures.

polynomial to the deformed profiles in MATLAB©, **Figure 5B** shows that the deformed curvatures are almost perfectly parabolic with different depths.

4.2 Absorption and Reflection Coefficients

For acoustic characterization of the elastomer in the impedance tube, two spacings $s_0 = 0.032\ m\ (1.25\ in)$ and $s_1 = 0.121\ m\ (4.75\ in)$ between microphones were used for different frequency ranges. The minimum working frequency is generally defined by

$$f_l > 0.05\frac{c_0}{s}, \tag{11}$$

while the maximum working frequency is defined by

$$f_u < \frac{0.58c_0}{d}, \tag{12}$$

where d is the inside diameter of the tube (0.040 m).

The larger spacing (0.121 m) gives us the frequency range from 141 Hz to 4916 Hz, while the smaller spacing (0.032 m) gives us the frequency range from 536 Hz to 4916 Hz. In practice, for better accuracy of different frequency bands, we used the smaller spacing for the higher frequency range and the larger spacing for the lower frequency range with some overlap. Besides geometrical dimensions, the working frequency range, especially the minimum limit, was also limited by the accuracy of the signal processing equipment and manufacturing errors. We further narrowed the lower frequency range to 300 Hz to 1k Hz, and higher frequency range to 800 Hz to 4500 Hz.

4.3 Polar Response

To characterize the directionality of the elastomeric dish, we used the sound source (Behringer C50A Active Full-Range Reference Studio Monitor) to output a steady single-

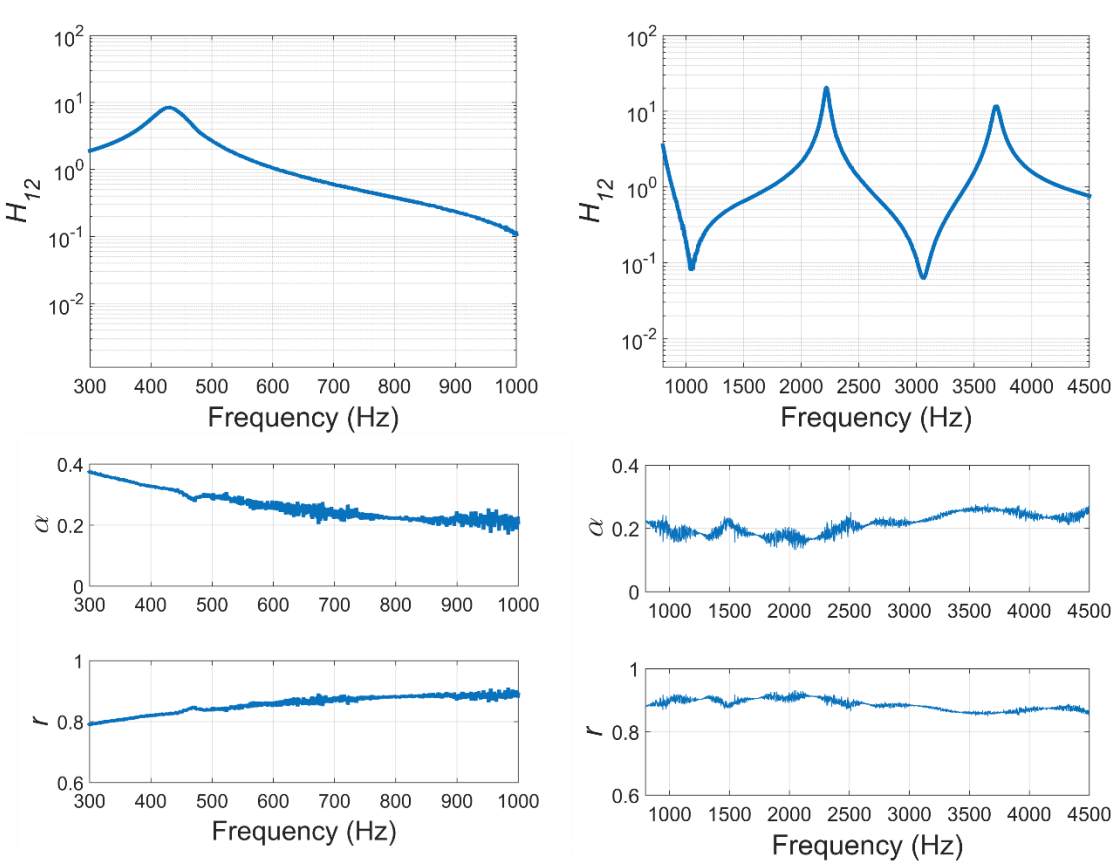

Figure 14. Acoustic reflection and absorption coefficients of a 20-mm thick Ecoflex 00-10 sample.

frequency signal, while we used an Arduino-controlled step motor driving the dish system to turn. The step motor drives the driving gear and the driven gear is attached to a short piece of slotted rail that holds the elastomeric dish and the microphone. The microphone coincides with the rotation axis of the step motor driven system, so that it would have zero translation while the system is rotating. It takes the system around 40 seconds to make one full revolution. In order to let the system turn at a constant angular velocity and have less effect of the acceleration and deceleration from the step motor, we chose the starting point at a negative angle so that the step motor would reach its target speed before 0 angle. The sound pressure level was measured and logged into MATLAB© continuously; we cut the full revolution into 360 steps (every 1 degree), and took 50 waveforms out of each step to fit sinusoid of the same frequency to them. **Figure 16A** shows the polar response of the microphone without elastomeric dish; it shows the omnidirectionality of B & K 4961 microphone. **Figure 16B** shows the directionality of a 6 in elastomeric dish at its maximum curvature; it agrees with general directionality response for parabolic microphones, which the directionality around 0 angle increases with frequency.

4.4 Dish Gain

We measured the dish gain by sweeping through 51 frequencies from 2k Hz to 10k Hz linearly with a 2 second pause between each frequency, as the whole sweep took 357 seconds. We measured the sound pressure level at the focal point of the dish, and logged the data for both with dish case and without dish case. We fit the sinusoid of the 51 target frequencies to the data for both cases separately and then take the ratio of $\frac{Fitted\ amplitude\ (without\ dish)}{Fitted\ amplitude\ (with\ dish)}$ to calculate the experimental dish gain.

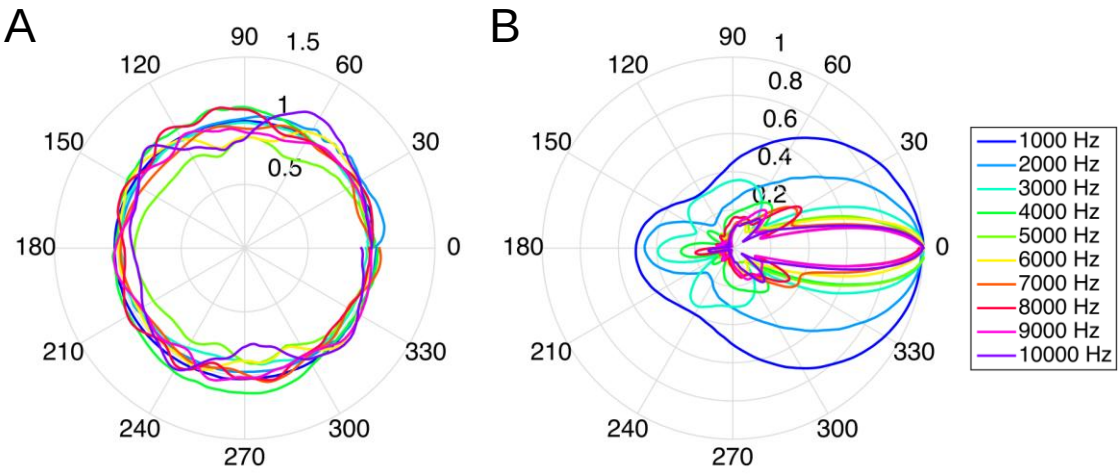

Figure 15. (A) Normalized polar response without the elastomeric dish (B) Normalized polar response with a 6 in elastomeric dish at maximum curvature

From Equation 2 we know that the dish gain increases with the diameter of the parabolic dish. A smaller dish normally increases the chances of loss and geometrical error, thus lowering the dish efficiency. We calculated the red curve as in **Figure 16** with 0.4 of the aperture efficiency e_A. It is noteworthy that the aperture efficiency for typical parabolic antennas is 0.55 to 0.70, as our elastomeric parabolic dish, considering the size of it is only 15 cm (6 in), the efficiency is quite remarkable.

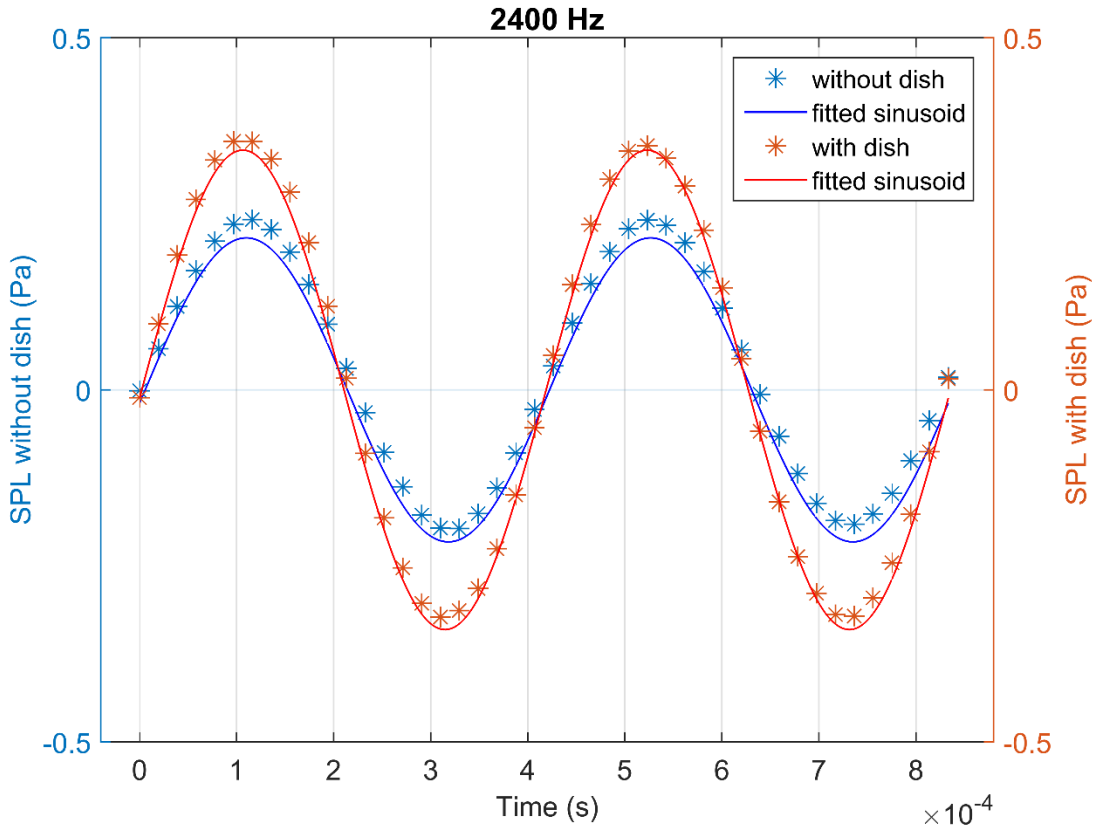

Figure 16. Time domain data with curve fitting result at 2400 Hz of the dish gain test

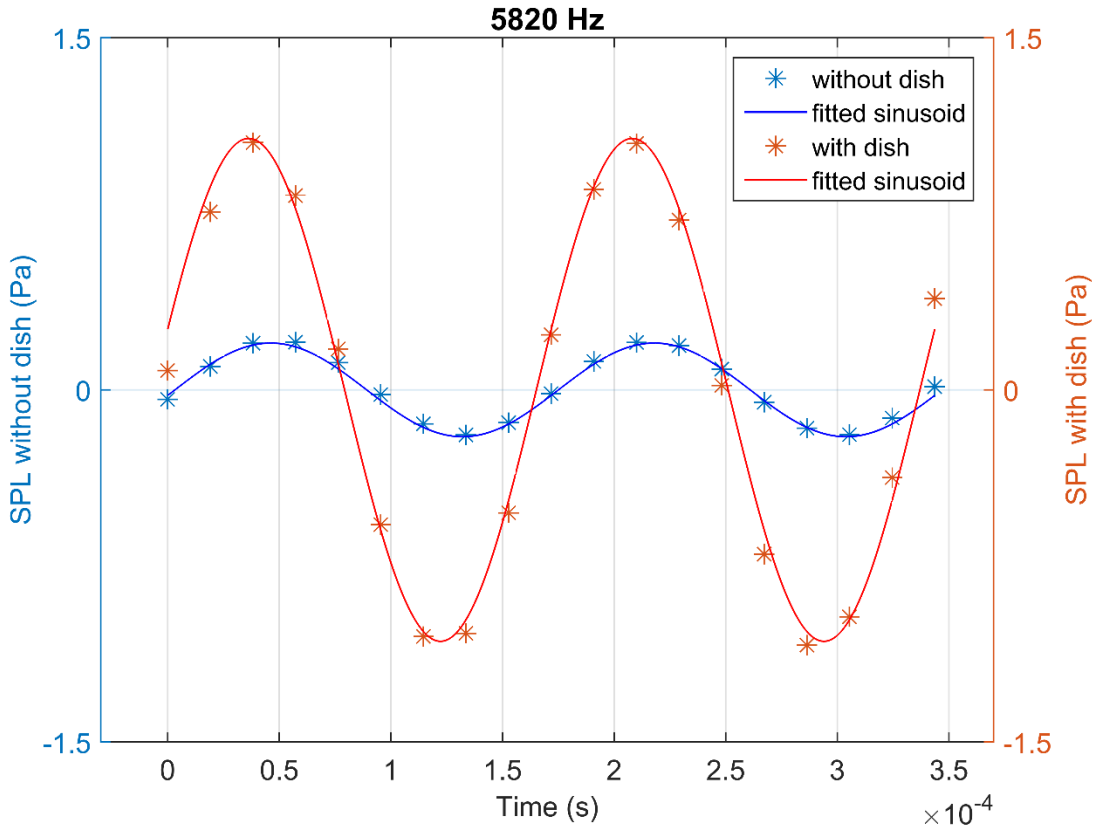

Figure 17. Time domain data with curve fitting result at 5820 Hz of the dish gain test

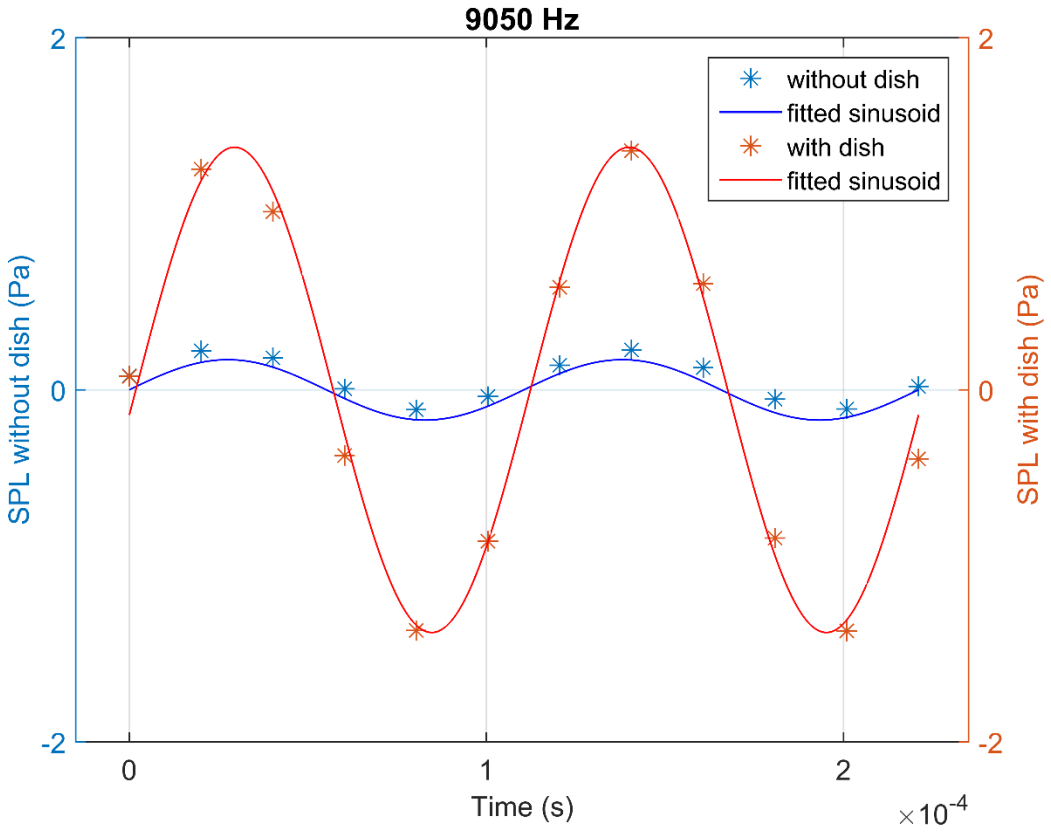

Figure 18. Time domain data with curve fitting result at 9050 Hz of the dish gain test

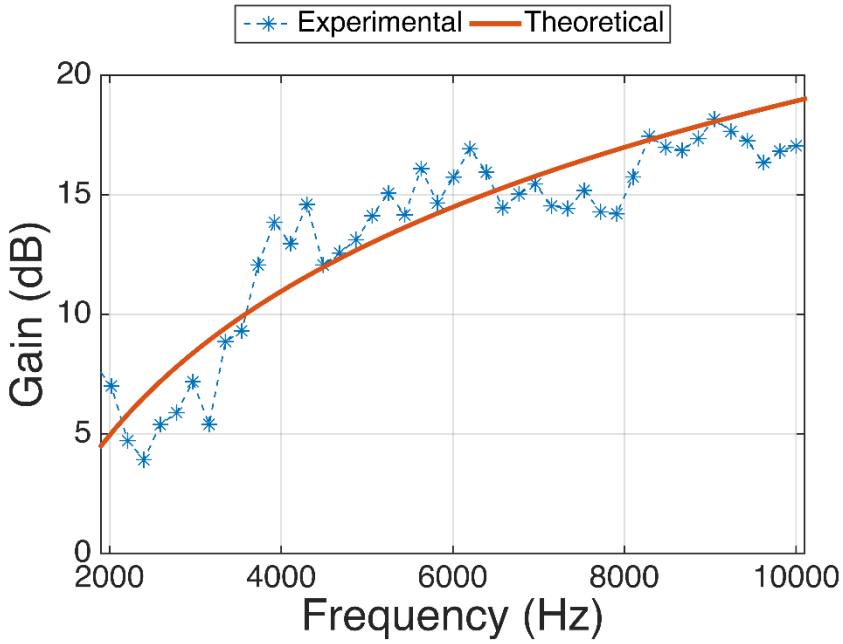

Figure 19. Dish gain of a 15 cm (6 in) elastomeric dish with theoretical dish gain curve at 40% efficiency

5. Conclusions

We characterized the reflection and absorption properties of Ecoflex 00-10 that we used in fabricating the elastomeric membranes for the parabolic microphones. The impedance tube experiments showed that Ecoflex 00-10, as one kind of silicone rubber material, possesses sufficient reflectivity for future exploration of soft acoustic manipulators.

We then fabricated elastomeric dishes with adjustable curvature controlled by applied vacuum and characterized the acoustic gain and directionality. The gain showed reasonable agreement with basic theoretical predictions at frequencies higher than the lower limiting frequency based on Rayleigh criterion. The disagreements might be due to the higher sensitivity of the dish to geometrical errors at higher frequency, and also the standing wave that built up in the room during experiments. The directionality results also showed reasonable agreements with general directionality response for parabolic microphones.

Although the elastomeric dish we created has been demonstrated to have similar dish gain and directionality response as regular hard dish-based parabolic microphone, the geometrical errors of the elastomeric dish are still limiting the efficiency of the dish. To optimize the design and create a better parabolic shape under certain pneumatic or hydraulic actuation, further study about the relationships between the cross sectional profile of the membrane and the deformation is required. We can also integrate an array of highly reflective surfaces or incorporate acoustic metamaterials to further increase the efficiency of the dish or even explore other potential functionalities of the system. These kinds of concepts, as a class of adjustable acoustic structures, might be useful to explore.

References:

[1] M. Konishi, “How the owl tracks its prey: experiments with trained barn owls reveal how their acute sense of hearing enables them to catch prey in the dark,” *Am. Sci.*, pp. 414–424, 1973.
[2] R. B. Coles, M. KONISHI, and J. D. Pettigrew, “Hearing and echolocation in the Australian grey swiftlet, Collocalia spodiopygia,” *J. Exp. Biol.*, vol. 129, no. 1, pp. 365–371, 1987.
[3] “RNC to Feature Unusual Forms of Sound,” *ABC News*, 07-Jan-2006. [Online]. Available: http://abcnews.go.com/Technology/story?id=99472&page=1. [Accessed: 29-Sep-2015].
[4] I. Jung, J. Xiao, V. Malyarchuk, C. Lu, M. Li, Z. Liu, J. Yoon, Y. Huang, and J. A. Rogers, “Dynamically tunable hemispherical electronic eye camera system with adjustable zoom capability,” *Proc. Natl. Acad. Sci.*, vol. 108, no. 5, pp. 1788–1793, Feb. 2011.
[5] C. Keplinger, J.-Y. Sun, C. C. Foo, P. Rothemund, G. M. Whitesides, and Z. Suo, “Stretchable, Transparent, Ionic Conductors,” *Science*, vol. 341, no. 6149, pp. 984–987, Aug. 2013.
[6] H. F. Olson and R. T. Beyer, “Acoustical engineering,” *Phys. Today*, vol. 10, p. 40, 1957.
[7] F. A. Everest, K. C. Pohlmann, and T. Books, *The master handbook of acoustics*, vol. 4. McGraw-Hill New York, 2001.
[8] F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides, “Soft Robotics for Chemists,” *Angew. Chem.*, vol. 123, no. 8, pp. 1930–1935, Feb. 2011.
[9] R. F. Shepherd, F. Ilievski, W. Choi, S. A. Morin, A. A. Stokes, A. D. Mazzeo, X. Chen, M. Wang, and G. M. Whitesides, “Multigait soft robot,” *Proc. Natl. Acad. Sci.*, vol. 108, no. 51, pp. 20400–20403, Dec. 2011.
[10] R. V. Martinez, J. L. Branch, C. R. Fish, L. Jin, R. F. Shepherd, R. M. D. Nunes, Z. Suo, and G. M. Whitesides, “Robotic Tentacles with Three-Dimensional Mobility Based on Flexible Elastomers,” *Adv. Mater.*, vol. 25, no. 2, pp. 205–212, Jan. 2013.
[11] ASTME1050, *Standard Test Method for Impedance and Absorption of Acoustical Material Using a Tube*. 2010.
[12] I. S. O. Standard, “10534-2: 1998 (E), Acoustics–determination of sound absorption coefficient and impedance in impedance tubes–Part 2: Transferfunction method,” *Int. Organ. Stand.*, 1998.
[13] E. Bickford, “Design of acoustic impedance analyzer,” 2015.

Appendices

Appendix A MATLAB code for impedance tube

```
clear all

close all


%% Code setup

load('low.mat');

freq_range='low';              % frequency range 'low' or 'high'

duration = 10;                 % duration of recording

temp = 21;                     % room tempreature

ap = 101.325;                  % atmospheric presssure, kPa

fs = 51200;                    % sampling rate


%% Hardware setup

% Discover Devices that Support Microphones

devices = daq.getDevices();

deviceInfo = devices(1)

% Create Session and Add Microphone Channel

s = daq.createSession('ni');

s.addAnalogInputChannel('cDAQ1Mod1', 1,'Microphone');

s.addAnalogInputChannel('cDAQ1Mod1', 2,'Microphone');

% Set Sensor Properties

s.Channels(1).Sensitivity = 0.0582598;
```

```
s.Channels(2).Sensitivity = 0.0578577;
s.Channels
% Configure And Start Acquisition
s.DurationInSeconds = duration ;
s.Rate = fs;
[data, time] = s.startForeground();
% Save Data
save('7_13_ecoflex.mat','data','time')


%% Define Constants:
% Measurement Positions
switch freq_range
   case 'low'
      s = 0.1175;    % microphone spacing
      x1 = 0.1947;   % distance between sample and the farther microphone
      freq_low_limit = 400;
      freq_high_limit = 1000;
   case 'high'
      s = 0.0302;    % microphone spacing high
      x1 = 0.1074;   % distance between sample and the farther microphone
      freq_low_limit = 800;
      freq_high_limit = 4800;
end
```

```
% Define Medium
rho = 1.29*(ap/101.325)*(273.5/(273.5+temp));   % density of air (kg/m^3) normally
1.21
c = 20.047*sqrt(273.15+temp);                   % speed of sound in air (m/s) normally 343
Zair = rho*c;                                   % characteristic impedance of air (kg/m^2/s)
% Fetch Data
ptch1=data(:,1);
ptch2=data(:,2);
t=time(:,1);


%% Calculation
% Transfer function
[H12,freq]=tfestimate(data(:,1),data(:,2),[],[],[],51200,'half');
% Coefficients
k = (2*pi*freq)/c;
R = (H12-exp(-1i.*k.*s))./(exp(1i.*k.*s) - H12) .*exp(2.*1i.*k.*x1);
alpha = 1 - abs(R).^2;       % Absorption coefficient
Zs = Zair.*((1+R)./(1-R));    % Surface impedance
Zs_n = ((1+R)./(1-R));       % Normalized Surface Impedance


%% Plot
% Transfer Function
figure(1)
```

```
semilogy(freq,abs(H12),'LineWidth',2);
xlim([freq_low_limit freq_high_limit]);
ylim([-10^2 10^2]);
grid on;
xlabel('Frequency (Hz)','FontSize',16);
ylabel('Transfer Function H_{12}','FontSize',16);
set(gca,'FontSize',14);
saveas(gcf,'tf.fig');
print('tf', '-dpng', '-r600');            %Save as PNG with 600 DPI

% Coefficients
figure(2)
semilogx(freq,alpha,'LineWidth',2);
xlim([freq_low_limit freq_high_limit]);
ylim([0 1]);
xlabel('Frequency (Hz)','FontSize',16);
ylabel('Absorption Coefficient','FontSize',16);
set(gca,'FontSize',14);
grid on;
saveas(gcf,'acoeff_low.fig');
print('acoeff', '-dpng', '-r600');        %Save as PNG with 600 DPI

figure(3)
```

```
semilogx(freq,abs(R),'LineWidth',2);
xlim([freq_low_limit freq_high_limit]);
ylim([0 1]);
xlabel('Frequency (Hz)','FontSize',16);
ylabel('Reflection Coefficient','FontSize',16);
set(gca,'FontSize',14);
grid on;
saveas(gcf,'rcoeff_low.fig');
print('rcoeff', '-dpng', '-r600');          %Save as PNG with 600 DPI

% Surface impedance
figure(4)
subplot(2,1,1)
plot(freq,real(Zs),'LineWidth',2);
xlim([freq_low_limit freq_high_limit]);
title('Surface Impedance ®C Real part','FontSize',16);
xlabel('Frequency (Hz)');
grid on;
subplot(2,1,2)
plot(freq,imag(Zs),'LineWidth',2);
xlim([freq_low_limit freq_high_limit]);
title('Surface Impedance ®C Imaginary part','FontSize',16);
xlabel('Frequency (Hz)');
```

```
grid on;


% Surface impedance ratio
figure(5)
subplot(2,1,1)
plot(freq,real(Zs_n),'LineWidth',2);
xlim([freq_low_limit freq_high_limit]);
title('Surface Impedance Ratio®C Real part','FontSize',16);
xlabel('Frequency (Hz)');
grid on;
subplot(2,1,2)
plot(freq,imag(Zs_n),'LineWidth',2);
xlim([freq_low_limit freq_high_limit]);
title('Surface Impedance Ratio ®C Imaginary part','FontSize',16);
xlabel('Frequency (Hz)');
grid on;
```

Appendix B MATLAB code for Dish Gain

```
clear all
close all


%% Setup
% Choose data files to read
filename_nodish = 'nodish.mat';
```

```
filename_concave = 'concave.mat';
% For corpping data
t_estimate = 4;  % estimate length of the pause before the first frequency
fs = 51200;
% Estimated noise level for the recognition of the signal starting point
noise_level = 0.1;
SNR_estimate = 5;
% Signal pattern
fq_low = 500;                          % start frequency
fq_high = 10000;                       % end frequency
fq_number = 51;                        % number of fq swept
fq_list = (linspace(fq_low,fq_high,fq_number))';
fq_duration = 2;                       % duration of each fq
pause_duration = 1;                    % pause duartion after each fq
number_delay = 0;                      % delay for locating the first peak
test_duration = fq_number*(fq_duration+pause_duration)
% Fitting
fq_delay = 0.5;                        % delay at the beginning of each fq
num_wave = 50;   % number of waveform taking into curve fitting for each fq
n_fitted_ch1 = [];
c_fitted_ch1 = [];


%% Fetch Data (no dish)
```

```
% Fetch no dish data
load(filename_nodish)
time_nodish = time(:,1);
data_nodish_ch1 = data(:,1);


%% Corp Data (no dish)
% Grab Ch1 and find peaks
peak_sig_nodish = data_nodish_ch1(1:t_estimate*fs,1);
[pks_nodish,locs_nodish] = findpeaks(peak_sig_nodish);
% Find the exact starting point
i =1;
while 1
   if pks_nodish(i) > SNR_estimate*noise_level
      break
   end
   i = i+1;
end
starting_point_nodish = locs_nodish(i)+number_delay;
end_point_nodish=test_duration*fs+starting_point_nodish;
starting_time_nodish = locs_nodish(i)/fs
end_time_nodish = end_point_nodish/fs
% Corp data
n_ch1 = data_nodish_ch1(starting_point_nodish:end_point_nodish,1);
```

```
sz_nodish = size(n_ch1(:,1));
% Zero time
n_time = time_nodish(starting_point_nodish:end_point_nodish,1)-...
    time_nodish(starting_point_nodish)*ones(sz_nodish);
% Plot time domain
figure
plot(n_time,n_ch1);
title('Time Domain (nodish)','FontSize',16);
xlabel('Time (s)','FontSize',16);
ylabel('Pressure (pa)','FontSize',16);
set(gca,'FontSize',14);
saveas(gcf,'nodish_time_domain.png');


%% Curve Fitting (no dish)
fq_count=linspace(0,fq_number-1,fq_number);
step=(fq_duration+pause_duration)*fs;
fq_start_pt=step*fq_count;


for i = 1:fq_number
    % locate 50 waveform
    n_start_pt=round(fq_start_pt(i)+fq_delay*fs);
    n_end_pt=n_start_pt+num_wave/fq_list(i)*fs;
    % fetch data
```

```
X = n_time(n_start_pt:n_end_pt,1);

Y1 = n_ch1(n_start_pt:n_end_pt,1);

Y0 = n_ch1((fq_start_pt(i)+1):(fq_start_pt(i)+fq_duration*fs),1);

time_start = n_time(round(n_start_pt));

time_end = n_time(round(n_end_pt));

x = linspace(time_start,time_end,1000000);

% curve fitting

fo = fitoptions('Method','NonlinearLeastSquares');

ft = fittype(' a1*sin(b1*x+c1)','problem','b1','options',fo);

[fn_ch1.fit,fn_ch1.G] = fit(X,Y1,ft,'problem',fq_list(i)*2*pi);

n_fitted_ch1 = [n_fitted_ch1;abs(fn_ch1.fit.a1)];

% FFT for Ch1

L1 = length(Y0);

NFFT = 2^nextpow2(L1);

Y0_fft = fft(Y0,NFFT)/L1;

f0 = fs/2*linspace(0,1,NFFT/2+1);

abs_Y0=2*abs(Y0_fft(1:NFFT/2+1));

% Plot

figure

subplot(2,1,1);


   plot(X,Y1,'.b',x,fn_ch1.fit.a1*sin(fn_ch1.fit.b1.*x+fn_ch1.fit.c1),'r','MarkerSize',14,'

   LineWidth',2);
```

```
    title([num2str(fq_list(i),'%.0f'),' Hz nodish Ch1'],'FontSize',16);
    xlim([time_start time_end]);
    grid on;
    set(gca,'FontSize',14);
    subplot(2,1,2);
    plot(f0,abs_Y0,'MarkerSize',14,'LineWidth',2)
    title('Amplitude Spectrum Ch1','FontSize',16)
    xlabel('Frequency (Hz)','FontSize',14)
    ylabel('|Y(f)|','FontSize',14)
    xlim([(1-fft_ww)*fq_list(i) (1+fft_ww)*fq_list(i)]);
    grid on;
    set(gca,'FontSize',14);
    saveas(gcf,['nodish_',num2str(fq_list(i),'%.0f'),' Hz.png']);
    close all;
end


%% Fetch Data (Concave)
% Fetch concave data
load(filename_concave)
time_concave = time(:,1);
data_concave_ch1 = data(:,1);


%% Corp Data (Concave)
```

```
% Grab Ch1 and find peaks
peak_sig_concave = data_concave_ch1(1:t_estimate*fs,1);
[pks_concave,locs_concave] = findpeaks(peak_sig_concave);
% Find the exact starting point
i =1;
while 1
   if pks_concave(i) > SNR_estimate*noise_level
      break
   end
   i = i+1;
end
starting_point_concave = locs_concave(i)+number_delay;
end_point_concave=test_duration*fs+starting_point_concave;
starting_time_concave = locs_concave(i)/fs
end_time_concave = end_point_concave/fs
% Corp data
c_ch1 = data_concave_ch1(starting_point_concave:end_point_concave,1);
sz_concave = size(c_ch1(:,1));
% Zero time
c_time = time_concave(starting_point_concave:end_point_concave,1)-...
   time_concave(starting_point_concave)*ones(sz_concave);
% Plot time domain
figure
```

```
plot(c_time,c_ch1);
title(['Time Domain (Concave)_',num2str(d,'%d')],'FontSize',16);
xlabel('Time (s)','FontSize',16);
ylabel('Pressure (pa)','FontSize',16);
set(gca,'FontSize',14);
saveas(gcf,['concave_time_domain_',num2str(d,'%d'),'.png']);
close all;

%% Curve Fitting (Concave)
for i = 1:fq_number
    % locate 50 waveform
    c_start_pt=round(fq_start_pt(i)+fq_delay*fs);
    c_end_pt=c_start_pt+num_wave/fq_list(i)*fs;
    % fetch data
    X = c_time(c_start_pt:c_end_pt,1);
    Y1 = c_ch1(c_start_pt:c_end_pt,1);
    Y0 = n_ch1((fq_start_pt(i)+1):(fq_start_pt(i)+fq_duration*fs),1);
    time_start = c_time(round(c_start_pt));
    time_end = c_time(round(c_end_pt));
    x = linspace(time_start,time_end,1000000);
    % curve fitting
    fo = fitoptions('Method','NonlinearLeastSquares');
    ft = fittype(' a1*sin(b1*x+c1)','problem','b1','options',fo);
```

```
[fc_ch1.fit,fc_ch1.G] = fit(X,Y1,ft,'problem',fq_list(i)*2*pi);
c_fitted_ch1(i,d) = abs(fc_ch1.fit.a1);
% FFT for Ch1
L1 = length(Y0);
NFFT = 2^nextpow2(L1);
Y0_fft = fft(Y0,NFFT)/L1;
f0 = fs/2*linspace(0,1,NFFT/2+1);
abs_Y0=2*abs(Y0_fft(1:NFFT/2+1));
% Plot
figure
    subplot(2,1,1);
    plot(X,Y1,'.b',x,fc_ch1.fit.a1*sin(fc_ch1.fit.b1.*x+fc_ch1.fit.c1),'r','MarkerSize',14,'
    LineWidth',2);
    title([num2str(fq_list(i),'%.0f'),' Hz concave
    Ch1_',num2str(d,'%d')],'FontSize',16);
xlim([time_start time_end]);
grid on;
set(gca,'FontSize',14);
subplot(2,1,2);
plot(f0,abs_Y0,'MarkerSize',14,'LineWidth',2)
title(['Amplitude Spectrum Ch1_',num2str(d,'%d')],'FontSize',16)
xlabel('Frequency (Hz)','FontSize',14)
ylabel('|Y(f)|','FontSize',14)
```

```
    xlim([(1-fft_ww)*fq_list(i) (1+fft_ww)*fq_list(i)]);
    grid on;
    set(gca,'FontSize',14);
        saveas(gcf,['concave_',num2str(d,'%d'),'_',num2str(fq_list(i),'%.0f'),'Hz.png']);
    close all;
end

%% Dish gain
figure
% Experimental
dish_gain(:,k) = 20.*log10(c_fitted_ch1./n_fitted_ch1);
plot(fq_list,dish_gain,'-x','MarkerSize',8,'LineWidth',2);
xlabel('Frequency (Hz)');
ylabel('Gain (dB)');
grid on;
set(gca,'FontSize',14);
hold on
% Theoretical
ff = linspace(fq_low,fq_high,50);
d2 = 0.1524;                                % diameter of the dish
GdB1 = 10.*log10(0.4*(d2*pi/343)^2.*ff.^2);
xlim([fq_low fq_high]);
plot(ff,GdB1,'-r','LineWidth',2);
```

```
saveas(gcf,'Gain.png');
legend('Experiment','Theoretical','Location','northoutside','Orientation','horizontal')
print('gain_6s_40in', '-dpng', '-r600'); %Save as PNG with 600 DPI
```

Appendix C Arduino Code for Step Motor

```
#include <IRremote.h>
#include <IRremoteInt.h>
#include <AccelStepper.h>

int RECV_PIN = 11;
long on1  = 0x00FFA25D;
long off1 = 0x00FFE01F;
IRrecv irrecv(RECV_PIN);
decode_results results;

#define HALFSTEP 8
// Motor pin definitions
#define motorPin1  3     // IN1 on the ULN2003 driver 1
#define motorPin2  4     // IN2 on the ULN2003 driver 1
#define motorPin3  5     // IN3 on the ULN2003 driver 1
#define motorPin4  6     // IN4 on the ULN2003 driver 1

void dump(decode_results *results) {
```

```
int count = results->rawlen;
if (results->decode_type == UNKNOWN)
  {
   Serial.println("Could not decode message");
  }
else
 {
  if (results->decode_type == NEC)
    {
     Serial.print("Decoded NEC: ");
    }
  else if (results->decode_type == SONY)
    {
     Serial.print("Decoded SONY: ");
    }
  else if (results->decode_type == RC5)
    {
     Serial.print("Decoded RC5: ");
    }
  else if (results->decode_type == RC6)
    {
     Serial.print("Decoded RC6: ");
    }
```

```
  Serial.print(results->value, HEX);
  Serial.print(" (");
  Serial.print(results->bits, DEC);
  Serial.println(" bits)");
 }
  Serial.print("Raw (");
  Serial.print(count, DEC);
  Serial.print("): ");

for (int i = 0; i < count; i++)
  {
   if ((i % 2) == 1) {
   Serial.print(results->rawbuf[i]*USECPERTICK, DEC);
  }
 else
  {
   Serial.print(-(int)results->rawbuf[i]*USECPERTICK, DEC);
  }
 Serial.print(" ");
  }
   Serial.println("");
  }
```

```
AccelStepper stepper1(HALFSTEP, motorPin1, motorPin3, motorPin2, motorPin4);
// Initialize with pin sequence IN1-IN3-IN2-IN4 for using the AccelStepper with 28BYJ-
48

void setup()
 {
  pinMode(RECV_PIN, INPUT);
  stepper1.setMaxSpeed(1000.0);
  stepper1.setAcceleration(100.0);
  stepper1.setSpeed(200);
  stepper1.moveTo(8000);
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
 }

int on = 0;
unsigned long last = millis();

void loop()
{
  if (irrecv.decode(&results))
   {
    // If it's been at least 1/4 second since the last
```

```
    // IR received, toggle the relay
    if (millis() - last > 250)
      {
       on = !on;
//       digitalWrite(8, on ? HIGH : LOW);
       digitalWrite(13, on ? HIGH : LOW);
       dump(&results);
      }
    if (results.value == on1 )
   //   void loop() {
          //
          //    //Change direction when the stepper reaches the target position
          //    if (stepper1.distanceToGo() == 0) {
          //      stepper1.moveTo(-stepper1.currentPosition());
          //    }
                stepper1.run();
     //             };


//    if (results.value == off1 )
//       digitalWrite(LED1, LOW);
    last = millis();
  }
}
```