A.U.R.A.L.
Audible Ultrasonic Realistic Artificial Larynx

An Audible Ultrasound Electrolarynx

By Patrick M. Mills

B.S. in Engineering, June 1994, Swarthmore College
M.S. in Electrical and Computer Engineering, May 1996, University of South Carolina

A Dissertation submitted to

The Faculty of
The School of Engineering and Applied Science
of the George Washington University
in partial fulfillment of the requirement
for the degree of Doctor of Philosophy

January 31, 2015

Dissertation directed by

Jason M. Zara
Associate Professor of Engineering and Applied Science

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

The School of Engineering and Applied Science of The George Washington University certifies that Patrick M. Mills has passed the Final Examination for the degree of Doctor of Philosophy as of March 26, 2014. This is the final and approved form of the dissertation.

A.U.R.A.L.
Audible Ultrasonic Realistic Artificial Larynx
An Audible Ultrasound Electrolarynx

Patrick M. Mills

Dissertation Research Committee:

Jason M. Zara, Associate Professor of Engineering and Applied Science,
Dissertation Director

Matthew Kay, Assistant Professor of Engineering and Applied Science,
Committee Member

David J. Nagel, Research Professor of Engineering and Applied Science,
Committee Member

Michael W. Plesniak, Professor of Mechanical and Aerospace Engineering,
Committee Member

Nader Sadeghi, Professor of Surgery,
Committee Member

Dedication

for

Samantha & Alexander

Thank you for your:

Support

Encouragement

Patience

Love

hours of listening to technical jargon,

reading & re-reading papers,

sitting through my presentations again and again,

helping me through the ups & downs of the whole process.

I would not have made it through this without you both!

Acknowledgements

None of this would have been possible without the help of family, friends, teachers, and even strangers. I would like to thank all those people who have never even met me, yet were willing to help by sharing their time & resources. In particular, I would like to thank Ola Vikholt for sharing his acoustic FDTD code, Adam Sawicki for sharing his MRI renderer code, and the UT Austin team for sharing the AustinMan model.

My advisor, Dr. Zara, put up with me for over a decade. During that time, I worked in his lab, ran a software development business full-time, had a child, and switched my dissertation topic SIX times. He never gave up on me, and always sat back down to listen to the next idea. And when the time came that I was ready to throw in the towel, he encouraged me to continue. During three critical moments, he stood up before his peers and defended my work – for that I am grateful. I would also like to thank my committee for their time and guidance.

I would like to thank Dr. Matthew Alford of the Scripps Institution of Oceanography, UC San Diego. As one of my undergraduate roommates and best friends, I knew he would help. However, as usual he went above and beyond – putting aside his own pressing work & family demands – to review my journal papers multiple times and provide valuable suggestions that ultimately led to my completing my publication requirement.

My family also helped; everyone chipped in. My mom, Dr. Elizabeth Mills of Davidson College, proofed my dissertation and papers. My younger brother, Thomas Mills, helped me with the statistic parts of my dissertation; his language appears in a few places in this document – Thanks T-Bird! And my older brother, William A. Mills, Jr.,

M.D., for his help with medical terms, potential resources, and helping me suggest appropriate reviewers for my journal papers.

And of course my immediate family: my wife Samantha and son Alexander. They put up with a decade of postponed/missed vacations, reading my dry technical writing, and listening to my presentations over and over. I love them very much.

Thanks to you all, this challenging endeavor is finally over.

Abstract

AN AUDIBLE ULTRASOUND ELECTROLARYNX

Every year, many thousands of people worldwide lose the ability to speak due to receiving a laryngectomy, typically for treatment of cancer. At some point in their recovery, most will use an electrolarynx to recover their ability to speak. Typical electrolarynxes utilize a piston to strike a disc pressed to the patient's neck which delivers a pressure wave into the soft tissue. This pressure wave mechanically couples with the vocal tract and generates the fundamental frequency necessary for creating vowels without which speech is not possible.

Commonly available electrolarynxes suffer from poor frequency control due to the nonlinear character of their impulse driver. They also create a great deal of "self-noise" which is distracting to listeners and makes using voice communication systems difficult.

We propose a novel electrolarynx implementation which utilizes two interfering ultrasonic waves to generate a fundamental frequency in the vocal tract required for speech restoration. The device is light weight, compact, inexpensive, and offers excellent control of all aspects of the output waveform. In addition, as the primary waveforms are above human hearing, there is little "self-noise" that can be heard by listeners and most communications devices filter such noise as part of their standard digitization process.

This device offers the potential to greatly improve the lives of those who have lost their voices and must rely on technology to allow them to communicate in the most efficient manner.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ARF	Acoustic radiation force
CUDA	Compute Unified Device Architecture
dB	Decibel
DTFT	Discrete-time Fourier transform
EL	Electrolayrnx
EMG	Electromyography
FDTD	Finite-difference time-domain
FEM	Finite element method
FFT	Fast Fourier transform
GPU	Graphics Processing Unit
Hz	Hertz
LFUS	Low frequency ultrasound
PCB	Printed circuit board
PML	Perfectly matched layer
PZT	Lead Zirconate Titanate
SPL	Sound pressure level
STFT	Short-time Fourier transform
TE	Tracheoesophageal
VA	Vibroacoustography

Glossary of Terms

Anterior – The front of a body; also referred to as ventral.

Coronal plane – A plane that divides the body into an anterior (font) and posterior (back) portions; also known as the frontal plane. In our simulations, it is the XZ plane.

Electrolayrnx – A medical device used to stimulate voicing sounds in someone who has undergone a laryngectomy.

Fourier transform – A mathematical transformation that converts from the time domain to the frequency domain

Impedance – Resistance to change. In mechanical systems, the ratio of force to velocity. Acoustic impedance is the ratio of pressure to flow rate through a surface.

Laryngectomy – The total or partial removal of the larynx; typically due to cancer.

Posterior – The back of a body; also referred to as dorsal.

Sagittal plane – A plane that divides the body into right and left sides; also known as the lateral plane. In our simulations, it is the YZ plane.

Spectrogram – A graphical representation of the frequency content of a signal over time.

Transverse plane – A plane that divides the body into superior (upper) and inferior (lower) portions; also known as the horizontal or axial plane. In our simulations, it is the XY plane.

Ultrasound – Sound waves with a frequency above the human audible range; typically any sound wave with a frequency above 20 KHz.

Voxel – A volumetric pixel is a three-dimensional pixel.

1. Introduction

The first artificial larynx was a pneumatic device produced in 1859 by Johann Nepamuk Czermak of Pesth to treat complete laryngeal stenosis. The first human laryngectomy was performed by Patrick Watson of Edinburgh in 1866; the first use for cancer was in 1873 by Theodor Billroth who also installed a pneumatic powered reed device for restoring the patient's voice. The first true electrolarynx was developed by Themistocles Gluck in 1910; it used an Edison style phonograph with a recording of a vowel sound which was played using a telephone receiver. [1] [2] Over the years improvements were made to Gluck's basic design; however, it wasn't until the invention of the transistor in the 1950's that the design evolved into the piston style device still used today. The design still utilized a telephone speaker derived electromechanical driver but attached a rigid disk to impart vibrations into the vocal tract. Modern implementations have the driver strike a flat disk which is pressed to the neck; this results in nonlinear impulse excitations. This design imparts higher amplitude signals into the vocal tract, but due to the nonlinear transducer, the characteristics – such as spectral envelope, wave shape, and dispersion – which can be easily controlled are limited. [3] [4]

1.1. Motivation

Modern electrolarynxes are light (≈ 4oz + 2oz battery), fit in the palm of your hand, have a rechargeable battery (≈ 4.5 hours of continual activation), have an adjustable volume and fundamental frequency, and have either multiple buttons or a pressure sensitive button to offer variable pitch. They continue to evolve as researchers investigate initiation and variable pitch via EMG of the laryngeal nerve and wireless

controllers, use of PZT transducers, reduction/cancelling of noise produced by the device itself, force sensing transducers, and "wearability."

However, even with all this ongoing research, the core design of the electrolarynx has not changed in over sixty years. Popular external commercial models are about 12 cm tall by 3 cm in diameter and designed to be held in your hand. About 4 cm of the device is for the battery, 3 cm for the transducer and the other 5 cm is packaging and controls. The solution presented here was inspired by an article about Joseph Pompei's Audio Spotlight after watching a documentary on the difficulties faced by laryngectomees. Pompei used a large ultrasonic parametric array to generate modulated ultrasound which self-demodulated based on nonlinear air effects and interaction with the ear canal. [5] While the transducers and arrangement used by Pompei are impractical for a small device, alternatives exist which would enable a lightweight, handheld or hands-free device which would have little or no external noise leakage.

1.2. Objective

We propose a using a small device with two PZT ultrasonic transducers to generate acoustic waves near 40 kHz. With proper timing, alignment, and amplitude, the waves will penetrate the skin, and create a pulsating interference pattern based on the difference between the two input waveforms. The pressure generated by this beat frequency is mechanically coupled to the vocal tract in the same way a standard electrolarynx functions. When the difference is between around 100 – 600 Hz, the vibrating tissue will generate an acceptable fundamental frequency used to form vowels, thus allowing speech. While there will be external sound leakage as with classic electrolarynxes, the leakage will be well above the human hearing range thus eliminating

one of the fundamental issues with piston driven electrolarynxes. As difference waves are used, varying the frequency is quite straightforward.

To achieve our goal, we will implement a simulator allowing various scenarios to be tested to determine the best location, signal type, frequencies and resultant output. From this information we will optimize a physical unit which can be tested to validate our hypothesis that this implementation will offer many benefits over the classical design.

Ultimately, we hope this device will be used to help those who have had a laryngectomy live a more fulfilled life. With an optimal MEMS parametric transducer, a commercialized model would be the size of a choker or neck-wire necklace. The ultrasonic transducer can also be used to locate the closest point of the vocal tract and then the waves can be steered to that location for maximum power transfer. Addition of an EMG sensor would allow completely hands-free operation, thus restoring a lost voice and freeing both hands for normal use.

2. The Fundamentals of Speech

Human social interaction depends heavily upon speech. Speech is our most efficient communication modality; other modalities require more concentration, restrict movement, and cause body strain. During spontaneous speech, an average of 2.0 to 3.6 words per second can be communicated. Skilled typists can type around 1.6 to 2.5 words per second, but only when typing prepared text. A skilled typist averages only 0.3 words per second when typing spontaneous text or when problem solving. This productivity is comparable to unskilled typists who type only 0.2 to 0.4 words per second under optimal conditions. The average speed of handwriting text is around 0.4 words per second. [6] Silent reading can achieve 2.5 to 9.8 words per second, but to achieve the higher rates with high retention, the reader must concentrate only on reading. [7]

Voice-only communication has been shown to be second only to a combination of modalities for problem solving. [8] Unfortunately each year about 3,000 - 5,000 people in the United States undergo a surgical procedure called a laryngectomy where their larynx, including the vocal folds and supporting cartilage and muscles, is removed. [9] The usual cause for the operation is cancer. While the rate of laryngeal cancer in the United States is reducing mainly due to reduction in tobacco use, and the rate of total laryngectomy is reducing even faster due to non-surgical treatments, there are still a substantial number of affected patients; in other countries the incidence of laryngeal cancer is still increasing. The ultimate result of a laryngectomy is the loss of the ability to created “voiced” sounds without a prosthesis or special biological manipulations.

2.1. Anatomy of the Larynx and Trachea

Figure 2-1 shows the human vocal system which consists of the vocal folds (more commonly referred to as the vocal cords), vocal tract (upper trachea, pharynx, tongue, and oral cavity), teeth, and lips. The vocal folds are the sound source and the vocal tract is a filter which spectrally shapes the sound into the voice. Typically the vocal folds vibrate at a fundamental frequency of around 120 ± 30 Hz for men and 190 ± 90 Hz for women. [10] Stemple et al. in a review of published research show that females have a mean fundamental frequency of 192 Hz with a range of 137 - 634 Hz and men have a mean of 106 Hz with a range of 77 – 482 Hz; children have a higher fundamental frequency with boys almost double adult males. [11] In addition to the fundamental frequency, Formby and Monsen point out that the harmonics of the fundamental frequency are present in normal speech providing important peaks in the spectra. [12]

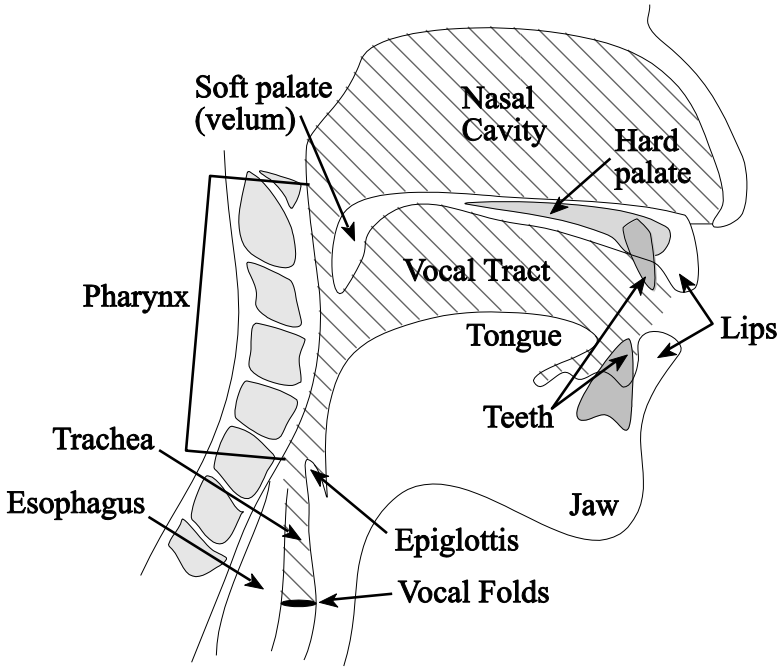


Figure 2-1: Human Vocal System [13]

The larynx is located between the base of the tongue and the top of the trachea and just in front of the C3-C6 vertebrae depending on age and sex. The larynx is protected by the Thyroid cartilage, a triangular shaped bony plate covering the front; the cricoid cartilage forms a ring around the base of the larynx and is thicker and stronger than the Thyroid cartilage. Located within the larynx are the vocal folds. Figure 2-2 shows several views of the larynx highlighting its structures and location. In men, an average larynx is about 43 mm coronal x 36 mm sagittal x 44 mm long and 41 mm coronal x 26 mm sagittal x 36 mm long for women. [14]

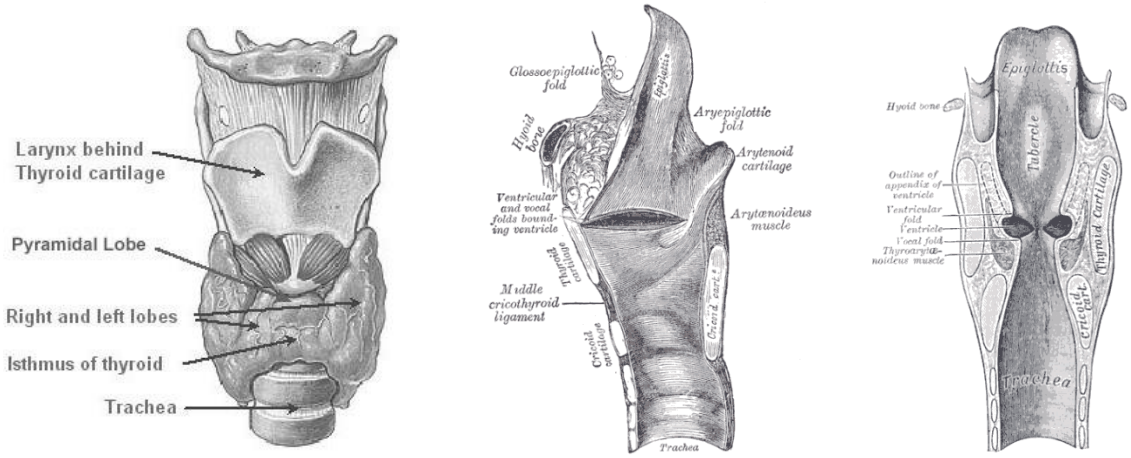


Figure 2-2: The Larynx [15] [14]

The trachea begins at the base of the larynx and continues down until it splits into the bronchi. The trachea contains multiple cartilage rings for stability that extend about 80% around; the rest of the structure is composed of smooth muscle. [16] A normal trachea has an elliptical shape; for men between 20-79 years old an average size diameter for sagittal and coronal axes are 27 mm x 25 mm (± 3 mm), for women 23 mm x 21 mm (± 2.5 mm). [17] The trachea is about 115 mm long. [14]

The lining of both the larynx and trachea are made up of layers of cells. The layer of cells exposed to the airway is the ciliated epithelium which contains about 200 cilia

per cell oscillating at 10-20 Hz towards the pharynx. Next is a submucous layer which along with goblet cells in the epithelial lining produces the mucous coating the outer epithelium. Layers of fiber separate the outer layers and connect to cartilage or muscle. Figure 2-3 shows a section of the mucous membrane.

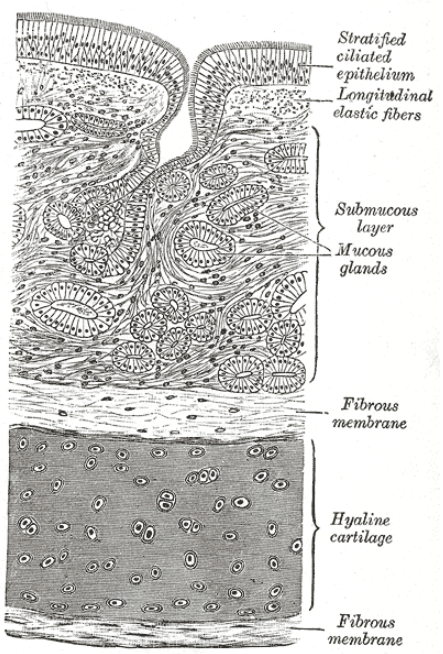


Figure 2-3: Cellular Section [14]

In our device, we will attempt to create vibrations in the outer epithelium; however, we may take advantage of the outer fibrous membrane to act like a taught spring thus allowing the entire submucosal layer to oscillate in a limited region of activation. The potential for oscillations to travel directly down the cilia into the airway is also an intriguing possibility especially for higher frequency harmonics although we anticipate the amplitude would be quite small.

2.2. Characterization of Speech

Speech can be broken down into several competing components: allophones, phonemes, diphones, syllables, or words. [6] Each representation has advantages and disadvantages. "Phone" denotes a minimal unit of speech sound. However, identifying an individual phone is practically impossible due to an anticipation effect which causes phones to overlap.

Researchers use allophones to represent the set of phones containing the same information content. Allophones can be identified reliably, but require complex, time consuming procedures. Also useful is the fact that allophones can be employed to identify word boundaries. However, like the phone, the allophone suffers from an anticipation or coarticulation effect. An even greater disadvantage is the large number of allophones that can be contained in a given language.

Phonemes are the collection of allophones that operate similarly in a language. Phonemes have an advantage over allophones in that the number of distinctive phonemes is quite small. However, phonemes are not easy to distinguish acoustically, overlap each other, and require complicated processing procedures. The 44 phonemes shown in Table 2-1 belong to the International Phonetic Alphabet (IPA). The symbol used for each phoneme is the most commonly accepted symbol, but others are sometimes used.

A diphone is a transitional sound identified by segmenting adjacent phones at their steady-state centers. By their very nature, diphones include transitional information that can be useful in identification. Like allophones, the number of diphones in a given language can be quite large and require a unique set of phonological rules for processing.

A syllable is basically "a vowel nucleus and its functionally related neighboring consonants."[1] Syllables are relatively easy to identify in the acoustic stream. As a bonus, many rules developed for dealing with phonemes are easily extended for use with syllables. The major problem with syllables is the difficulty in identifying boundaries. Again, as with phonemes and diphones, the number of syllables in a given language can be unmanageably large.

Phoneme	Example	Phoneme	Example
æ	pat	ŋ	sing
e	pay	o	toe
ə	about	U	book
a:	father	a	pot
ɛr	care	u	boot
b	bet	ɔ	bought
tʃ	church	aU	out
d	debt	ɔI	boy
ɛ	bet	p	pet
i	bee	ɹ	rent
f	fire	s	sat
g	get	ʃ	shut
h	hat	ɵ	thing
I	bit	t	ten
aI	by	ʊ	that
ir	pier	ʌ	but
ər	butter	ʒr	term
d	judge	v	vat
k	kit	w	wit
l	let	y	you
m	met	z	zoo
n	net	ʒ	azure

Table 2-1: International Phonetic Alphabet

[1] Lea, p. 129.

Finally, every speaker in a language knows instinctively what a word is, yet it is difficult to define phonologically. However, a word is the smallest unit of information that communicates a complete message.

In order for a spoken language to convey information, it must consist of a finite number of distinguishable and mutually exclusive sounds. [7] The following features can be used to identify a spoken sound regardless of the unit of identification. However, researchers have primarily used these features when identifying phonemes; Chomsky and Halle defined them for this reason in 1968. Due to the variance in the speech waveform, the classifications become less useful as the unit of identification becomes larger. The features are generally grouped into mutually exclusive groups. But as with many other parts of the sound waveform, absolute distinctions are difficult. Used together, the groups can accurately define a sound unit.

Voiced vs. Unvoiced:

Voiced sounds are caused as air pressure pushes the vocal folds open and causes them to vibrate. The sound produced has a pitch or fundamental frequency that is directly related to the frequency of vibration. The peak amplitude of a voiced sound is much higher than an unvoiced sound. Unvoiced sounds occur when the vocal folds are held open allowing air to pass through unaffected. The two fundamental unvoiced sounds are the *plosive* and the *fricative*. A plosive is generated by a build-up of air behind the lips which is rapidly expelled. A fricative is generated by a turbulent airflow through a constriction such as the teeth. [18]

Vocalic vs. Nonvocalic:

Vocalic sounds have a sharply defined formant structure. The formant is a band of high energy concentrated in a specific frequency range. Formants are generally found in vowels. Nonvocalic sounds lack a defined formant structure.

Consonant vs. Nonconsonant:

Consonant sounds have high total energy, while nonconsonant sounds have a lower total energy.

Tense vs. Lax:

Tense sounds have high total energy over a relatively long period of time in a wide frequency band. Lax sounds have lower total energy concentrated in a shorter time period in a tight frequency band.

Nasal vs. Oral:

Nasal sounds occur when the nasal passage is used as an auxiliary acoustic tube. These sounds contain a wide frequency spread and a reduction in the intensity of formants. Oral sounds contain a more defined frequency range and usually high intensity formants.

Strident vs. Mellow:

Strident sounds contain higher intensity noise than mellow sounds.

Grave vs. Acute:

Grave sounds have a higher concentration of energy in the lower frequencies, while acute sounds have a higher concentration of energy in the upper frequencies of the spectrum.

Compact vs. Diffuse:

Compact sounds contain a high concentration of energy in a narrow region of the spectrum. Diffuse sounds spread their energy across a wider region.

Flat vs. Plain:

Flat sounds are characterized by a weakening of the higher frequency components. Plain sounds contain no such weakening.

Figure 2-4 and Figure 2-5 show an example of the speech waveform for the word "recognition." The word was sampled at 11 kHz and contains 7,465 signed 8-bit samples. The word is easily segmented into syllables as rec•og•ni•tion, or as time periods 1-3, 4-5, 6-7, 8-10. When segmented into phonemes, the word is represented as ɹ ɛ k ə g n I ʃ ə n, with each phoneme taking one time period [1, 10].

Each phoneme can be characterized by its distinctive features. The phoneme ɹ is voiced, consonant, oral, acute, and flat. The phoneme ɛ is voiced, nonconsonant, lax, acute, and compact. The phoneme k is unvoiced, nonvocalic, consonant, tense, mellow, and compact. The phoneme ə is voiced, nonconsonant, tense, acute, diffuse, and flat. The phoneme g is unvoiced, nonvocalic, consonant, lax, mellow, and compact. The

phoneme n is voiced, consonant, nasal, acute, and diffuse. The phoneme I is voiced, nonconsonant, lax, acute, and diffuse. The phoneme ʃ is unvoiced, vocalic, consonant, tense, acute, and compact.

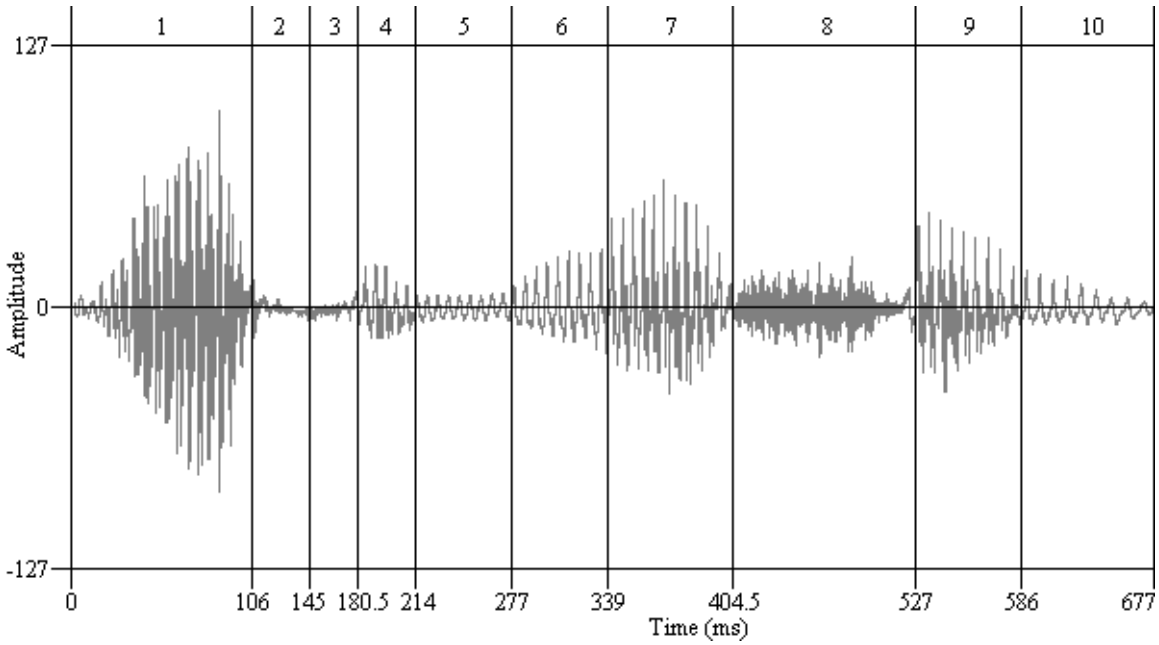


Figure 2-4: Amplitude Graph of the Word "recognition" [13]

There are two ə phonemes in the word (segments 4 and 9 in Figure 2-4 and Figure 2-5 – denoted p_4 and p_9), and the comparison is enlightening. Notice the amplitude difference; p_4 has a lower total energy content than p_9. This is a prosodic effect caused by the natural stress positions in the word. Also p_9 contains more high frequency components than p_4. The ə phoneme is a flat sound; therefore, the higher frequency components in p_9 are actually carried over from the preceding ʃ phoneme. This carryover is called a coarticulation effect. There are also two n phonemes (denoted p_6 and p_{10}). Again, the amplitude difference is caused by prosodic effects. Notice that the

frequency components are quite similar, since the preceding phonemes are either unvoiced or contain similar frequency components.

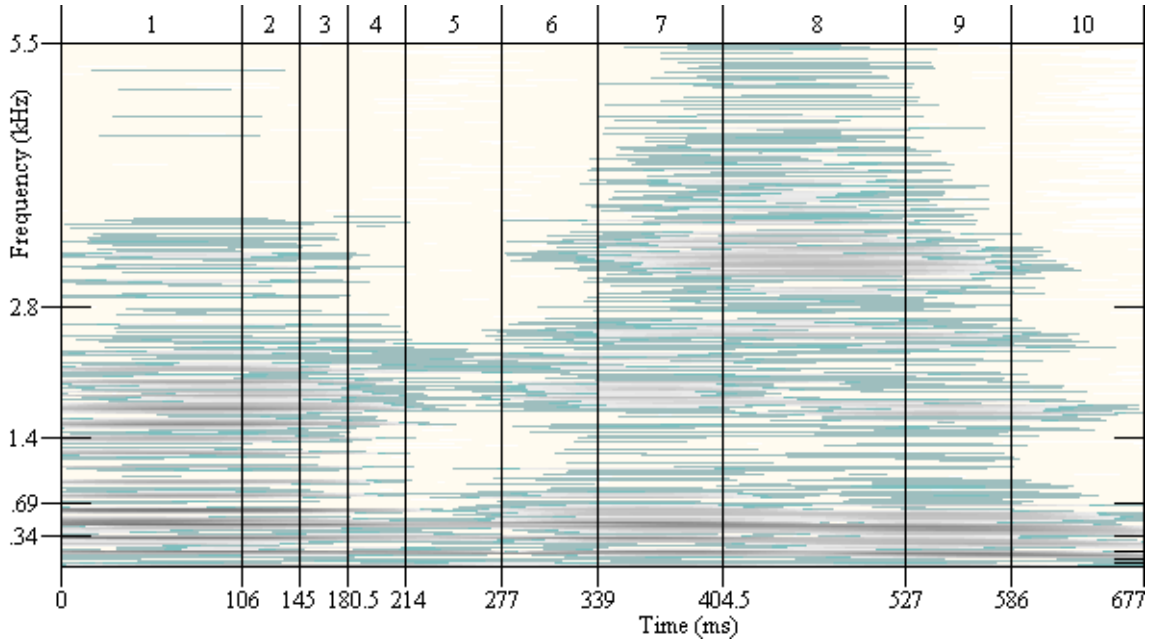


Figure 2-5: Spectrogram of the Word "recognition" [13]

These classifications and the analysis of them will be extremely useful once our device is ready for testing so that the driving inputs can be tuned to produce the best possible outputs corresponding to the user's original speech waveform and for analyzing the clarity of their resultant speech.

2.3. Laryngectomy

During a laryngectomy, bone, cartilage and tissue in the neck is removed. With the patient under anesthesia, the surgeon begins with a skin incision below the mastoid tip to the trapezius. The strap muscles are transected, the thyroid isthmus is divided dissecting the thyroid gland away from the trachea, and the tracheostoma is cut. Dissection of the trachea occurs below the tracheal ring when possible to preserve the

soft tissue for stoma closure. The inferior pedicle of ipsilateral thyroid lobe and ligate vessels are isolated; when possible the pedicled inferior parathyroid is preserved and recurrent laryngeal nerve relocated after transection. The constrictor muscles are detached from the thyroid ala and the suprahyoid musculature is separated from the superior border of the hyoid bone. The larynx is separated at the vallecula (just below the epiglottis), then separated by cutting along the avascular plane between the esophagus and trachea allowing the larynx to be removed. The openings that connected to the larynx are sealed, the tracheostoma is sutured in place, and the skin is sutured. [19]

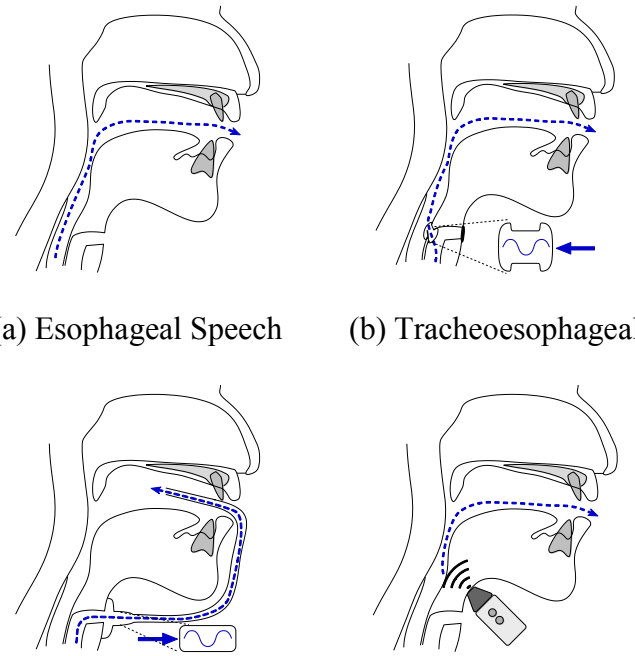
A total laryngectomy is a serious surgery requiring hospitalization for about a week with a feeding tube. After initial recovery, additional surgeries and treatments (such as radiation) may be required; in addition, psychological counseling may be required as many patients suffer depression. Radiation therapy usually results in fibrosis and edema which hardens the neck tissue; the effects of radiation therapy eventually subside and the neck tissue softens. [20] Fortunately, in most cases, a laryngectomee still has a functioning vocal tract and vocalization should be possible as long as a suitable sound source is provided. [21] As noted at the beginning of the chapter, the number of total laryngectomies performed per year in the United States has been declining over the past decade; even so, thousands are still performed every year and the procedure is performed in other countries as well. A Scottish survey noted that the rates of laryngeal cancer from 1980 – 2002 were still increasing even as the number of laryngectomies decreased; the survey also questioned the efficacy of organ preservation citing reduced survival rates in North American where the practice originated. [22] Servox®, a maker of electrolarynx devices notes that they have over 100,000 users of just one model of

their device and there are many other manufacturers of artificial larynxes. [23] Given these number of affected patients, an artificial larynx is still a necessary device with a worldwide user base.

2.4. Post-laryngectomy Speech

Figure 2-6 shows some of the commonly used methods of recovering speech after a laryngectomy. In esophageal speech, air is swallowed and released in a belch which vibrates the esophagus imitating the vocal cord vibration. Tracheoesophageal prosthetics are a class of devices that provide a bridge between the trachea and the esophagus, often utilizing a one-way valve to prevent backflow from entering the lungs. Tapia's Artificial Larynx is a tube that is placed in the neck breathing hole or stoma; a reed in the middle vibrates as the air flows past it. The other end of the tube is inserted into the mouth providing the acoustic waves necessary for speech; the method suffers from distortion due to the sound source being introduced in the mouth and from the tube interfering with normal mouth and tongue motions. The final figure shows the electrolarynx in use.

To achieve a natural sounding voice, and ideally one that is close to the user's original voice, we draw upon years of research into speech recognition and synthesis. The larynx can be mechanically modeled in two main ways: parametric waveform (mathematical) or bio-mechanical (single-mass, dual-mass, multiple-mass, or continuous-mass). [24] [25] [26]



(a) Esophageal Speech (b) Tracheoesophageal

(c) Tapia's Artificial Larynx (d) Electrolarynx

Figure 2-6: Speech Production Alternatives [27]

The vocal tract area function is determined from the positions of the main articulators: the lips, tongue, and soft palate. Resonators include the mouth, nose, nasal sinuses, pharynx, and chest cavity. [28] The sound generated by the vocal folds consists of a single frequency, the pitch frequency, and integral harmonics of the pitch frequency. [27] Ramalingam and Kumaresan note that voiced-speech consists of nominally harmonic components, they show that the deviations can be extracted from recordings (or generated). Using these deviated waveforms creates a more normal sounding voice. [29] Veaux, et al. further the idea with "Voice Banking" and while their target user is different, the base concept is the same – available information about a user's original voice should be used to enhance post-operative speech. [30] The vocal folds are only necessary for creating vowels; consonants are created using the lips, teeth, tongue, and

soft palate. [27] Since all languages require vowels, unassisted speech is not possible without the vocal folds.

Mallis, et al. found after a total laryngectomy that 23.9% of patients in their study felt embarrassed about their voice, decreased their social interactions, and experienced negative effects on their sex lives. 56.5% had trouble communicating with strangers and 78.3% had difficulties on the telephone. An overwhelming number reported negative effects on their work life. [31]

2.5. The Electrolarynx

Many tracheoesophageal (TE) shunt valve speech prosthetic devices are available such as the airstream activated aerodynamical-mechanical devices including the Blom-Singer prosthesis, the Panje button, the Groningen button, the Singh button, and the Provox™ device. Each of these devices is surgically implanted to provide a bridge between the trachea and the esophagus via a one-way valve so that air can be forced into the vocal tract. [28] [21] While expensive, modern implantation devices achieve about 90% success rates and are the preferred treatment in developed countries. [22] [32] [33] Esophageal speech can also be used; in this technique, air is swallowed and temporarily trapped in the stomach, it is then released in a belch which causes the esophagus to vibrate simulating the vocal folds. The resulting sound is clear but very unnatural as it has a broad power spectrum unlike normal speech. The technique can be difficult to learn. [24] As an alternative or a backup to one of these devices, the electrolarynx (EL) can be used.

The EL is a handheld electromechanical vibrating device that creates acoustic pressure waves. When the device is pressed against the throat near the normal larynx

position, the acoustic waves travel through the skin into the vocal tract where they are transformed in a manner consistent with normal speech. [34] Electrolarynxes are easy to use, and the speech produced is intelligible but sounds mechanical and monotonous. One reason for the poor sound quality is acoustic sound leakage which results in a steady background noise. Since the EL is always vibrating when activated, even during unvoiced segments when the vocal folds are normally still, it provides a continuous periodic excitation which results in a lower Signal-to-Noise ratio and listener confusion. [20]

Commercially available electrolarynxes utilize a voice coil vibrating between 80 – 125 Hz which strikes a coupling plate which generates at least 50 dB sound pressure level (SPL). Figure 2-7 shows the Servox® Inton; the device is 12 cm tall by 3 cm in diameter. About 4 cm of the device is for the battery, 3 cm for the transducer and the other 5 cm is packaging and control. This device also includes two buttons implementing "variable frequency" although the difference is less than 10 Hz.

Figure 2-7: Servox® Inton

Choi, et al. created a haptic electrolarynx utilizing a force sensing resistor to vary the intensity as well as the frequency (as much as 30 Hz). [33] The ability to vary the fundamental frequency is an important characteristic that affects speech intelligibility. [35] Meltzner notes that the lack of pitch control, low frequency bandwidth, and self-noise are the major issues with current transcervical electrolarynxes. [3]

Modern intraoral electrolarynxes are mounted inside the mouth with a dental appliance. Merlo, et al. built an experimental intraoral device housed in dentures that is wirelessly controlled as an improvement on commercially available models such as the UltraVoice®. [36] Ahmadi et al. use a tube similar to Tapia's AL to inject ultrasound directly into the mouth; the ultrasound is shaped like normal sound. [37] There is no automatic conversion to audible sound; the shaped ultrasound is recorded by an external device, such as a cellphone, and converted to text using speech recognition techniques or frequency shifted down to normal speech range. The primary design was for silent communication as either text or telephone communications. For laryngectomees attempting to use it in normal communication, the device suffers from the same issues as Tapia's artificial larynx such as the distortion caused by having a tube to inject ultrasound.

While users get quite adept at controlling their electrolarynxes using whatever controls are available: pulsing the on/off, pitch control, frequency control, or a combination, the fundamental weakness is dedicating a hand to holding or controlling the device. Heaton, Goldstein, Hillman et al. experimented with monitoring the preserved laryngeal nerve via electromyography (EMG) and constructed a proof-of-concept device and compared voicing results with normal speakers, electrolarynx, tracheoesophageal,

and their EMG device. [38] [39] More recently Stepp, Heaton, Hillman et al. showed successful trials where voicing, pauses, and silence were correctly initiated. [40] Heaton et al. are continuing to push the capabilities of EMG controlled devices with the latest generation of their system which enables not only on/off but pitch control as well. [41] These early results are promising, and the continual miniaturization and increasing computational power of technology ensures that this type of activation is inevitable.

Previous work on extracting features from the speech waveform has provided techniques that will be utilized to help determine when a voiced sound is being attempted. [42] [43] Cole, et al. utilize spectral and cepstral subtraction methods to remove vocal tract noise created by the EL device itself. [44] Pandey, et al. note that the leakage, due to incomplete coupling, produces constant noise. Shielding results in only marginal reduction of the noise. [34]

Elderly people have a harder time using complex devices and a more difficult time understanding noisy speech. They also have a harder time handling peripheral devices. [45] A hands-free device that can react to biological stimulus rather than requiring conscious interaction will aid in the quality of life for these people. Depression is a common experience after a laryngectomy for many of these reasons; any improvement to the device, how it is controlled and how it is handled could have a dramatic effect on a laryngectomee's real life.

3. The Theory of Sound

An acoustic wave is a propagating pressure disturbance (a longitudinal compression wave). In simple terms, sound is produced when an object vibrates. An initial pressure change compresses matter – in solid, fluid, or gaseous form – creating compressed and rarefied regions, these areas in turn create pressure changes which repeat the process and propagate the wave. Attenuation will dissipate the wave's energy due to thermal effects caused by the medium's viscosity.

3.1. The Wave Equation

To describe the acoustic wave, the following physical properties are required:

- P = Pressure (Pascal = N/m^2)
- x = Displacement (m) ; v = dx/dt = velocity (m/s)
- t = time (s)
- ρ = Density (kg/m^3)

These properties are then related by three laws as derived in Dr. O'Brien's lecture on acoustics. [46]

3.1.1. Equation of State

Assume that pressure and density are made up of an equilibrium state and an excited state:

$$P = P_o + p$$
$$\rho = \rho_o + \rho_e \qquad 3.1$$

The excited states are assumed to be much smaller than the equilibrium states; i.e. $p \ll P_o$, and $\rho_e \ll \rho_o$.

Furthermore, assume that pressure is a function of density alone and expand in a Taylor series around the equilibrium (P_o, ρ_o): [2]

$$P_o + p = P(\rho_o + \rho_e) = P(\rho_o) + \rho_e \left(\frac{\partial P}{\partial \rho}\right)_o + \frac{\rho_e^2}{2!}\left(\frac{\partial^2 P}{\partial \rho^2}\right)_o + \frac{\rho_e^3}{3!}\left(\frac{\partial^3 P}{\partial \rho^3}\right)_o + \cdots \quad 3.2$$

Noting that $P_o = P(\rho_o)$, equation 3.2 simplifies to:

$$p = \rho_e \left(\frac{\partial P}{\partial \rho}\right)_o + \frac{\rho_e^2}{2!}\left(\frac{\partial^2 P}{\partial \rho^2}\right)_o + \frac{\rho_e^3}{3!}\left(\frac{\partial^3 P}{\partial \rho^3}\right)_o + \cdots \quad 3.3$$

Noting that $\rho_e = \rho - \rho_o$, equation 3.3 can be rewritten as:

$$\begin{aligned} p &= (\rho - \rho_o)\left(\frac{\partial P}{\partial \rho}\right)_o + \frac{(\rho - \rho_o)^2}{2!}\left(\frac{\partial^2 P}{\partial \rho^2}\right)_o + \frac{(\rho - \rho_o)^3}{3!}\left(\frac{\partial^3 P}{\partial \rho^3}\right)_o + \cdots \\ &= \rho_o \left(\frac{\partial P}{\partial \rho}\right)_o \left(\frac{\rho - \rho_o}{\rho_o}\right) + \frac{\rho_o^2}{2!}\left(\frac{\partial^2 P}{\partial \rho^2}\right)_o \left(\frac{\rho - \rho_o}{\rho_o}\right)^2 + \frac{\rho_o^3}{2!}\left(\frac{\partial^3 P}{\partial \rho^3}\right)_o \left(\frac{\rho - \rho_o}{\rho_o}\right)^3 + \cdots \end{aligned} \quad 3.4$$

Ignoring the higher-order terms:

$$p = A\left(\frac{\rho - \rho_o}{\rho_o}\right) + \frac{B}{2}\left(\frac{\rho - \rho_o}{\rho_o}\right)^2 = As + \frac{B}{2}s^2 \quad 3.5$$

where $s = \frac{\rho - \rho_o}{\rho_o}$ is the condensation, the first term is the linearized approximation, $p = \rho_e c_o^2$, and B/A is used to express the nonlinear properties of the medium.

3.1.2. Continuity Equation

The continuity equation is a conservation of mass equation; in a fluid, mass is given by $\iiint_V \rho dV$ where ρ is the density and V is volume. Fluid dynamics conventions

---

[2] For this to be true, the process must be adiabatic – heat conductivity is negligible when thermal conductivity is low. This is a standard assumption for acoustic waves based on reasoning that the alternation of compression and rarefication occurs so rapidly that no heat can flow from compressed areas to rarefied areas before the compression is over; this reasoning is in fact, incorrect. However, the assumption still holds as long as $\omega \ll c^2/\kappa$ where ω is the angular frequency and κ is the thermal diffusivity due to the fact that heat does not have time to diffuse between the compressed and rarefied areas. [61] For air, water, and tissue, this condition holds well in the megahertz range or higher.

for finite control volumes consider a unit time in which mass is flowing through a surface S; **n** is a unit vector normal to the flow facing outwards – flow out of the volume is positive. The change or flow of mass is then given by:

$$\dot{m} = \iint_S \rho \boldsymbol{v} \cdot \boldsymbol{n} dS = -\frac{\partial}{\partial t} \iiint_V \rho dV \qquad 3.6$$

Conservation of mass says that the rate of mass flowing out of the control volume minus the rate of mass flowing into the control volume plus rate of mass buildup inside the control volume must be zero.

$$\iint_S \rho \boldsymbol{v} \cdot \boldsymbol{n} dS + \frac{\partial}{\partial t} \iiint_V \rho dV = 0 \qquad 3.7$$

Using the divergence theorem and since the equation must be valid for any volume:

$$\nabla \cdot (\rho \boldsymbol{v}) + \frac{\partial \rho}{\partial t} = 0 \qquad 3.8$$

where ∇ is the gradient operator. Equation 3.8 is the continuity equation. [47]

3.1.3. Equation of Motion

Euler's equation of motion is used to express Newton's second law for the conservation of momentum applied to a fluid. Assuming a cubic fluid particle with dimensions (δx, δy, δz) small enough to have constant density and ignoring external forces such as gravity, the mass of the particle is:

$$\delta m = \rho(\delta x)(\delta y)(\delta z) \qquad 3.9$$

The particle's acceleration is:

$$a = \frac{D\boldsymbol{v}}{Dt} = \frac{\partial \boldsymbol{v}}{\partial t} + u\frac{\partial \boldsymbol{v}}{\partial x} + v\frac{\partial \boldsymbol{v}}{\partial y} + w\frac{\partial \boldsymbol{v}}{\partial z} = \frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} \qquad 3.10$$

where D/Dt is the total derivative, also known as the Eulerian or material derivative and ∇ is the gradient operator.

Assuming no viscosity, the force on the particle in any dimension is:

$$\delta F_x = \left[P - \frac{\partial P}{\partial x}\left(\frac{\delta x}{2}\right)\right](\delta y)(\delta z) - \left[P + \frac{\partial P}{\partial x}\left(\frac{\delta x}{2}\right)\right](\delta y)(\delta z) = -\frac{\partial P}{\partial x}(\delta x)(\delta y)(\delta z)$$
$$\delta F_y = (\delta x)\left[P - \frac{\partial P}{\partial y}\left(\frac{\delta y}{2}\right)\right](\delta z) - (\delta x)\left[P + \frac{\partial P}{\partial y}\left(\frac{\delta y}{2}\right)\right](\delta z) = -\frac{\partial P}{\partial y}(\delta x)(\delta y)(\delta z) \quad 3.11$$
$$\delta F_z = (\delta x)(\delta y)\left[P - \frac{\partial P}{\partial z}\left(\frac{\delta z}{2}\right)\right] - (\delta x)(\delta y)\left[P + \frac{\partial P}{\partial z}\left(\frac{\delta z}{2}\right)\right] = -\frac{\partial P}{\partial z}(\delta x)(\delta y)(\delta z)$$

Using Newton's second law:

$$\delta F = \delta m a = \rho(\delta x)(\delta y)(\delta z)\left(\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}\right) = -\nabla P(\delta x)(\delta y)(\delta z) \quad 3.12$$

Divide by $\rho(\delta x)(\delta y)(\delta z)$ and take the limit as δx → 0, δy → 0, δz → 0:

$$\frac{D\boldsymbol{v}}{Dt} = -\frac{\nabla P}{\rho} \quad 3.13$$

3.1.4. Wave Equation

Combining these three equations (3.5, 3.8, 3.12) a nonlinear wave equation can be derived as detailed by Rasmussen et al. [48]

We assume that the flow is inviscid-irrotational, $\mu = 0$ and $\nabla \times \mathbf{v} = 0$, therefore

$$\boldsymbol{v} \equiv -\nabla\Phi \quad 3.14$$

where Φ is the velocity potential and

$$\nabla^2\Phi = \nabla\cdot(\nabla\Phi) = 0 \quad 3.15$$

where ∇^2 is the Laplacian. Equation 3.15 is Laplace's equation.

Rasmussen et al. derive a more general nonlinear equation with heat transfer and viscosity, however, we simplify their derivation based upon our assumptions thus far:

$$\frac{\partial^2 \Phi}{\partial t^2} - c_o^2 \nabla^2 \Phi = \frac{\partial \Phi}{\partial t} \nabla^2 \Phi + \frac{\partial}{\partial t}\left(\frac{\kappa}{\rho_o}\left(\frac{1}{c_v} - \frac{1}{c_p}\right) + (\nabla \Phi)^2 + \frac{B/A - 1}{2c_o^2}\left(\frac{\partial \Phi}{\partial t}\right)^2\right) \quad 3.16$$

where κ is the heat conductivity coefficient, c_v is the specific heat at constant volume, and c_p is the specific heat at constant pressure. For a full derivation, see [49].

In the linearized form, we linearize each of the three constituent equations. The equation of state (3.5) becomes:

$$p = \rho_o \left(\frac{\partial P}{\partial \rho}\right)_o \left(\frac{\rho - \rho_o}{\rho_o}\right) = \rho_e \left(\frac{\partial P}{\partial \rho}\right)_o = \rho_e c^2 \quad 3.17$$

Equation 3.17 is a good approximation when pressure and density are low; however a better approximation for an adiabatic process uses the ratio of specific heats, $\gamma = c_p / c_v$, yielding $c^2 = \gamma\, p / \rho_o$.

For the continuity equation (3.8), we use the product rule for divergence to expand the left hand part and approximate $\rho \approx \rho_o$:

$$\nabla \cdot (\rho \boldsymbol{v}) + \frac{\partial \rho}{\partial t} = \nabla \cdot \boldsymbol{v} + \frac{\boldsymbol{v} \cdot \nabla \rho}{\rho_o} + \frac{1}{\rho_o}\frac{\partial \rho}{\partial t} = 0 \quad 3.18$$

As long as the equilibrium density does not change, $\partial \rho_o / \partial t = 0$, then there is no net flow at equilibrium $\mathbf{v}_o = 0$, and

$$|\nabla \cdot \boldsymbol{v}| \gg \left|\frac{\boldsymbol{v} \cdot \nabla \rho}{\rho_o}\right| \quad 3.19$$

therefore, the right hand side can be safely ignored when $|\mathbf{v}| \ll c_o$, yielding the final linearized continuity equation:

$$\frac{\partial \rho_e}{\partial t} = -\rho_o \nabla \cdot \boldsymbol{v}_e \quad 3.20$$

For the equation of motion (3.12), the total derivative is expanded and approximate $\rho \approx \rho_o$:

$$\frac{D\boldsymbol{v}}{Dt} = \frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = -\frac{\nabla P}{\rho_o} \tag{3.21}$$

As long as the equilibrium pressure is the same, $\nabla p_o = 0$, then there is no net flow at equilibrium $\mathbf{v}_o = 0$, and

$$\left|\frac{\partial \boldsymbol{v}}{\partial t}\right| \gg |(\boldsymbol{v} \cdot \nabla)\boldsymbol{v}| \tag{3.22}$$

therefore, the right hand side can be safely ignored when $|\mathbf{v}| \ll c_o$,, yielding the final linearized equation of motion:

$$\frac{\partial \boldsymbol{v}_e}{\partial t} = -\frac{\nabla p}{\rho_o} \tag{3.23}$$

Combining the linearized equations (3.17, 3.20, 3.23) will yield the linearized wave equation. We start by differentiating the linearized equation of state with respect to time:

$$\frac{\partial p}{\partial t} = c_o^2 \frac{\partial \rho_e}{\partial t} \tag{3.24}$$

Next, we substitute equation 3.20, yielding:

$$\frac{\partial p}{\partial t} = -\rho_o c_o^2 \nabla \cdot \boldsymbol{v}_e \tag{3.25}$$

Then, differentiate with respect to time, yielding:

$$\frac{1}{\rho_o c_o^2}\frac{\partial^2 p}{\partial t^2} = -\frac{\partial}{\partial t}(\nabla \cdot \boldsymbol{v}_e) \tag{3.26}$$

And then we differentiate equation 3.23 with respect to space, yielding:

$$-\nabla \cdot \left(\frac{\partial \boldsymbol{v}_e}{\partial t}\right) = \nabla \cdot \left(\frac{\nabla p}{\rho_o}\right) \tag{3.27}$$

As the divergence yields a scalar, we can change the order of differentiation in the right-hand side of equation 3.27 and equate it to the left-hand size of equation 3.26:

$$\frac{1}{c_o^2}\frac{\partial^2 p}{\partial t^2} - \rho_o \nabla \cdot \left(\frac{\nabla p}{\rho_o}\right) = 0 \qquad 3.28$$

Finally, as previously stated, the equilibrium pressure is the same, $\nabla p_o = 0$, therefore:

$$\nabla^2 p - \frac{1}{c_o^2}\frac{\partial^2 p}{\partial t^2} = 0 \qquad 3.29$$

Equation 3.29 is the linearized wave equation which will be used in the next section for the finite-difference time-domain derivation. Various substitutions can be used to derive the wave function in terms of density and displacement; in both cases, it has the same form.

3.2. Sound wave terminology

As noted earlier, unlike electromagnetic waves, sound waves require matter to propagate. In general, the speed of sound varies with the density, elastic modulus and temperature of matter as these properties impact how atoms move due to pressure fluctuations.

The primary parts of a wave are shown in Figure 3-1. The amplitude of a wave is the distance from resting state to the crest. For sound waves, amplitude is the measure of pressure created by the wave. The crest or peak is the maxima of the amplitude, and the trough is the minima.

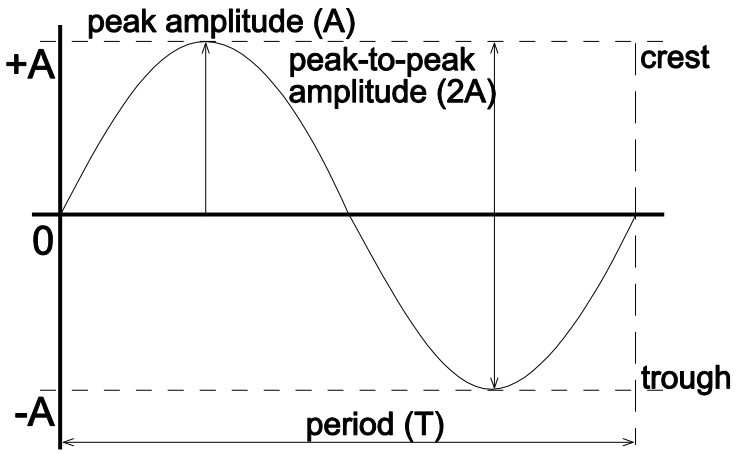


Figure 3-1: Properties of a Wave

The period of a wave (T) describes one cycle of the wave from rest to crest through rest to trough and back to rest. The frequency of a wave is the reciprocal of the period. The wavelength is the distance from any two phase related points within a length of one period; for example from crest-to-crest or trough-to-trough. The frequency is related to wavelength by equation 3.30, where f is frequency, c is the speed, and λ is the wavelength. Pitch refers to the perceived frequency of a sound.

$$f = \frac{c}{\lambda} \qquad 3.30$$

The phase of a wave describes the relationship between the wave's angle and the origin defined as the distance from the origin to the first rest or zero-crossing. Phase in conjunction with frequency determines where in the wave's cycle the zero-crossings, crest and trough occur.

The loudness of a sound, or sound intensity level (SIL), is determined by its amplitude and intensity and measured in decibels (dB) as shown in equation 3.31. Decibels are a relative (dimensionless) measurement; for loudness, the reference $I_o = 10^{-12}$ W/m^2 is the minimum intensity detectable by the human ear of a pure tone at 1 kHz.

$$L = 10 \log_{10} \frac{I}{I_o} \qquad 3.31$$

Sound pressure level (SPL) is often confused with loudness, but where loudness is a somewhat subjective measure (regardless of what equation 3.31 says), SPL is a definite measurable quantity. SPL is a relative measure of sound pressure at a specified distance given in decibels. The reference, p_{ref}, is typically 20 μPa representing the RMS sound pressure threshold of human hearing for a 1 kHz tone in air; for water 1 μPa is used. SPL is expressed as:

$$SPL = 10 \log_{10} \frac{p_{rms}^2}{p_{ref}^2} = 20 \log_{10} \frac{p}{p_{ref}} \qquad 3.32$$

where p is pressure measured at a specific distance.

For reference, normal talking at a distance of 1m has an SPL of 40-60 dB (Fox and Ramig recorded an average SPL of 72.6 ± 1.1 dB at 30 cm for healthy men and women [50]) and the threshold of pain measured at the ear is 134 dB.

When a sound source is moving relative to an observer or the medium transmitting the sound is in motion, the observed frequency is changed. This change due to motion is called the Doppler effect.

3.3. Properties of sound

Sound waves share many properties with electromagnetic waves, but here we focus on the properties as they apply to sound waves.

Sound waves passing between differing materials will undergo both reflection and refraction. At the boundary, part of the wave's energy will be reflected back into the original material and part will pass into the new material but on a different trajectory.

Snell's law relates the angle of incidence, angle of reflection/refraction, and the velocities of sound in each material.

Acoustic impedance or characteristic impedance describes the degree of mobility of a medium and is calculated as the product of density and the speed of sound in that medium. At the boundary of two tissues with differing impedances, reflection occurs. The percentage of reflection can be calculated using:

$$R = 100\left(\frac{Z_2 - Z_1}{Z_2 + Z_1}\right)^2 \qquad 3.33$$

therefore, the percentage of refraction is 100 – R.

When sound encounters an opening or edge, it bends. As in Young's famous double slit experiment where he deduced the wave nature of light, diffraction describes how waves deflect when encountering an obstacle. Diffraction and interference are basically synonymous with the distinction that interference is most often used with few sources and diffraction is used when there are many sources. [51] Given a plane wave, the minima can be calculated by:

$$\sin\theta = \frac{n\lambda}{d} \qquad 3.34$$

where d is the opening distance (or the fringe spacing depending on usage), λ is the wavelength, θ is the diffraction angle for the first minima, and n depends on the opening shape (1 for rectangular, 1.22 for circular). The location of the maxima and minima may be reversed depending on the situation such as in the double slit experiment.

Interference occurs when two or more waves intersect. At any given location, the waves can be summed together based on their individual amplitude, frequency and phase; an increase in amplitude is called constructive interference and a decrease is destructive

interference. When two waves interact in such a way that the interference pattern causes the SPL to vary at a given location at a stable third frequency, the new frequency is called the beat frequency and is the absolute difference of the original two waves:

$$f_b = |f_2 - f_1| \qquad 3.35$$

3.4. Analyzing waves

Each of the major properties of a wave can be measured. The entire wave can be sampled in the time-domain as a relative sound pressure measurement using a microphone which converts pressure to voltage; care must be taken to make sure that the sampling rate is at least twice the wave frequency (Nyquist rate) otherwise higher frequency components are folded or aliased into lower frequencies. Figure 2-4 showed an example of a time-domain waveform of human vocalization. Direct frequency measurement is also possible although rarely practical; the human ear works this way with thousands of cilia which respond to specific frequencies based on their position within the curvature of the cochlea.

Given a time sampled pressure signal, the frequencies and phase (average) can be obtained using the Fourier transform. In a digital signal, the best method is the discrete Fourier transform (DFT) as it works with a finite sample. The result of the DFT is a number of bins where each bin represents a bandwidth of f_s / N Hz where f_s is the sampling frequency and N is the number of samples used in the DFT. Each bin contains a complex number which encodes the average magnitude and phase for all frequencies within that bandwidth; for real signals such as sound waves, the imaginary portion is usually zero.

The short-time Fourier transform (STFT) is the DFT with a window function which allows selecting whether frequency (wide window) or time (short window) is more important. A portion of the signal is multiplied by the window function and the result is run through DFT. An important property of the STFT is that it is invertible.

There are many window functions such as Hanning, Hamming, and Blackman-Harris. Typically a sliding window is used where a portion of the signal is analyzed, then the window is shifted some percentage of the signal forward such that the window function overlaps some of the originally analyzed portion of the signal and a new portion of the signal. A sliding window helps minimize jumps and helps display a smooth spectrum when adjacent STFT signals are displayed side-by-side. Care should be taken in selecting a window function so that the percentage of overlap and the window function results in the overlapped coefficients summing as closely to 1 at every location as possible.

The STFT is a three-dimensional construct. It provides time, frequency, and the magnitude of each frequency bin. A spectrogram is a series of STFT transforms laid side-by-side; typically, the magnitude of each frequency bin is assigned a gradated color so that a two-dimensional plot can display the surface. Figure 2-5 shows an example of a spectrogram.

Interestingly, while the STFT can display the primary frequencies in a signal, beat frequencies as described in Section 3.3 are not shown. Beat frequencies appear in the time-domain as an amplitude modulation of the waveform. In order to visualize beat frequencies, the envelope of the waveform must be taken. Given a spectrogram, the

envelope can be extracted by applying a STFT version of the discrete Hilbert transform. [52]

$$\mathcal{H}[i(n)] = \mathcal{F}^{-1}\{\mathcal{F}\{i(n)\} \cdot u(n)\} \tag{3.36}$$

where i(n) is the input signal, F and F^{-1} are the STFT and inverse STFT, and $u(n)$ is:

$$u(n) = \begin{cases} -1, & n = 0, {}^{N}/_{2} \\ -j, & n = 1, \dots, {}^{N}/_{2} - 1 \\ j, & n = {}^{N}/_{2} + 1, \dots, N - 1 \end{cases} \tag{3.37}$$

$u(n)$ is a phase shift of 90° and is basically a swapping of the real and imaginary parts of the signal except for n=0 and n=N/2 (the DC and Nyquist components which are always real). The envelope is then given by:

$$e(n) = \sqrt{i(n)^2 + \mathcal{H}[i(n)]^2} \tag{3.38}$$

Given a real input, the envelope is also a real signal and may be run through the STFT itself to generate a spectrogram of the envelope. This spectrogram will not show the original component frequencies, but will now show the beat frequency.

4. Simulating the Wave Equation

The acoustic wave equation presented in Section 3.1.4 is an exact analytical solution, accounting for the linearization assumptions; however, due to limitations in how computers represent numbers, a discretized solution that minimizes numerical error and provides acceptable performance is required.

Every number in a computer is represented by a finite number of bits. Integers in modern systems are typically 32 or 64 bits and may be signed or unsigned. A signed integer has a range of $[-2^{b-1}, 2^{b-1}-1]$ where b is the number of bits. Real numbers are most commonly approximated using IEEE 754 floating point numbers. Approximated is the important distinction that is often overlooked.

IEEE floating point numbers are represented by a sign (positive or negative), the mantissa, and a signed integer exponent; value = $m \cdot 2^e$. There are two common floating point precisions: single and double. A single precision floating point number consists of 32 bits with 1 sign bit, 23 mantissa bits, and 8 exponent bits; a double precision floating point number consists of 64 bits with 1 sign bit, 52 mantissa bits, and 11 exponent bits. Notice that a double precision floating point actually has more than twice the mantissa bits and therefore is more than twice as precise. Within this representation are many hidden computational issues that are not present in analytic solutions. For example, analytically x = x + y implies that y is 0; however, with floating point numbers, this is not necessarily true. When x and y have very different exponent values, y may not contribute anything to the final value; the significant digits from the mantissa may have been shifted off when equalizing the exponents. Similarly, many numbers such as irrational numbers or even simple fractions cannot be exactly represented with floating point numbers.

Section 4.2 describes the voxel model used in the simulation, but for now, it is only important to note that the model consists of a segmented three-dimensional volume. Just like numerical representations, this segmentation has discrete boundaries and therefore inherent errors as each area may consist of more than one tissue type. The finer the segmentation, the lower the error, but given a finite amount of memory and computational time, an exact cellular representation is not possible at this time.

Many solutions exist for numerically calculating the wave equation. One of the best class of solutions are the pseudospectral methods as they approximate derivatives over the entire model and therefore offer higher accuracy and lower computational overhead. However, implementation requires a relaxed or variable grid spacing which is inappropriate for the voxel model. [53] The finite element methods and finite volume methods deal well with complex geometry, allow for bounded error estimates, and have "moderate" memory requirements; however, implementation can be complicated.

4.1. Finite-Difference Time-Domain (FDTD)

The finite difference methods also work with complex geometry, but suffer from increased computational and memory requirements; however, it has the simplest implementation and many of the memory issues are mitigated by the memory requirements of the model itself. Furthermore, GPU implementations for finite difference methods are readily available which reduces the impact of the computational overhead and allows real-time or near real-time computation for many problems.

4.1.1. FDTD Derivation

In Chapter 12 of his online book, Schneider derives a two-dimensional acoustic FDTD implementation which will guide the three-dimensional derivation provided here. [54] The finite-difference time-domain (FDTD) method represents derivatives as local functions of a computational grid. While there are various ways of computing the "differences", the central difference will be used here. Therefore, the derivative of a function *f* at a point x_o is given by:

$$\dot{f}(x_o) = \lim_{\delta \to 0} \frac{f(x_o + \delta) - f(x_o - \delta)}{2\delta} \tag{4.1}$$

In a numerical approximation, δ does not go to zero, therefore an error will be present. This error can be calculated by expanding each function using a Taylor series expansion. Below is the error for a second-order approximation.

$$\dot{f}(x_o) \approx \dot{f}(x_o) + \frac{1}{3!}\frac{\delta^2}{2^2}\dddot{f}(x_o) + \cdots = \frac{f(x_o + \delta) - f(x_o - \delta)}{2\delta} + O(\delta^2) \tag{4.2}$$

As δ becomes "reasonably" small, the $O(\delta^2)$ error term becomes negligible. If a fourth-order approximation is used, the error term becomes $O(\delta^4)$; the rest of the derivation assumes a second-order approximation, but the simulation uses a fourth-order central difference approximation for spatial derivatives and a second-order for temporal derivatives.

Section 3.1 defined the physical properties required to describe an acoustic wave. The linearized equations of velocity (equation 3.23) and pressure (equation 3.25) must be discretized in both space and time. The problem is that both equations depend on both unknowns. The trick first proposed by Kane Yee in May 1966 for solving electromagnetic equations is to "stagger" the fields in both space and time updating each

equation in turn and using the results for the next equation and repeating until the simulation is complete.

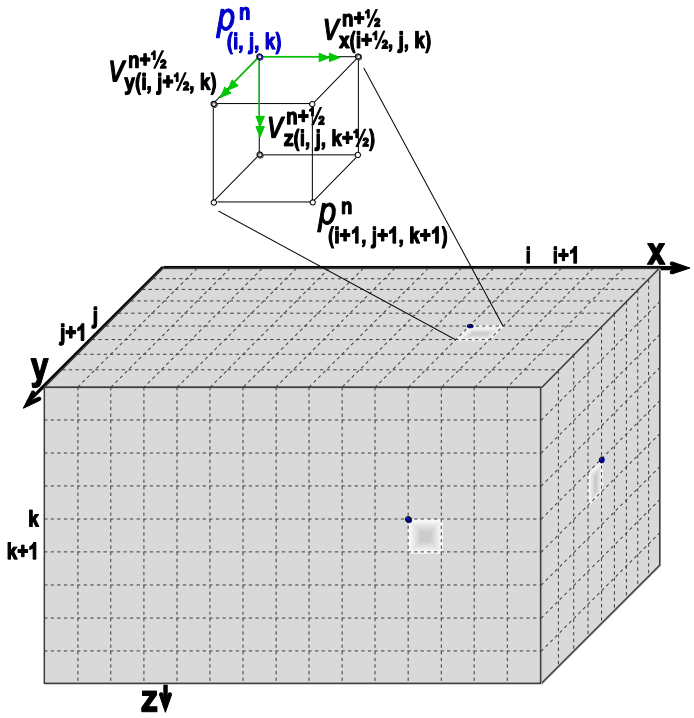


Figure 4-1: FDTD grid and acoustic unit simulation node [55]

Figure 4-1 shows a three-dimensional computational domain or mesh with a single simulation grid node expanded showing the staggered space and time. Note that the staggering is a conceptual artifact; during implementation integer indexing is used as shown in the final form of each equation below. The staggered space and time offsets results in the following discretization:

$$
\begin{aligned}
p(x,y,z,t) &= p(i\Delta_x, j\Delta_y, k\Delta_z, n\Delta_t) = p^n[i,j,k] \\
v_x(x,y,z,t) &= v_x\left((i+½)\Delta_x, j\Delta_y, k\Delta_z, (n+½)\Delta_t\right) = v_x^{n+½}[i,j,k] \\
v_y(x,y,z,t) &= v_y\left(i\Delta_x, (j+½)\Delta_y, k\Delta_z, (n+½)\Delta_t\right) = v_y^{n+½}[i,j,k] \\
v_z(x,y,z,t) &= v_z(i\Delta_x, j\Delta_y, (k+½)\Delta_z, (n+½)\Delta_t) = v_z^{n+½}[i,j,k]
\end{aligned}
\qquad 4.3
$$

Inserting the discrete indexes into equation 3.25 and assuming that the spatial step sizes are equal ($\Delta_x = \Delta_y = \Delta_z = \delta$), yields the pressure update equation:

$$p^n[i,j,k] = p^{n-1}[i,j,k] - \rho c^2 \frac{\Delta_t}{\delta}\begin{pmatrix} v_x^{n-½}[i,j,k] - v_x^{n-½}[i-1,j,k] + \\ v_y^{n-½}[i,j,k] - v_y^{n-½}[i,j-1,k] + \\ v_z^{n-½}[i,j,k] - v_z^{n-½}[i,j,k-1] \end{pmatrix} \quad 4.4$$

Equation 4.4 depends only on the previous pressure at this location and the previous staggered velocity. Inserting the discrete indexes into equation 3.23 and keeping the spatial step sizes equal, yields the velocity update equations:

$$\begin{aligned} v_x^{n+½}[i,j,k] &= v_x^{n-½}[i,j,k] - \frac{1}{\rho}\frac{\Delta_t}{\delta}(p^n[i+1,j,k] - p^n[i,j,k]) \\ v_y^{n+½}[i,j,k] &= v_y^{n-½}[i,j,k] - \frac{1}{\rho}\frac{\Delta_t}{\delta}(p^n[i,j+1,k] - p^n[i,j,k]) \\ v_z^{n+½}[i,j,k] &= v_z^{n-½}[i,j,k] - \frac{1}{\rho}\frac{\Delta_t}{\delta}(p^n[i,j,k+1] - p^n[i,j,k]) \end{aligned} \quad 4.5$$

It is worth noting that the density and speed of sound need not be global but can be functions of space. In this case, both can be "located" on the same staggered grid as pressure. The speed of sound is not needed during the calculation for velocity, but the density is; since velocity is spatially staggered from pressure, the two opposing density values should be averaged.

Since pressure is a scalar field, time zero consists of the initial pressures and the simulation starts with calculating the velocity field. At each time step, the calculation switches between the staggered grids, first updating velocity and then pressure.

As mentioned earlier, one of the advantages of FDTD methods is the ability to calculate the wave equation over complex geometry. Since the method makes no physical approximations, propagation includes refraction, diffraction and reflections. [56] This same fact brings some complications; as the propagation reaches the simulation

boundary, phantom reflections are generated. To reduce these aberrations, it is necessary to highly attenuate signals as they approach the simulation boundary. There are many methods of doing so, but two of the most popular are Mur's "Absorbing Boundaries" and Bérenger's "Perfectly Matched Layers." (PML) [57] [58] Mur's method is decidedly simpler to implement, but Bérenger's is more computationally exact and therefore generates less artifacts. Both methods have been extended over the years and are still actively researched and utilized. [59] [60] PML requires splitting the signal into various components and while very effective and accurate, it is difficult to implement efficiently on the GPU. We decided to use the Mur method as it can be implemented directly into our GPU code with minimal impact to efficiency and has acceptable results especially considering that waveforms reaching the boundary area are highly attenuated already as they have passed through the model.

4.1.2. FDTD Error Analysis

While it is always possible to perform calculations at each time step, the results may not be valid. As pointed out at the beginning of this section, numerical precision is an important consideration in any discretized numerical calculation. Some steps should be taken to minimize error. First, in order to avoid spatial and/or spectral aliasing, the Nyquist rate requires the time step to encompass at least two points per frequency period. Also, a general rule of thumb is that the spatial spacing should be 5-10 times less than the smallest signal wavelength. Petropoulos provides an equation for calculating the number of points per wavelength for the fourth-order spatial and second-order temporal FDTD as: [61]

$$N_{PPW} \sim \left(\frac{3}{20}\right)^{\frac{1}{4}} \pi^{\frac{5}{4}} (sin^6\theta + cos^6\theta)^{\frac{1}{4}} \left(\frac{P}{e_\phi}\right)^{\frac{1}{4}} \quad 4.6$$

where N_{PPW} is the number of points per wavelength, θ is the wave propagation angle, P is the number of cycles of the highest frequency component, and e_φ is a maximum phase error in radians. $P = t \cdot f$ where t is the simulated time and f is the frequency. Equation 4.6 can also be used to estimate the phase error for a given points per wavelength. For the leapfrog FDTD presented in sub-section 4.1.1, the dispersion is maximal along the coordinate axes and minimal diagonally; setting θ to nπ/2 where n is an integer will therefore allow calculating the maximum phase error. [62]

The maximum stable time step can be calculated from the Courant number. [63]

$$\Delta_t \leq \frac{\delta}{c_{max}\sqrt{3}} \quad 4.7$$

where Δ_t is the time step, δ is the spatial step, and c_{max} is the maximum material velocity in the simulation domain.

Given a selection for δ and Δ_t, we can calculate the error introduced by the finite-difference truncation and dispersion. Warnick presents a solution for the error given a second-order central difference FDTD with a time-harmonic source which we will follow to derive a similar solution for the fourth-order central difference FDTD. [64] Given:

$$\ddot{f}(x_o) = \frac{-f(x_o + 2\delta) + 16f(x_o + \delta) + 30f(x_o) + 16f(x_o - \delta) - f(x_o - 2\delta)}{12\delta^2} + O(\delta^4) \quad 4.8$$

We can represent the central difference, CD_{FDTD}, using exponential functions.

$$\ddot{f}(x_o) \approx \frac{-e^{ik(x_o+2\delta)} + 16e^{ik(x_o+\delta)} - 30e^{ikx_o} + 16e^{ik(x_o-\delta)} - e^{ik(x_o-2\delta)}}{12\delta^2} \quad 4.9$$

$$= \frac{e^{ikx_o}}{12\delta^2}\left[-e^{ik2\delta} + 16e^{ik\delta} - 30 + 16e^{-ik\delta} - e^{-ik2\delta}\right]$$

$$= \frac{e^{ikx_o}}{12\delta^2}\left(2 - e^{ik\delta} - e^{-ik\delta}\right)\left(e^{ik\delta} + e^{-ik\delta} - 14\right)$$
$$= \frac{e^{ikx_o}}{3\delta^2}\left[1 - \left(\frac{e^{ik\delta} + e^{-ik\delta}}{2}\right)\right]\left[\left(\frac{e^{ik\delta} + e^{-ik\delta}}{2}\right) - 7\right]$$
$$= \frac{e^{ikx_o}}{3\delta^2}\left(\frac{1 - \cos k\delta}{2}\right)(\cos k\delta - 7) = \frac{e^{ikx_0}}{3\delta^2}\sin^2\frac{k\delta}{2}(\cos k\delta - 7)$$

where k is the wavenumber (ω/c).

We can now calculate the relative error, RE_{FDTD}, as:

$$RE_{FDTD} = \frac{\left|CD_{FDTD} - \ddot{f}(x_o)\right|}{\left|\ddot{f}(x_o)\right|}$$
$$= \frac{\left|CD_{FDTD} - \frac{\partial^2 e^{ikx_o}}{\partial x_o^2}\right|}{\left|\frac{\partial^2 e^{ikx_o}}{\partial x_o^2}\right|}$$
$$= \frac{\left|\left[\frac{e^{ikx_0}}{3\delta^2}\sin^2\frac{k\delta}{2}(\cos k\delta - 7)\right] + k^2 e^{ikx_o}\right|}{\left|-k^2 e^{ikx_o}\right|} \qquad 4.10$$
$$= \frac{\left|2\sin^2\frac{k\delta}{2}(\cos k\delta - 7) + 3k^2\delta^2\right|}{\left|-3k^2\delta^2\right|}$$
$$= \left|1 + \frac{\sin^2\frac{k\delta}{2}(\cos k\delta - 7)}{6\left(\frac{k\delta}{2}\right)^2}\right| = \left|1 + \frac{1}{6}\mathrm{sinc}^2\frac{k\delta}{2}(\cos k\delta - 7)\right|$$

RE_{FDTD} represents the truncation error $O(\delta^4)$ from equation 4.8. Finally, we can calculate the total phase error in radians due to dispersion from a plane wave propagating in a homogenous material in one dimension:

$$|Error| = \frac{1}{2} \times \left|1 + \frac{1}{6}\mathrm{sinc}^2\frac{k\delta}{2}(\cos k\delta - 7)\right| \times (kD) \qquad 4.11$$

where the middle term is RE_{FDTD} and the final term represents dispersion over error a distance D. As noted previously, the worst case dispersion occurs parallel to the

axes in the classic leapfrog FDTD configuration, so this error represents an upper bound on the error in any single direction.

4.2. The AustinMan Model

The AustinMan Electromagnetic Voxels Model was created by Jackson Massey in 2011 as his undergraduate thesis for simulating electromagnetic radiation exposure from wireless devices. [65] Massey started with high-resolution cryosection images provided by the National Library of Medicine's Visible Human Project, developed semi-automated segmentation and tissue labeling algorithms, and manually reviewed and adjusted the masks and labels. [66] The results were shared as sets of voxels in UCD format at various resolutions. Since the original dataset was released in January 2011, the Computational Electromagnetics Group has continued the work and created detailed voxel sets of both the entire male and female bodies.

The AustinMan voxels provide us with a high resolution digital representation of a real person. While the data set is not a cellular representation of a person, such as set would be too large and complex for our simulation anyway. Rather than using an abstract or artist's 3D drawing of a human, the voxels encode a real body with actual boundaries within an acceptable error limit. Using this data set saves months of potential work developing our own model and provides as accurate a representation of the internal structures as our simulator can handle.

The voxels are available in various resolutions. We utilized the 1x1x1 mm^3 version for our simulator.

5. Ultrasound

Ultrasound is any sound wave above human hearing which is typically accepted as being 20 kHz. Ultrasound is used for medical imaging, cleaning (industrial such as waste treatment, commercial such as surgical instruments, and residential such as jewelry and teeth), industrial measurement (thickness and flow rates), plastic fusing, and surgery (such as breaking up kidney stones).

In medical imaging, an ultrasonic wave is sent into the body. As the wave encounters tissue boundaries, the acoustic impedance causes a portion of the wave to be reflected. Although not absolute, most imaging wands function both as emitters and receivers, rapidly switching between the two functions. When the reflected wave is received, the total "time of flight" (round-trip) is captured. Given that most soft tissues have very similar densities and speeds of sound, an average value can be used to calculate the reflection's depth. As the signal is attenuated the further it travels (and also based on its frequency), the signal is adjusted for depth and then an intensity value is calculated based on the adjusted signal.

In A-mode, a single wave is generated resulting in a one-dimensional "image." In B-mode, an array of transducers emit at specific times (phased array) to steer the wavefront allowing the beam to be swept back and forth in a plane generating a two-dimensional slice. M-mode makes multiple images of the same plane in rapid succession allowing internal motion to be displayed.

The spatial resolution increases in proportion to the frequency, however, the penetration depth falls off with increasing frequency at about 1 dB / 2 cm / MHz.

Acoustic radiation force imaging (ARFI) is a relatively new imaging modality that utilizes nonlinear effects of attenuation and scattering to induce vibrations in tissue. While its use in imaging is new, the acoustic radiation force has been investigated since the late 1920s. [67] When tissue is subjected to a focused sound wave, shear displacement occurs generating low-frequency shear acoustic waves. The shear waves have a very low velocity and therefore generate tiny displacements. These displacements are measured using standard B-mode or Doppler ultrasound. [68]

Vibroacoustography is another relatively new imaging mode that uses two ultrasonic waves of slightly different frequencies to induce low frequency mechanical excitations. These vibrations result in a beat frequency in the kHz range which is re-emitted and detected. [69] This method allows gathering information about the mechanical properties of the induced tissue such as elasticity which can be of great use as abnormal tissue often has very different properties. Fatemi and Greenleaf show that "the acoustic emission field amplitude is linearly proportional to the square of the incident ultrasound pressure." [70]

Similarly to vibroacoustography, Yang et al. investigated the use of dual ultrasound beams to generate audible sound. [71] They derived a set of nonlinear and quasilinear equations to quantify the beam interactions. They further experimentally verified their theory in an air filled anechoic (echoless) chamber. Using a pair of parametric arrays consisting of 91 ceramic transducers, they generated source waves of 41 kHz and 39 kHz and 42 kHz and 38 kHz (their transducers had a limited bandwidth of 1 kHz). The resulting 2 kHz and 4 kHz difference waves were measured with SPL of 57.2 ± 2 dB and 49.2 ± 2 dB at a distance of 0 to 0.2 m.

As noted in section 2.5, Ahmadi et al. utilize ultrasound in a unique way; they directly inject it into the mouth of a speaker allowing them to shape it as they would normal frequencies. [37] The same device then records the speech waveform, at ultrasonic frequencies, and uses a linear predictive algorithm on a cellphone to convert the speech to text or frequency shift it back to normal range. The utility is for quiet communication in areas where normal speech is not possible, for "secret" discussions, and for those who have lost normal speech. Hueber et al., turn this process on its head and use a standard B-mode ultrasound probe and a camera to image the vocal tract while lip-reading. The final output is similar to Ahmadi et al. However, this approach does not seem practical as users would require a correctly positioned camera and ultrasound machine (even though both of these can be quite small these days). [72]

5.1. Safety considerations

From a safety standpoint, airborne ultrasound exposure at high SPL has been shown to cause nausea, fatigue and headaches. At levels of 140-150 dB slight heating of the skin can occur. In the 40 kHz range, Japan, Acton, US Dept. of Defense, IRPA, and Canada have exposure limits for continuous airborne exposure at 110 dB. [73] Nagel and Nagel found that low frequency ultrasound could cause blood cell damage at levels as low as 20 kPa (180 dB) and exposure time of under 1 minute. [74] Ahmadi et al. group ultrasound intensities below 3 W/cm^2 into "low" intensity and 5+ W/cm^2 into "high" intensity. They point out that at very low frequencies, cavitation and thermal effects predominate while radiation forces and micro-streaming are minimal. Thermal effects increase with frequency and are dependent on the absorption coefficient of the particular tissue. [75]

6. Simulator Design

The goal of the simulator is to allow us to evaluate the efficacy of our hypothesis before building an actual device. In addition, it allows us to test various inputs (amplitude, frequency, phase, and actuation) as well as placement locations to determine potential settings for testing the physical device. Another important use of the simulator is to allow us to visualize what happens as the waves enter the body; with a physical device this is very difficult if not impossible. It is important to remember that the simulator is only a rough approximation of reality; and while such visualization can be quite helpful, we must not become fixated on a particular set of numerical values as in the real world complex interactions often cause very different results from simulation.

To build a working simulator, we started with a review of recent research on modeling the wave equation with a particular focus on acoustic wave solutions and human body modeling. After finding the AustinMan, which was designed for use in electromagnetic modeling, we requested and were granted access to the dataset.

6.1. Using the AustinMan Voxels

In order to keep the model manageable but still have the highest quality results, we selected the v1.1 Partial Body male model at a resolution of 1x1x1 mm^3; this model includes slices from the top of the head down to the upper chest. Lower resolution models exist as do full body models; however, this particular model has all the areas of interest for our simulation and the tissue segmentation resolution is excellent.

In order to use the data set in our simulator, we developed a conversion program which read the UCD formatted data, sorted the voxels, and converted into our system's data format. In addition, we replaced the electromagnetic tissue properties with the

properties needed for our ultrasonic acoustic simulation. Figure 6-1 shows the converted data set rendered in our 3D renderer at two different opacities and orientations.

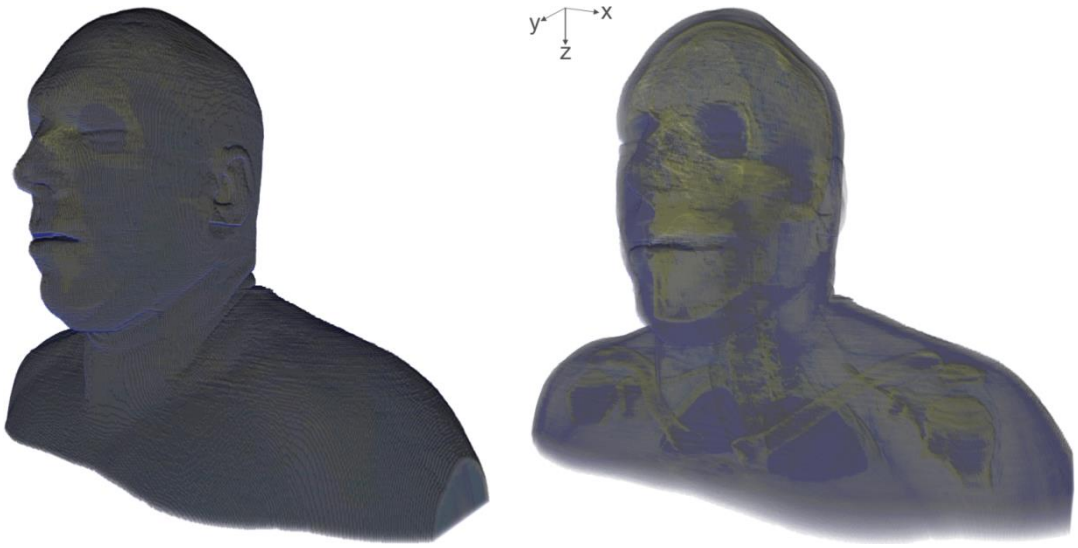


Figure 6-1: AustinMan

The properties required to simulate ultrasound in tissue include density, speed of sound, attenuation, and acoustic impedance. These properties were gathered from a number of sources including Azhari, Duck, Andreuccetti et al., Hasgall et al., McIntosh et al., and NPL's Kaye & Laby and are shown in Table 6-1. [76] [77] [78] [79] [80] [81] [82] The source of each property or derivation is shown in the "Reference" column. The "Voxel Count" indicates how many voxels of this tissue type are present in the model. The following abbreviations are used: *avg* = average, *lin int* = linear interpolation, and *typ* = typical.

ID	Voxel Count	Name	Density (kg/m^3)	Speed of Sound (m/s)	Attenuation dB/cm/MHz	Reference
0	87,692,282	Air @ 25°C, 60% Rel. Hum	1.2	347	0.012	[81] ; [81] ; [81]
1	29,979	Int.Air @ 30°C, 90% RH [83]	1.15	351	0.012	[81] ; [81] ; [81]
2		Norprene Ext. Foam / EP0M	130	1571	42	[84] ; [85] ; [85]
12	822,305	Brain (Grey Matter)	1045	1562	0.625	[80] ; [78] ; [78]
28	159,667	Blood Vessel	1102	1584	0.2	[80] ; [81] ; [77]
32	492,584	Brain (White Matter)	1041	1562	1.05	[80] ; [78] ; [78]
72	324	Cornea	1051	1588	0.5	[80] ; lin int [77] ; *typ*
80	109,897	Cerebrospinal fluid	1007	1528	0.5	[80] ; lin int [77] ; *typ*
84	725,361	Bone (Cortical)	1908	2740	20	[80] ; avg [81] ; [77]
92	619,919	Bone (Marrow)	1029	1560	5	[80] ; lin int [81] ; [77]
96	2,180	Eye (Sclera)	1032	1560	0.5	[80] ; lin int [77] ; *typ*
104	474,305	Lung	722	1125	0.5	avg [80] ; lin int [77] ; *typ*
112	78,158	Cartilage	1100	1729	0.5	[80] ; lin int [77] ; *typ*
116	508	Eye (lens)	1076	1656	2.0	[80] ; avg [81] ; [82]
124	6,322	Dura	1174	1563	0.5	[80] ; lin int [77] ; *typ*
136	1,706,776	Fat	911	1450	0.65	[80] ; [77] ; [77]
144	12,489	Esophagus	1040	1562	0.5	[80] ; lin int [77] ; *typ*
152	176,288	Gland	1050	1570	0.5	[80] ; lin int [77] ; *typ*
156	51,743	Tongue	1090	1562	0.5	[80] ; lin int [77] ; *typ*
168	7,642	Lymph	1035	1570	0.5	[80] ; lin int [77] ; *typ*
172	14,049	Teeth	2063	4695	20	[80] ; avg [81] ; [78]
180	5,492,061	Muscle	1090	1579	3.3	[80] ; avg [81] ; [82]
184	47,956	Spinal cord	1075	1562	0.5	[80] ; lin int [77] ; *typ*
192	19,903	Nerve	1075	1562	1.55	[80] ; lin int [77] ; avg [78]
196	13,281	Eye (vitreous humor)	1009	1528	0.1	[81] ; avg [81] ; [82]
204	911,595	Skin	1100	1729	0.8	[81] ; [81] ; [77]
208	10,926	Mucosa	1102	1570	0.5	[80] ; lin int [77] ; *typ*
216	11,325	Trachea	1080	1729	0.5	[80] ; lin int [77] ; *typ*
220	1,347,600	Tendon/Ligament	1142	1729	4.7	[80] ; lin int [78] ; [78]
		Distilled Water @ 37°C	999.3	1524	0.0022	NIST ; (Marczak eqn) ; [82]
		Soft tissue	1000	1540	0.5	NIST ; [75] ; *typ*

Table 6-1: Tissue Properties

As can be seen from the voxel count, the AustinMan head and neck model consists of over 101 million individual voxels although over 87.5 million of them are air. The other 13.5 million voxels consist of 26 tissue types. The density of each tissue type is fairly well known, but the speed of sound and attenuation vary greatly based on the research used and many are based on the general *soft tissue* type. The speed of sound in soft tissue is generally accepted as 1540 m/s, which is very close to the speed of sound in distilled water at normal body temperature. The attenuation of ultrasound in soft tissues is about 0.2-0.5 dB/cm/MHz. [86] This states that greater distances and higher frequencies result in more attenuation. Many researchers just use the soft tissue values

for the entire model; however, skin and fat have quite different values and are in our primary signal path so we assigned specific values to each tissue type where possible.

Luo et al. describe attenuation as a result of scattering and absorption where absorption is due to shearing motions, heat losses due to conduction, and chemical relaxation processes. [87] They state that sound absorption is a relaxation process which occurs when the equilibrium constant of a chemical reaction is affected by temperature and/or pressure changes. They show that the majority of low-frequency ultrasound absorption in tissue can be accounted for by chemical relaxation and that absorption is proportional to the square of the frequency. Sehgal and Greenleaf show that the relaxation time for tissue is sub-microsecond, and for ultralow-frequency ultrasound the effects are negligible. [88]

The acoustic impedance is of specific concern due to the large impedance mismatch between tissue and the internal air cavity which will result in 99.9% reflection of ultrasound energy. This turns out not to be an issue for our device, as we are not attempting to have sound waves directly penetrate the trachea or pharynx, but rather to cause the tissues themselves to vibrate and through this mechanical coupling, generate a new wave in the vocal tract.

6.2. Rendering the AustinMan Voxels

For the 3D renderer we found an existing MRI renderer developed by Adam Sawicki from Gdańsk, Poland. After speaking with Adam, he agreed to open source his code and shared it on his blog. [89] His code utilizes GPU textures for fast rendering, however, the features and formats did not really meet our needs, so we kept the core GPU functionality and rebuilt the rest to support our data. Our renderer supports six

independently controlled hardware clipping planes (one of which can be seen in use in the right-hand image in Figure 6-4), 3D snap to axis and selected-axis-only rotation, quaternion rotations to minimize cumulative rotation errors, and 3Dconnexion SpaceNavigator™ 6DOF 3D mouse integration for easy navigation in addition to the normal mouse and keyboard. For tissue visualization, the renderer supports density threshold, general opacity, and brightness controls that allow selecting which parts of the model are visible and in what proportions. Using the program we can load, display, and manipulate in real-time all 101 million AustinMan voxels.

To aid in visualization, we used an isomorphic color map which features a monotonically increasing luminance and monotonically decreasing saturation. [90] The color map is based on the normalized tissue density with dark blue representing low density (air) and bright yellow representing high density (bone). Figure 6-2 shows the color map.

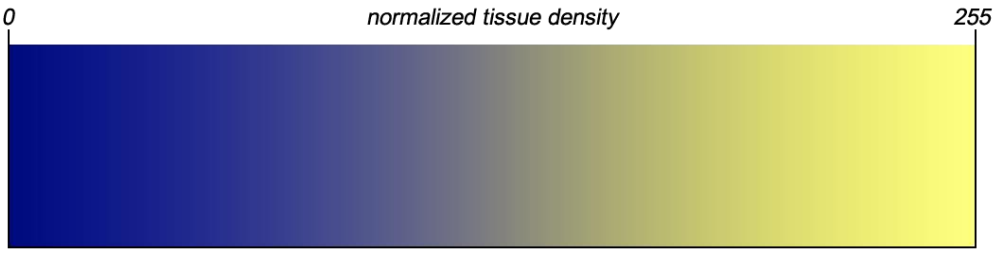


Figure 6-2: Renderer color map

Visualizing the model is an important step. Using the renderer, we can quickly move, rotate, expose interior sections, measure distances, and understand how our simulator needs to interact with the model and how to best attack the complexity of such a large dataset.

6.3. Building an acoustic wave simulator

The next step was to consider the various solutions to simulating the wave equation. Initially, we were very interested in using the pseudospectral method, however, the lack of a fixed grid made it difficult to integrate the existing model. We then considered many commercial FEM packages such as ANSYS®, Solidworks/CosmosWorks, PZFlex, and ADINA and open source solutions such as OpenFOAM, CalculiX, Sailfish, SOFA, and k-Wave. k-Wave was a strong possibility, but the requirement of using MATLAB would make the simulation too slow. Eventually, we settled on doing a simulation using FDTD and began a search for existing tools. We came across the thesis of Ola Brunborg Vikholt. [91] After viewing videos of simulations from his thesis research, we contacted him. He allowed us access to his source code which is a C++ / CUDA (Compute Unified Device Architecture) implementation of a Fortran FDTD simulator created by Shinichi Sakamoto of the University of Japan.

Vikholt's thesis is on "Standing-wave problems in recording studios" and his model implements one material and one wall type and assumes air everywhere else. The code is an extension of his undergraduate work creating a GPU enabled FDTD simulator. The GPU code is written in NVIDIA's proprietary CUDA language. CUDA utilizes the GPU's parallel processors to drastically speed up code execution assuming that the problem can be suitably segmented / parallelized. FDTD is a perfect fit for parallelization as each node in a particular time step is independent of other nodes.

The GPU consists of one or more multiprocessors, each multiprocessor is broken down into a number of CUDA cores. Physically, each core executes a kernel (the code)

on multiple threads. Executing threads are organized into a group of 32 called a warp; only one warp executes on a core at a time. Logically, threads are grouped into blocks, and the collection of all blocks is a grid. In our simulation the entire model would be a grid. The number of blocks is based on the capabilities of the hardware on which it runs. Typically, each block can have 16 or 32 threads; the number of threads assigned to a block determines how many blocks are needed to represent the grid.

After much work, we were able to get Vikholt's code to compile; we had to install the correct NVIDIA drivers, get the CUDA GPU Computing SDK and build it, and then we had to port Vikholt's code to Visual Studio 2010, convert from a 32-bit to a 64-bit platform, and update all the references to the libraries. Once we had an executable, we were able to evaluate for ourselves what it could do and how fast it was. The basics of the code seemed excellent; however, if left running for more than a minute or so, errors starting appearing on the FDTD staggered grid.

After finding and installing NVIDIA's Nsight parallel debugger, we were able to analyze the CUDA code. In a simple mistake, the pressure and velocity update routines were being run by the same kernel. While threads were synced before leapfrogging, this only syncs the threads in a given block. And as the FDTD grid extends beyond a single block, and there are typically more blocks than can execute in parallel at the same time, this means that random areas of the grid were being updated with values from either the previous leapfrog or the next – it all depended on the GPU loading and which blocks executed first. This is a terribly difficult bug to detect, but we noticed that the errors seemed to appear in similar areas – rarely the same area from run to run, but in an ordered chaos. We implemented a visual grid overlay at block boundaries and saw that

errors always started on the boundaries. This allowed us to identify the cause – there is no block level synchronization routine in CUDA. Ultimately, we just needed to split the update kernel into two separate pieces and run velocity first and then pressure. This solved the problem as each kernel runs completely before beginning the next; runs could now go on for hours and in the absence of a sound source, the model correctly settled back to an equilibrium pressure state.

We were now able to test the system with known source values, perform single and double slit experiments and double source interference tests to verify that the simulator was accurately modeling the wave equation. Once correct behavior was verified, we began to customize the code for our simulations.

First we optimized as many functions as possible, removing old or unnecessary code, generalizing routines and extending classes. We also made the code run optimally on our particular hardware by utilizing the CUDA device query to learn what features a particular GPU supported such as the number of threads per block, blocks per grid, etc. We were then able to create a routine which optimizes the GPU "occupancy" to ensure that as many threads as possible run concurrently without blocking. We also extended the CUDA code from the float to double type; this results in about a 33% decrease in runtime speed, but ensured stable runs that can last upwards of 30 – 60 minutes and encompass millions of iterations of leapfrogging. We also extended the FDTD wave equation from a second-order in space and time to a fourth-order in space and second-order in time. This also results in a decrease in runtime speed, but much of the impact can be optimized away by careful coding to ensure that the extra data points are

effectively cached reducing the latency due to memory access conflicts. Ultimately we saw less than a 2% reduction in runtime; we achieve 450 fps on average.

As mentioned earlier Vikholt's code only supported a limited number of hardcoded materials. To extend this, we wrote a lexer/parser to input our material dataset. As the material set could change if our model changed, or more likely if we used a subset of our model, we did not want to hardcode values. Using flex and bison, we implemented a custom parser which can read our material file, parse each value and pass it to the appropriate class for instantiation. The material file associates a given voxel type with the material parameters such as speed of sound, density, and attenuation as well as displaying the material type.

The material file is generated by a custom Perl script which imports our material parameters, parses the AustinMan Voxel file, spatially normalizes values to an integer basis (without loss of generality or resolution as they are all equal size), matches the voxels to our material properties, and extracts both the material type and the referenced properties. The script produces three outputs: the tab-delimited text readable materials file, a binary representation of the voxels (each byte represents a material at a specific location), and a set of bitmap image slices for each z coordinate labeled for easy cross-reference.

With our materials now available in CPU memory, we extended the CUDA code to make them available in the device memory. Synchronization routines were developed so that updates made on the host side are automatically sent to the device at appropriate times to prevent any errors. We then updated the CUDA FDTD calculations to incorporate the specific material information during each node's calculation. Again,

careful coding is required to ensure that memory accesses do not cause the thread to stall. As noted in Section 4.1, we "locate" the speed of sound and density parameters at the same nodes as pressure. Since the speed of sound is only required for the pressure calculation, this isn't a problem. But the density is required in both pressure and velocity calculations. To correctly solve this, we average the densities between the two pressure nodes that a particular velocity node lies between.

Finally, we updated the CUDA display routine to account for spatially dependent speed of sound and density (otherwise, the display looked completely red as it was originally optimized for air only which has a low speed and very low density), and display a slice from our binary voxel representation. In order to correctly display the image on the grid and overlay the pressure/velocity plots, we implemented an alpha blending routine which correctly layers and blends all the visual information and places a single color value into the video buffer for display. We later extended this to include gridlines as a top most layer as it is often helpful to have a spatial reference when examining the results – this option may be turned on and off at any time.

We now had a stable system, but with only a single source; new classes were added to generalize both the sources and microphone. The system now supports a compile time option for a maximum number of sources, each source can be enable/disabled, have its amplitude, frequency, and phase changed at any time before, during, and after a simulation.

The system ran stably and quickly, but the waves were not behaving correctly; the issue seemed to be located near the teeth. The problem came down to an issue with the sampling rate and spatial spacing. The spatial spacing should be 5-10 times less than the

smallest wavelength. Equation 3.30 relates frequency to wavelength and the speed of sound. The minimum wavelength will be given by the lowest speed of sound and the highest frequency in our system. From our materials file, we know that the minimum speed of sound in our model is for air with a value of 347 m/s. We are interested in waves in the 40 kHz range and our spacing should be 5-10 times less, so 347 / (8 · 40kHz) results in a spatial step in the millimeter range which is a perfect fit for the model. Therefore, we set $\delta = 0.001\text{m}$.

The maximum stable time step is calculated from the Courant number as shown in equation 4.7. We know the spatial step, and we know from our materials file, as shown in Table 6-1, that the maximum speed of sound is for teeth with a value of 4695 m/s. Δt is therefore equal to 123 ns, so our sampling frequency is 8.13 MHz. However, as we want to be able to record audio at 96 kHz, we need the sampling frequency to be a multiple of that, so we pick 85 as the multiple which results in a system sampling frequency of 8.16 MHz.

Running the system now results in stable waveforms which do not misbehave as they encounter teeth. However, we cannot easily view the system state as it never leaves the GPU; pulling data from the GPU to the CPU (device to host memory copy) is a fairly slow operation. Viewing the GPU memory in the debugger is possible, but it tends to make the system unstable; debug points must be selected carefully as a second run through will most likely not work correctly. A quick solution, which we began implementing above, is to copy the microphone value once every 96 kHz (every 85 leapfrog cycles). This was implemented in the GPU which can store a compile time fixed number of milliseconds worth of samples and copy them on request to the CPU which

generates a WAV file at the base sampling frequency. Initially, we used a parallel reduce to average the 85 cycles; this seems like a better solution than just taking one sample every 85 cycles. However, the output contained massive aliasing, so rather than implementing a FIR low pass filter and wasting GPU cycles, we opted for decimation and just store the single value. The output is now quite clean, and we can visualize the system status at the microphone at any time using any audio program we choose. This worked well in the beginning, but it is slow to output a file, load it up in another application, and then view it. We needed a way to visualize the waveform within the simulator itself.

We wrote a custom CUDA kernel which implements the STFT to generate a spectrogram as described in Section 3.4. The DFT and IDFT were performed using the NVIDIA CUFFT library, and the window function, normalization and display were done in CUDA. One problem with this is that the system has no concept of windows; rather than try to rewrite the entire front-end display, we opted for some complicated resizing code which extends the pixel buffer object (PBO) to leave room for the spectrogram to be drawn directly below the simulation.

Given that our system has a sampling frequency of 8 MHz, we need to calculate how many samples to run through STFT so that our frequency bins are useful. We would prefer to have the bins be about 1 kHz. Dividing f_s / 1 kHz yields 8160, however for optimal execution of the DFT, this should be a power of two – that is more important than having bins of exactly 1 kHz. So we round 8160 up to the next power of two which gives 8192. Given the sampling frequency and number of samples in the DFT, our bins

are 996 Hz wide. The CUFFT DFT yields only the non-redundant Fourier components; furthermore as our signal is real, we can halve the work by using the real DFT version.

To generate the spectrogram, we record a sample from the microphone at each calculation step. While the number of samples in our DFT is 8192, we using a sliding window with 50% overlap, so for the first run the first 4096 entries will be zero, in the next run the upper 4096 samples are shifted down (in actuality, no data is moved, rather we index into the sample array which is much faster although tricky to implement). After every 4096 time steps, we apply the window function to the data, run the DFT, use a parallel_norm to find the data norm, and then normalize the output and clamp values to between 0 and the number of entries in our color map.

We implemented a number of window functions such as the Hanning, Hamming, and Blackman-Harris, and the selection is a compile time option, however, in practice we utilize the Blackman-Harris window function as it reduces side-lobe levels.

The color map was generated from a custom Perl script which utilizes a HSL-to-RGB conversion routine allowing easily blending colors around the color wheel by varying the Hue over the width of the bar. The map was carefully designed to maximize the number of colors. This map has 1024 entries and consists of two blended ranges. This allows us to visually indicate the magnitude of each frequency bin. The color map is shown below:

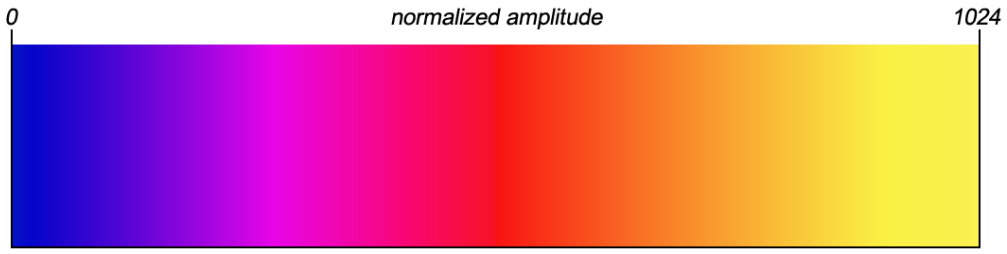


Figure 6-3: Spectrogram color map

Running the simulation now generates a new spectral bar every 4096 samples which is about every 500 μs of simulation time. During the simulation, we can now see our input frequencies. However, we do not see the difference frequency. When we output the waveform and examine it in an audio package such as Adobe Audition, we can see that the time-domain amplitudes are modulated which occurs with a beat frequency. This is one of the features of the DFT, it allows you to see the constituent parts of your signal; it will show the base (carrier) frequencies and sidebands not the envelope. In order to visualize the difference waveform, we must extract the envelope. Using the Hilbert transform as shown in equations 3.36, 3.37, and 3.38, we created a filter which when recombined with the original signal returns the envelope. This turns out to be quite easy to integrate into our existing spectrogram code. The one implementation hazard is that the DFT returned by CUFFT is non-normalized; therefore, we must divide each bin by the number of samples (N=8192) otherwise each bin from the IDFT will be N times larger than the original waveform which will not give us the result we want when we combine them. The envelope waveform can now be run through the same spectrogram code as we used before, and we can visualize the envelope which will show our difference wave / beat frequency.

6.4. Simulator Validation

Using the 3D renderer, we created a number of visualizations of the AustinMan voxel dataset. Figure 6-1 shows the exterior of the model. The left figure has 100% opacity while the right figure has threshold density and opacity setting to allow the skeletal system to show through. Figure 6-4 shows a sagittal view of the head. The left image is from Grey's Anatomy while the right image was generated with dual clipping

planes on either side of the midline from the AustinMan voxels using our 3D renderer. In Figure 6-5, a CT with contrast dataset is used in the renderer and the same sagittal view is shown for comparison. All three are visually similar which is encouraging that we have correctly extracted, processed, scaled and rendered the AustinMan voxels. Comparing the rendered image with the Grey's Anatomy diagram, we can identify specific features and verify scales; comparison with an actual CT scan confirms that the proportions and details are "visually" correct.

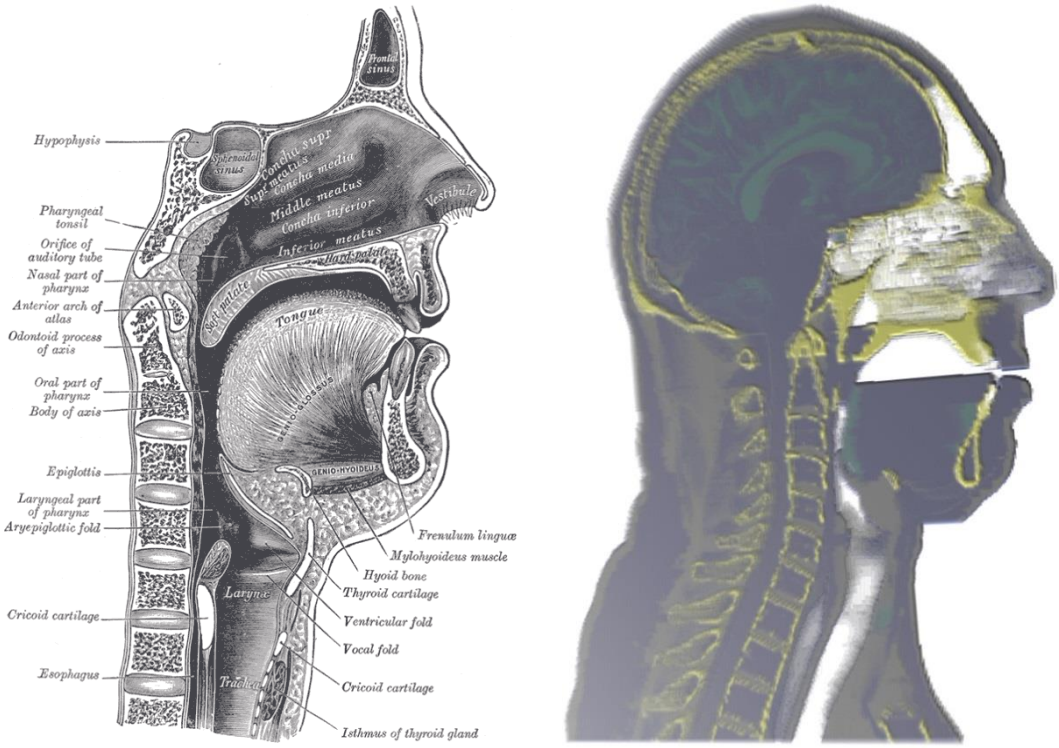


Figure 6-4: View of the AustinMan Trachea [14]

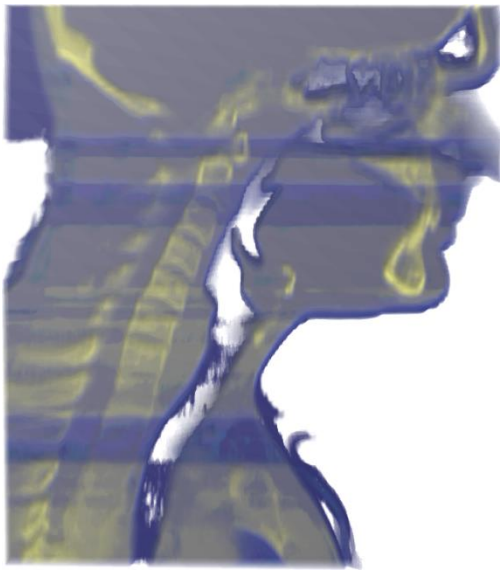


Figure 6-5: CT Scan of Trachea

In Figure 6-6, a transverse view is shown, again with dual clipping planes. We can clearly see the trachea and measure its cross section as 19 mm x 17 mm in the sagittal and coronal axes, which is smaller than one might expect based on the average sizes discussed in Section 2.1, but that is because this cross section is of the lower trachea about 1.5 cm below the vocal folds.

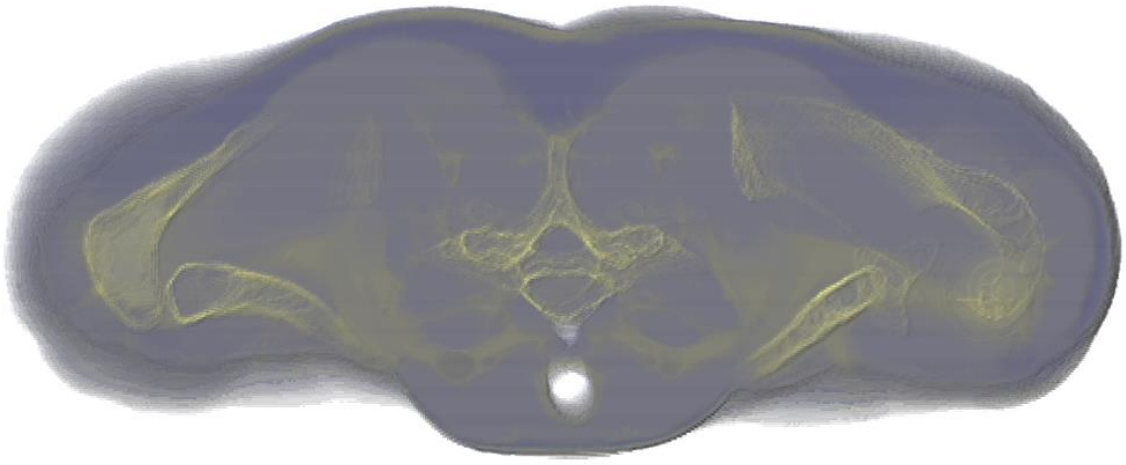


Figure 6-6: AustinMan transverse view

For the FDTD simulator, we started with some basic validations of wave interaction. As it may be difficult to identify particular points on the figures, we have included Table 6-2 which summarizes the experiment, analytical solution, and observed FDTD value. The errors are all due to the discretization of the grid – the minimum resolution is 1 mm.

Figure 6-7 shows the classic double slit experiment. The grid is 1 cm x 1 cm and each pixel is 1 mm^2. To ease visual validation, we selected a frequency of 34.7 kHz which by equation 3.30 gives a wavelength of 1 cm in air (where we take the speed of sound in air to be 347 m/s as noted in Table 6-1). Equation 3.34 allows us to calculate the locations of the maxima and minima using some basic trigonometry. Given a slit opening distance of 24mm, if we assume the vertical distance, L, from the slit boundary to some point P is 3 cm, we can calculate the horizontal distance perpendicular to L where the maxima and minima will occur. In the figure we have indicated the first two maxima (P_y = 0 cm, ± 1.375 cm) with upwards pointing fuchsia arrows; the first minima (P_y = ± 0.64 cm) is indicated with a downward pointing red arrow. As the simulation advances, the pattern is easy to see; pausing on gridlines makes comparing calculations with the visual output easier.

Experiment	**Index**	**Analytical Solution**	**FDTD value**	**Notes**
double slit	n = 0	(0 cm, 3 cm)	(0 cm, 3 cm)	maxima – first
double slit	n = ±1	(±1.375 cm, 3 cm)	(±1.4 cm, 3 cm)	maxima – second
double slit	n = ±1	(±0.64 cm, 3 cm)	(±0.6 cm, 3 cm)	minima – first
single slit	n = 0	(0 cm, 3 cm)	(0 cm, 0 cm)	maxima – first
single slit	n = 0	7.5 cm	7.6 cm	first maxima – width
single slit	n = ±1	(±37.5 cm, 3 cm)	(±38 cm, 0 cm)	minima – first
interference	n = 0	(0 cm, ±x cm)	(0 cm, ±x cm)	center maxima
interference		(±0.196 cm, ±1.8 cm)	(±0.2 cm, ±1.8 cm)	minima

Table 6-2: Validation Results

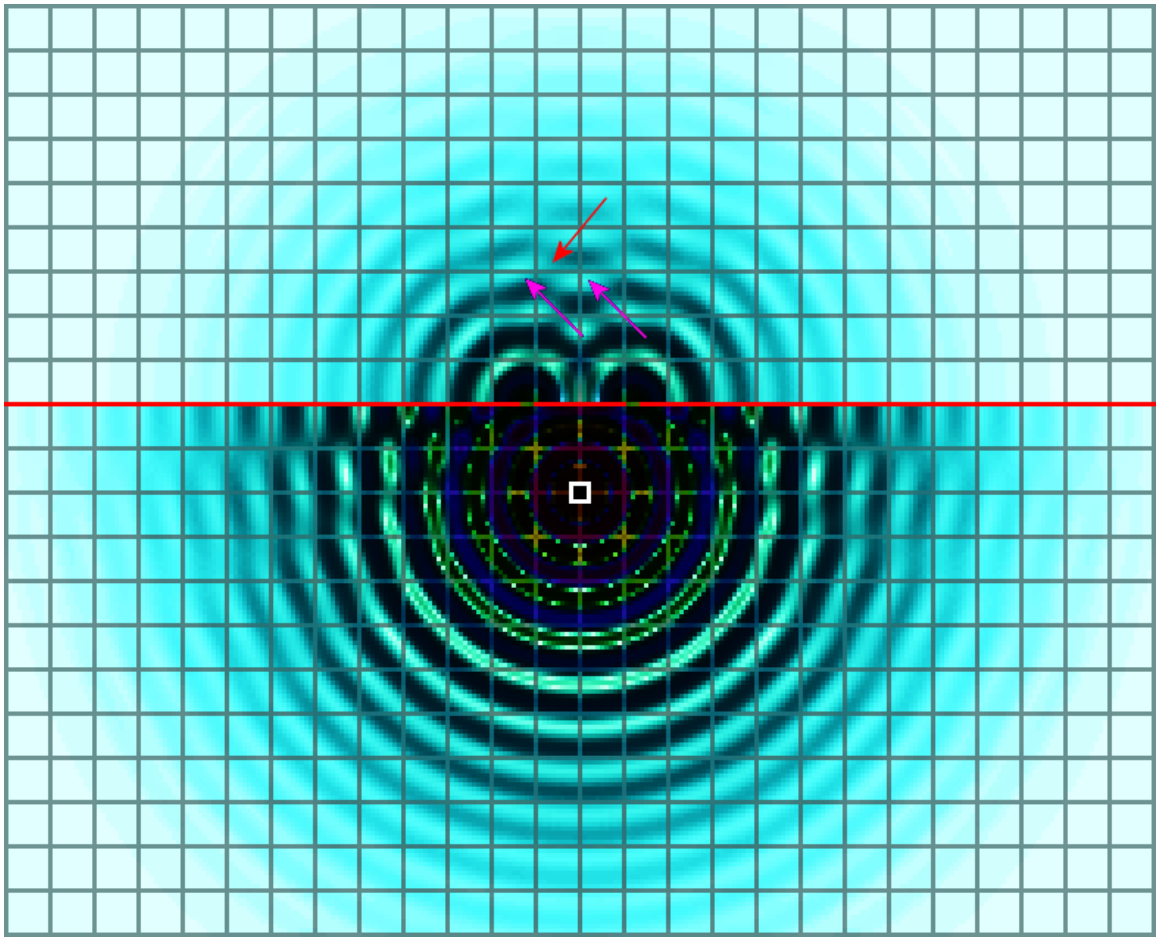


Figure 6-7: Simulator double slit validation

Next, we performed the single slit experiment. At these wavelengths it is more difficult to visually validate. In Figure 6-8, we have turned off the gridlines, raised the amplitude, and increased the frequency to 48 kHz to aid visualization. The gap is set to 0.8 cm. The primary maxima occurs dead center as expected and has a width of 7.5 cm at a height of 3 cm.

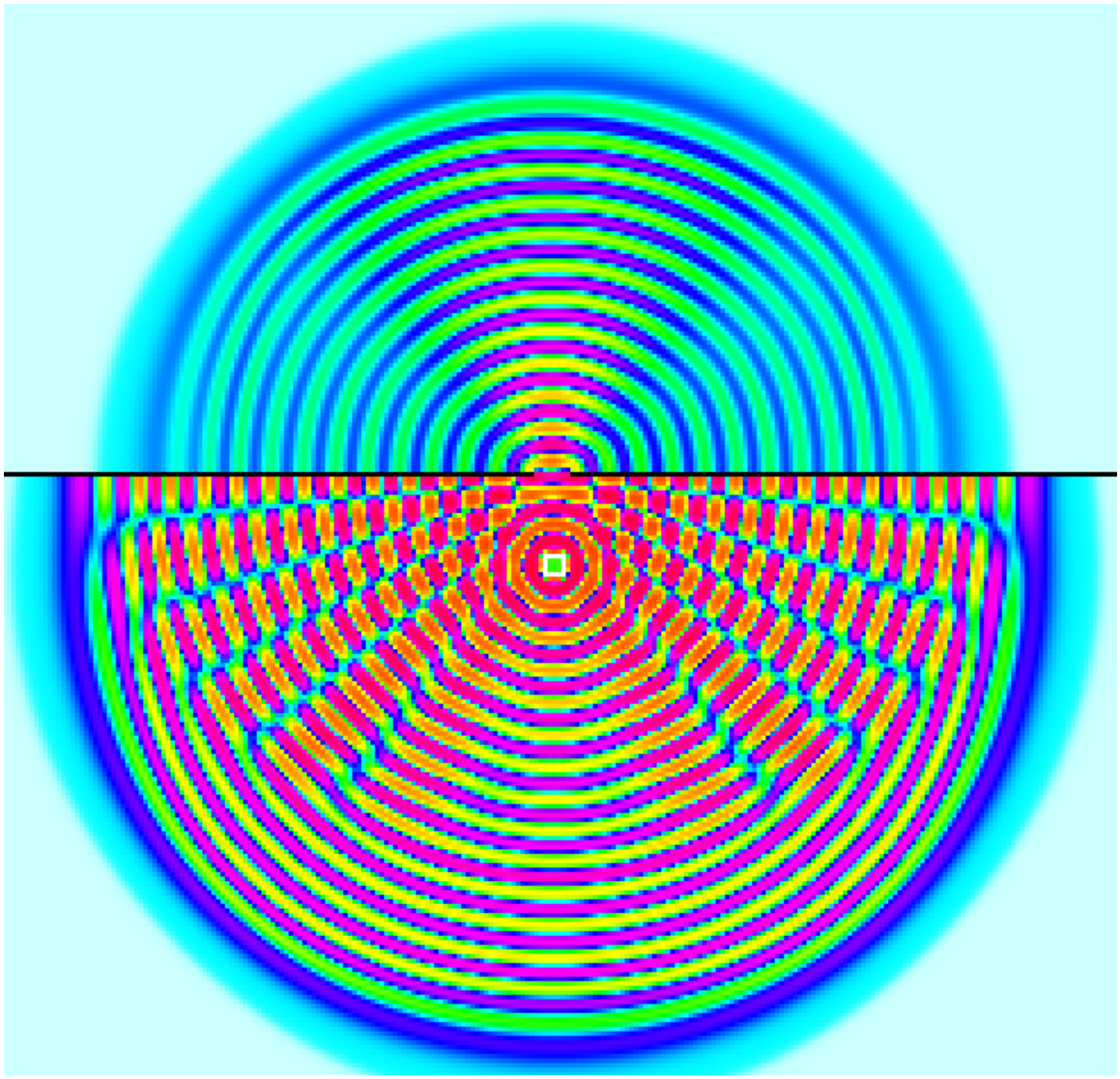


Figure 6-8: Simulator single slit validation

Finally, in Figure 6-9, we have placed two 40 kHz waves 4 cm apart. Source 1 is located at (-2, 0) and Source 2 is located at (2, 0); both have a wavelength of 0.8675 cm. As expected, maxima occur whenever the distance from Source 1 and Source 2 are equal. We have indicated a minima in the figure at (-0.2, -1.8); the difference between the two lines shown is ½ wavelength (0.43375 cm).

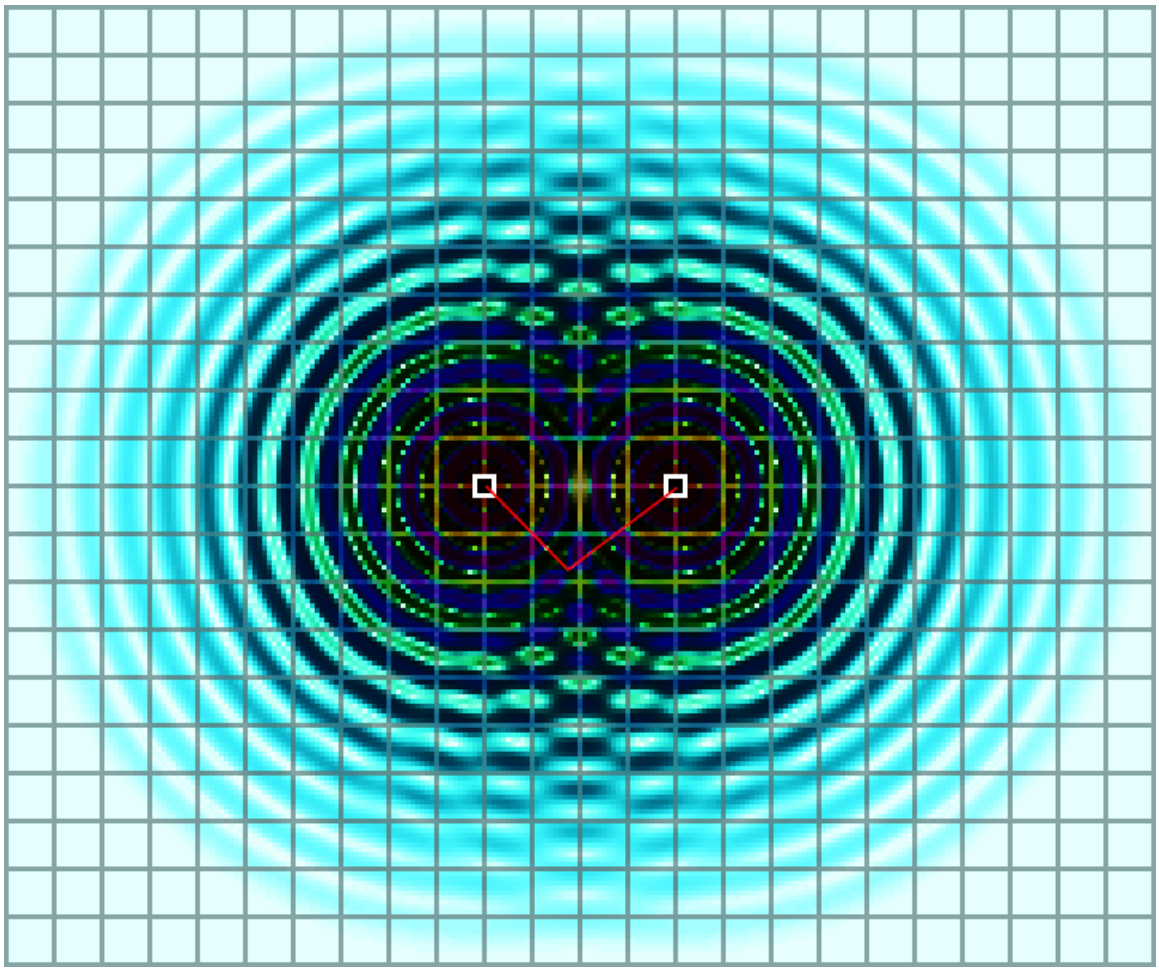


Figure 6-9: Simulator dual wave interference validation

Finally, we need to validate that our spectrogram and envelope detection system is functioning. The left image in Figure 6-10 shows the spectrogram produced by the simulator for a 30 kHz wave and a 40 kHz wave spaced 6 cm apart with the microphone located in the middle.

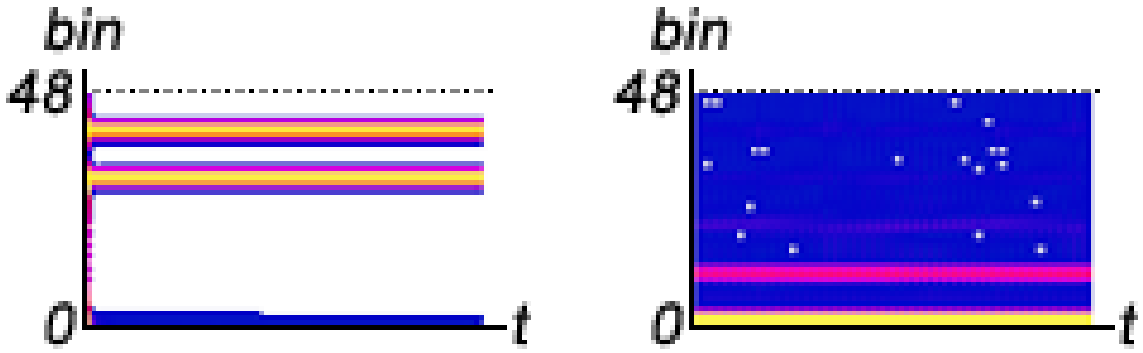


Figure 6-10: Simulator Spectrogram (left) and Envelope Spectrogram (right)

Each vertical pixel represents one bin. As previously discussed, we have set the STFT values so that each bin represents 996 Hz. Therefore we expect to see a bright line (a color from the right hand side of Figure 6-3) at y=30 and another at y=40 (bin 0 is the DC offset). And this is exactly what the display shows along with some spectral leakage that is typical of the window function. As this display is designed for visual feedback only – we expect to use a professional audio package for detailed spectrographic analysis – the display contains minimal information/interactive controls. However, we are able to verify the correct output by a.) using the debugger to examine the output bins, b.) dumping the bins to a text file for offline verification, and c.) exporting the captured audio and comparing the results in Adobe Audition. For this example run, we performed all three of these steps. The exported the audio data was opened it in Adobe Audition; Figure 6-11 shows the spectrogram run using a 8192 sample Blackman-Harris window function. Other than the higher resolution, the output is very similar with some leakage. Notice that the spectrogram in Figure 6-11 shows both waveforms at exactly their declared values.

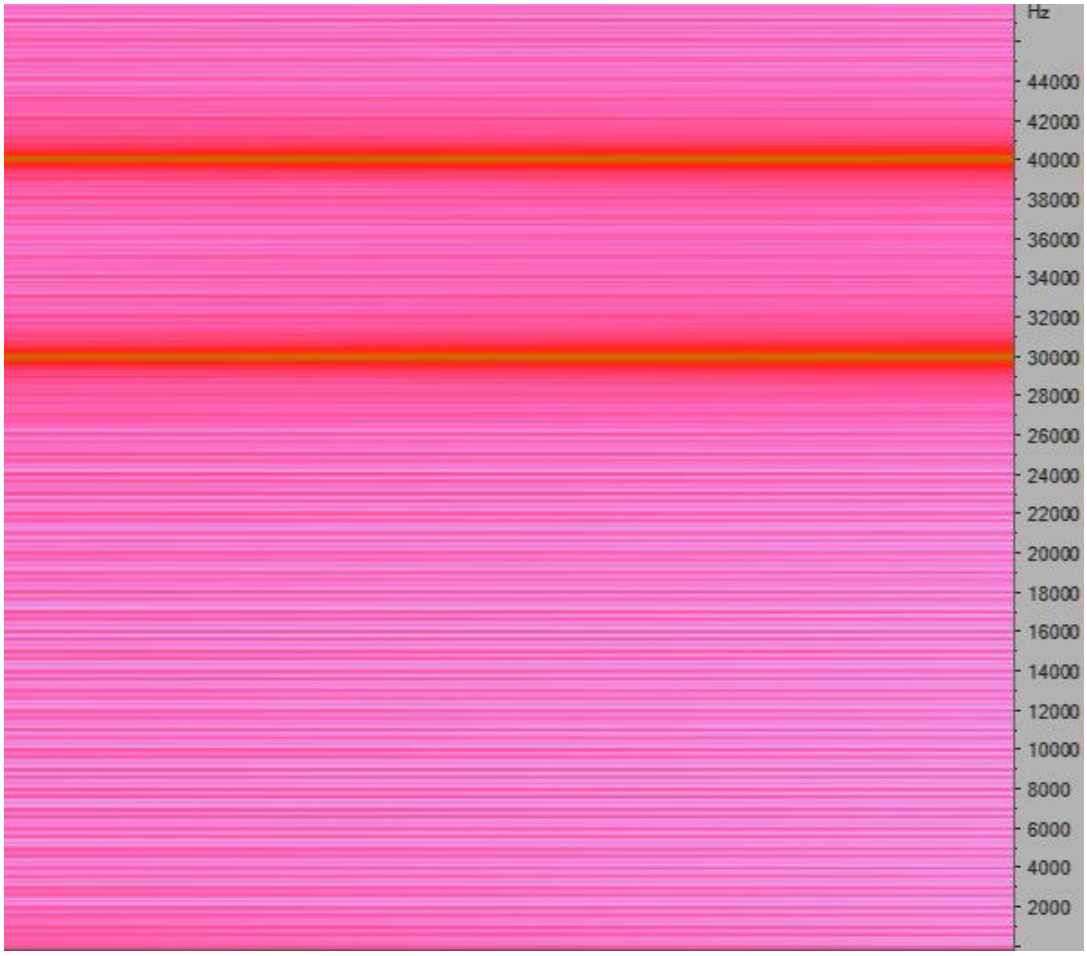


Figure 6-11: Adobe Audition Spectrogram

The right image in Figure 6-10 shows the Hilbert transform envelope spectrogram. We verified that the bright pink line is at y=10 (bin 10 → 9.96 – 10.95 kHz) along with some worse spectral leakage caused by two STFTs and one ISTFT. While Adobe Audition can modulate and demodulate signals, it does not perform envelope detection, so there is no comparable output to show.

6.5. Simulation Results

The simulations were all run on a Windows 7 64-bit machine with an Intel i7 2.8 GHz 8 core processor, 24 GB of RAM, and a NVIDIA GeForce GTX 460 video card with driver version 5.0 and runtime version 4.2. The video card is a CUDA Capability

2.1 card with 1GB of global memory, 336 cores running at 1.3 GHz, a warp size of 32, and a maximum of 1024 threads per block.

A video of the simulations can be viewed at the following URL:
http://goo.gl/HosIjH

For simulations, we selected two slices of the AustinMan voxels in different orientations. The first is a transverse section at z = 27.9 cm from the top of the head as shown in Figure 6-12. Two sources were configured spaced 2 cm apart. The model was adjusted to allow the sources to be tucked up against and almost surrounded by skin on three sides in a similar manner in which an electrolarynx is pressed against the neck. The sources have frequencies of 36036 Hz and 43243 Hz (which match a run of the physical device discussed later); the second source is 90° out of phase with the first source. The microphone was placed inside the vocal tract.

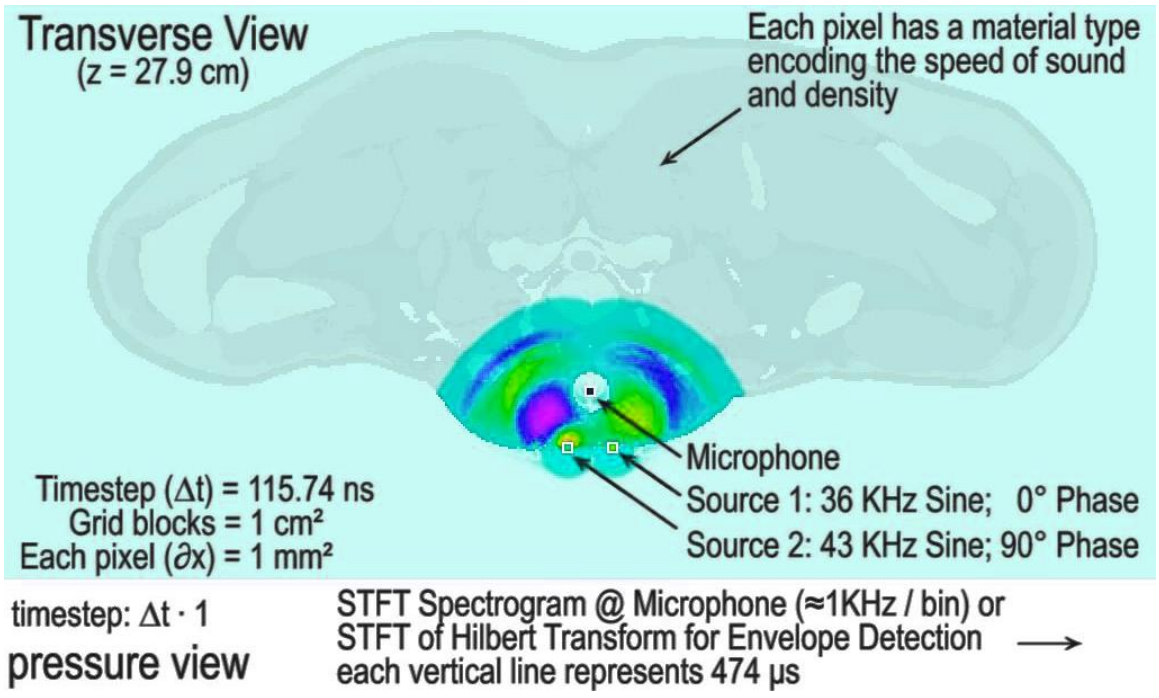


Figure 6-12: Simulation Run 1

The model was run for 200 ms of simulated time, during which the 7 kHz difference wave was detected as shown in the left hand side of Figure 6-13. The right hand side shows the velocities of disturbances permeating through the body after both sources were turned off and the pressures began returning to equilibrium.

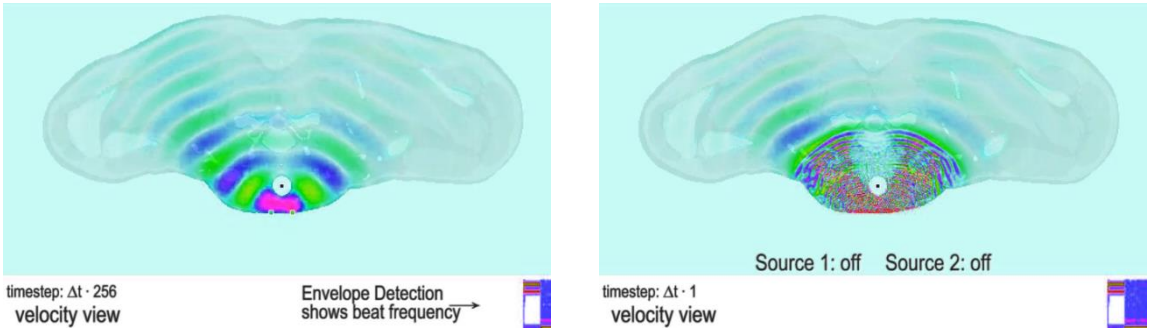


Figure 6-13: Simulation Run 1 - additional views

The second run is a sagittal section near the midline at x = 27.1 cm as shown in Figure 6-14. As in the first run, two sources were configured spaced 2 cm apart and tucked up against the neck, the sources have frequencies of 36036 Hz and 43243 Hz, and the microphone is placed in the trachea just below the vocal folds. All images of the sagittal view show pressure compared to several velocity views in the transverse images. In Figure 6-15, the left hand image shows how far the pressure wave has made it through the body as well as self-noise reflecting from the device skin boundary; it should be noted that this noise is quite low and also mostly high frequency as by the time the waves interfere they have been severely attenuated and the resulting audible noise is miniscule. This will be further limited in the physical device with acoustic EP0M foam padding.

The right hand image shows the resulting pressure wave when both sources are turned off. The spectrograms of the sources and the envelope are also visible.

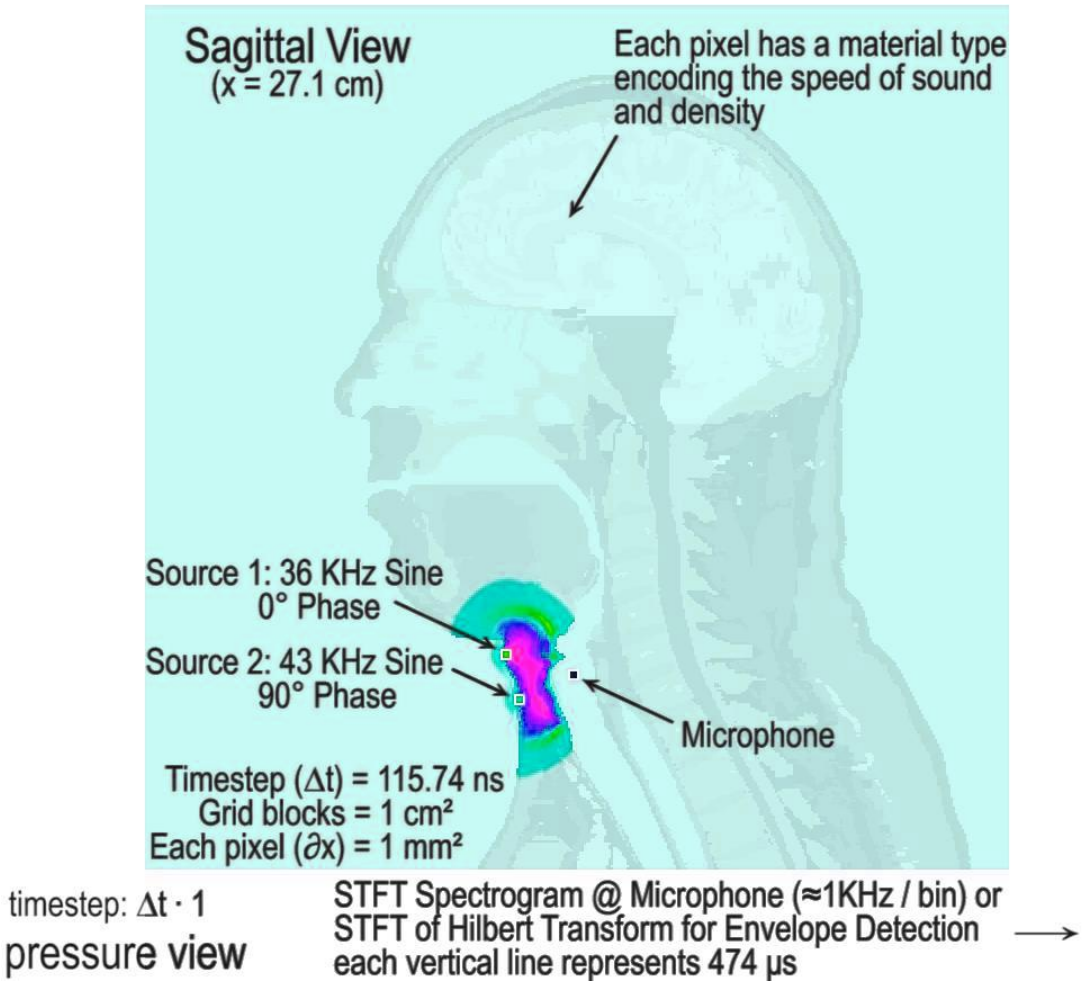


Figure 6-14: Simulation Run 2

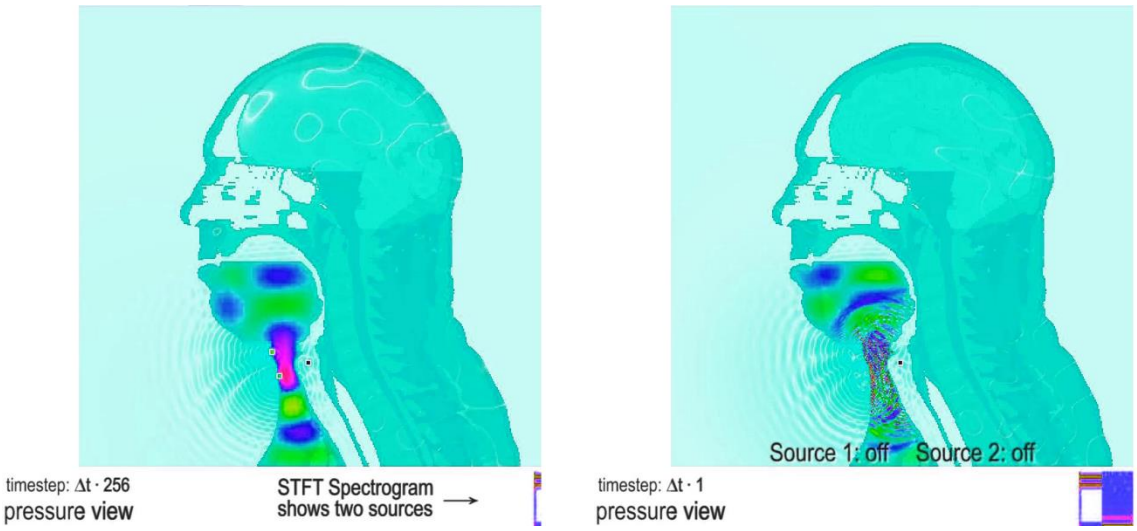


Figure 6-15: Simulation Run 2 - additional views

While neither of these two runs generated a frequency suitable for use as a fundamental frequency, the results were encouraging. The waves penetrated the body and even at a low amplitude setting were strong enough to create an interference pattern at the trachea boundary to create a detectable difference wave.

In Figure 6-16, we show the output from a recent run of the simulator. The sources were spaced 2 cm apart in the transverse cross-section at z=27.9 cm with frequencies of 39.7 kHz and 40 kHz. After running for 500 ms of simulated time, the waveform was output, demodulated, and compared to a reference wave generated in Adobe Audition's "Generate → Tones" feature. The top wave is the 300 Hz reference wave, the bottom is the demodulated output. The waves have been positioned so that they have the same starting phase; this is not maintained throughout, but output is remarkably close to the reference.

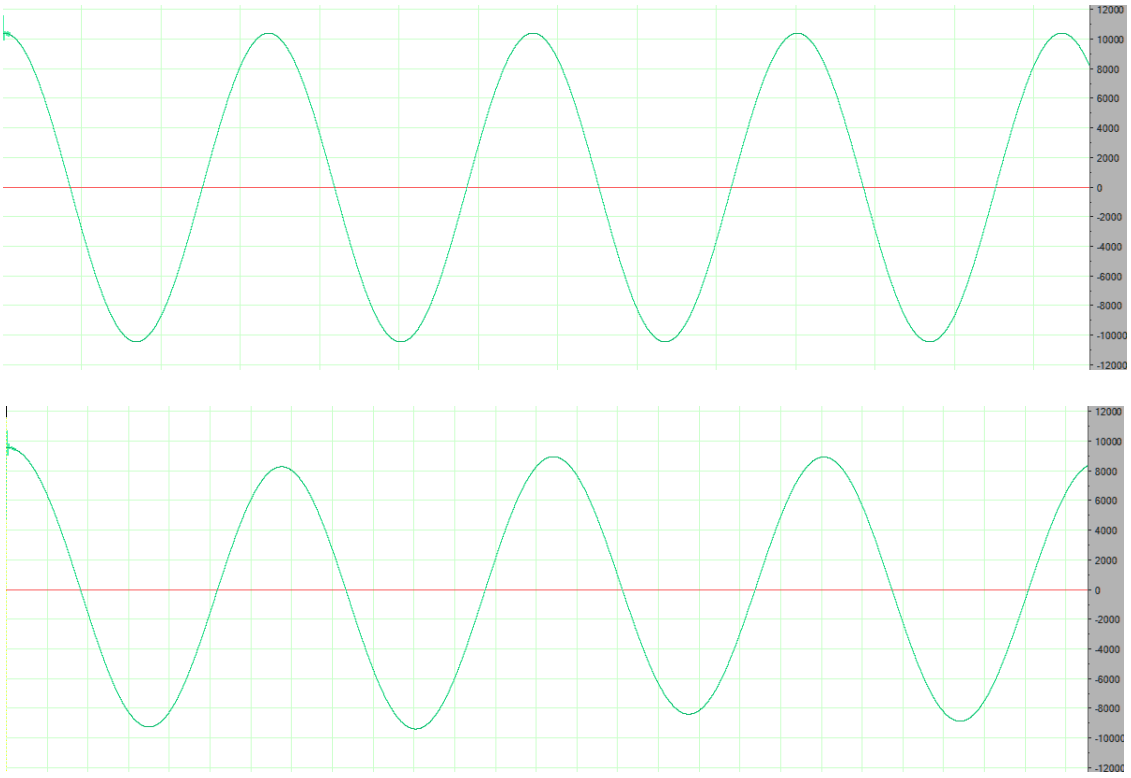


Figure 6-16: Simulation 300 Hz difference wave output

The simulation results have served one of their primary purposes which was to show that the approach was feasible and worth investing more money and time into building a working physical device. The other goal of the simulator was to help identify potential source locations, amplitudes, frequencies, phases and timings that would allow the physical device to work at a near optimal level; in that regard the simulator will continue to be improved and utilized.

7. Full 3D Simulation

While the simulator presented in the previous section is fully 3D capable, we are only calculating a single slice of the model in a given simulation. In order to gain greater insight into how a physical device would work, we need to simulate using all layers.

Before continuing, we must define the 3D model coordinate axes to facilitate a coherent discussion. We use a right handed coordinate system, as shown in Figure 6-1, where in a ventral view of the coronal plane: the z axis is the intersection of the coronal and sagittal planes and increases positively from superior to inferior, the x axis is the intersection of the coronal and transverse planes and increases positively from left to right lateral, and the y axis is the intersection of the sagittal and transverse planes and increases positively from posterior to anterior.

7.1. Extending the simulator

The simulations presented in Section 6.5 used a single slice with dimensions optimized to fit the smallest x and y dimensions divisible by 32, the warp size. For the transverse view, the domain was 512 x 256 x 1 mm^3. However, the maximum dimensions of the AustinMan v1.1 Partial Body male model voxels are 535 x 285 x 353 mm^3; therefore, to accommodate all slices, the transverse size needs to be increased. In order to simplify the resizing and maintain divisibility by the warp size, we exported and visually examined all transverse slices and identified where the maximum extents occurred. If we limit the number of slices to the first 319, we can fit every slice within a 512 x 288 mm^2 area with a 2 mm border on both x boundary edges and a 3 mm border on both y boundary edges. We can then insert an air slice at the first slice for a 1 mm border and have 320 total slices which is also divisible by our warp size.

We updated our Perl script to support cropping and minimum/maximum settings; the binary voxel output now includes all 320 slices and is 45 MB in size. While we were at it, we had the script redefine the air voxels within the airway with a unique type. This allows us to treat them as standard air just like we have outside the body or as a more hot and humid air typical of a living person. The new type is shown in Table 6-1 as ID 1 and encompasses about 30K voxels. This is the only change we made to the AustinMan voxels as we hope other researches will validate, compare and extend our results and having an unmodified model makes that much more likely and easier.

When we attempted to load the updated voxels into our simulator, we immediately encountered a problem. We did not have enough device memory to support the entire model. Our simulator needs space for the voxel types (1 byte per node) and for 4 doubles which represent the pressure and velocity components at each node. Each double is 64-bits, so at a minimum we need 72-bits times the dimensions or almost 1.5 GB of device memory. The NVIDIA GeForce GTX 460 only has 1 GB of device memory. There are four possible solutions: switch from double to float, use host mapped memory, segment the slices, or install a video card with more device memory. We explored all four options.

First, we tried the fastest and cheapest solution which is to switch from using double precision floating point to single precision floating point. This immediately nearly halved our device memory needs down to a manageable 0.75 GB. However, we did not view this as a long-term solution as we were unsure if we would need the additional precision as we continued to extend our simulator and there wasn't much room for additional data. Adding one more single precision float at each node would push our

requirements to almost 1 GB and some space is always required by other processes such as the operating system. Therefore, we rejected this solution; and as will be shown in the next section, this choice was the correct one as we did indeed need to add more data to each node to support a nonlinear model.

All CUDA devices with a compute capability greater than 1.0 can map host memory into the device memory space. This allows for zero copy access to the host's RAM. As our machine has 24 GB of host memory, this was an intriguing option. We implemented a version of the simulator that loaded the voxels into host memory and then locked them and shared a pointer with the device. We were able to successfully run the simulation, but a simulation of 50 ms would take over a month of computation. With some optimizations we were able to speed things up, but even with optimized access and efficient caching, a run would still take 24 days. As we want to test many different scenarios, this amount of compute time is unacceptable.

A step beyond a full host mapped memory solution is to segment the slices into subsets and transfer those to the device for efficient calculation. We developed a routine which analyzes the device properties and calculates at runtime how many slices fit into device memory. The simulator then swaps segments between host and device memory. One complication with this setup is that the slice being displayed may be swapped out and the slices at the beginning and end of a segment depend on slices no longer in device memory. To solve this issue, we implemented a set of halos which are extra slices kept in device memory. For fourth-order central difference calculation, we require two slices above the current segment and two slices below. In addition, we added another slice for the currently displayed slice so that it was always in device memory. Using this

configuration, we could complete a 50 ms simulation in less than 9 days, a 266% speedup. This is much better but still not acceptable, and there is very little room to improve as the device must copy the entire dataset, about 1.5 GB, to and from memory every time step; there is a fundamental limit to how quickly the device and host can move 3 GB across the bus even at 16x PCIe 2.0 speeds. Therefore, we opted to spend some money and see what performance would be gained from having the entire model in memory.

To investigate the final option, we purchased a NVIDIA GeForce GTX 660 Ti video card. This card is a CUDA Capability 3.0 card with 3 GB of global memory, 1344 cores running at 1.1 GHz, a warp size of 32, a maximum of 1024 threads per block, and runs with driver version 5.0 and runtime version 4.2. With no modification to our simulator, we were able to use this card to load the entire model into device memory. A 50 ms simulation run now takes less than 17 hours to run; a 12x speedup over the best alternative.

7.2. 3D Simulation Results

To align with the physical device presented in Section 9, we have setup the simulation as follows: Source 1 is 40 kHz @ 0° phase and located at (249 mm, 202 mm, 278 mm); source 2 is 39.6 kHz @ 90° phase and located at (269 mm, 202 mm, 278 mm). The recording microphone is located at (259 mm, 179 mm, 278 mm) which is within the trachea and equidistant to each source. Figure 7-1 shows the two sources and microphone as square icons. The shading indicates the intensity of the pressure at the node's location (screen shot taken at 67 ms). To examine the acoustic leakage, we also

ran the simulation with the recording microphone located at (259 mm, 252 mm, 278 mm).

We should note that the location of the sources and microphone are in the subglottal region below the larynx, which in our model is intact. As our goal is to determine whether or not a fundamental frequency can be generated in the vocal tract using a beat frequency, we do not view this as a problem. We view the ability of other researchers to use an unmodified AustinMan dataset to duplicate our simulation as more important. Furthermore, simulation through tracheal cartilage provides a worst-case scenario; patients having undergone a laryngectomy will have much less dense tissue in the signal path. Similarly, while radiation therapy often results in fibrosis and edema which hardens the neck tissue, these effects have been ignored in this simulation as the effects eventually subside and the neck tissue softens. [20]

Commercially available low-frequency ultrasonic transducers have a typical radius of 5-10 mm. We model 6 mm radius ultrasonic transducers as sinusoidal hard point sources; to simulate the hemispherical nature of the transducers we included a 4 mm EP0M baffle behind the sources.

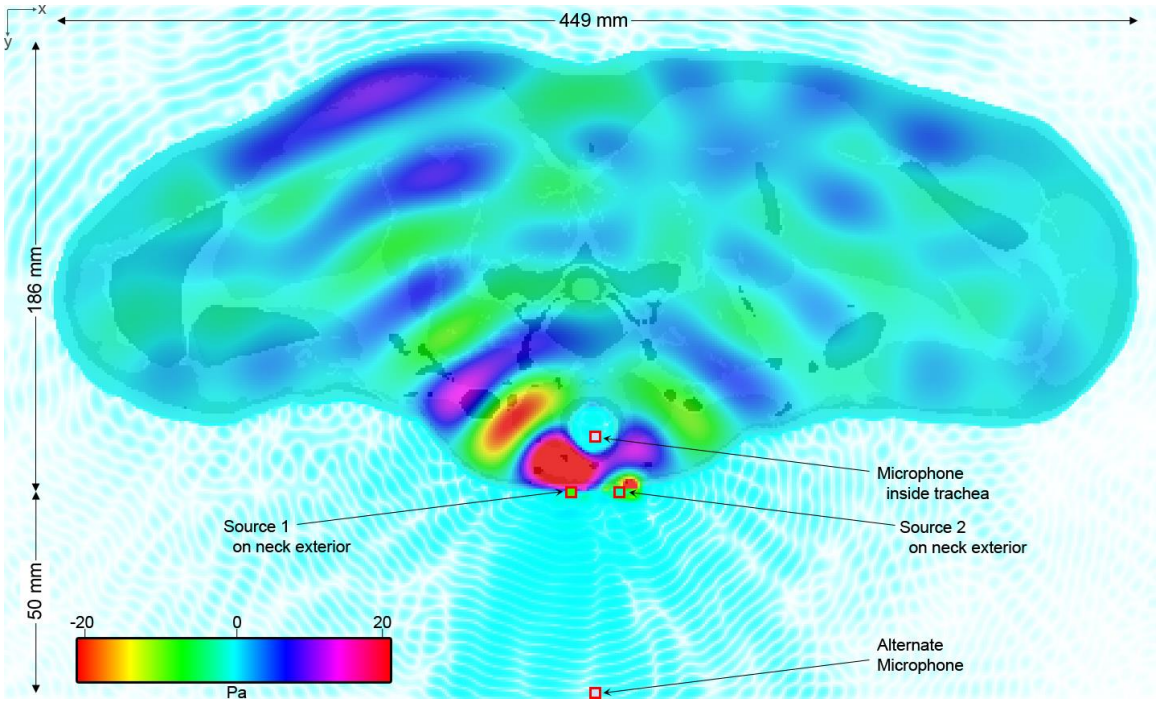


Figure 7-1: Transverse view of simulation pressures at z = 278 mm, t = 67 ms

The simulation was run for 545K iterations or just over 67 ms of simulated time. This is more than enough time for the pressure waves to propagate throughout the entire model and reach dynamic equilibrium. At this stage, the double-precision floating point pressure measurements at the microphone location were exported to a 96 kHz 16-bit PCM wave file.

Viewing the time-domain amplitude waveform, the beat frequency presents as an amplitude modulation of the two input frequencies. In order to demodulate the low-frequency signal, we take the envelope of the waveform using a Short-time Fourier Transform (STFT) version of the discrete Hilbert Transform. The recorded and demodulated signals are shown in Figure 7-2.

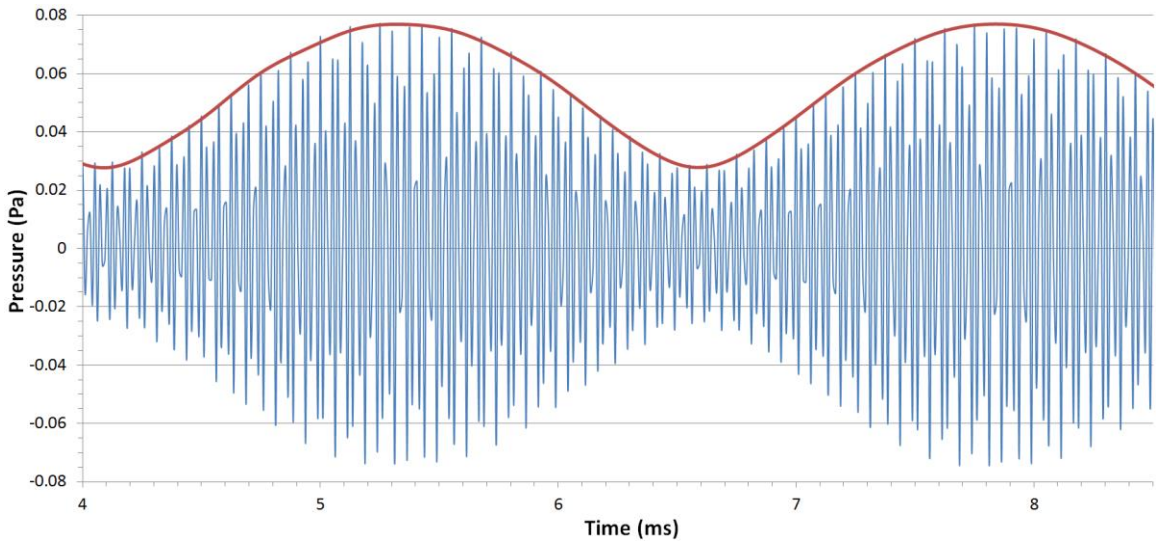


Figure 7-2: The recorded waveform showing the amplitude modulation indicative of a beat frequency and an overlay of the demodulated signal showing the beat frequency

The spectrum of the demodulated signal is shown in Figure 7-3. The 400 Hz beat frequency has the highest amplitude, followed by the harmonics at 800 Hz, 1200 Hz, 1600 Hz, and 2000 Hz.

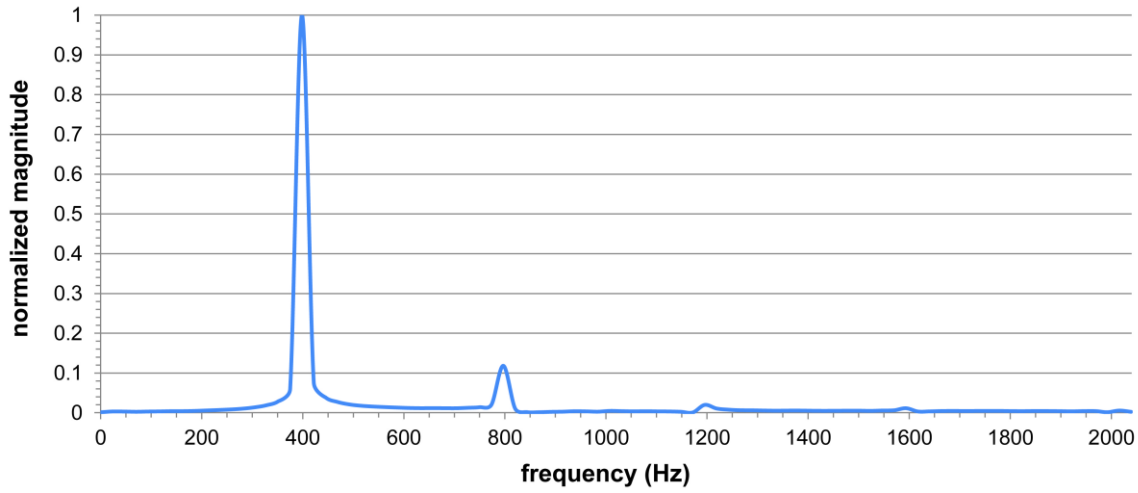


Figure 7-3: F_0 spectrum as measured at the microphone in the vocal tract

7.3. 3D Simulation Discussion

Table 7-1 shows the magnitude of the inputs and outputs from the simulation. Coleman et al. show that the minimum SPL at 15.25 cm for the fundamental frequency,

F_0, is 48 dB and the maximum controlled SPL observed was 126 dB. [92] Schindler et al. observed TE voicing at 30 cm as having a minimum of 50 ± 4.8 dB and a maximum of 68 ± 4.7 dB. [93]

Location	Primary Frequency (Hz)	SPL (dB)
Source 1	40000	105
Source 2	39600	105
Microphone	400	61.47
Alt Microphone	400	53.69

Table 7-1: Input and output magnitudes (*re: 20 μPa*) for locations in Figure 7-1

In addition to the vocal fundamental frequency, Formby and Monsen point out that the harmonics of the F_0 are present in normal speech providing important peaks in the spectra. [12] Markel and Gray show that a first-order estimate of the harmonic roll off rate can be modeled by −12 dB/octave. [94] We also see harmonics of the F_0 with a median roll off rate of −10.9 dB/octave with a standard deviation of 5.88 dB over the first five harmonics. Weiss et al. showed that roll off with a typical electrolarynx averages about -14 dB/octave, but is even higher for the critical lower harmonics. [95] The presence of these harmonics at higher pressures indicates the potential for a more natural sounding artificial voice.

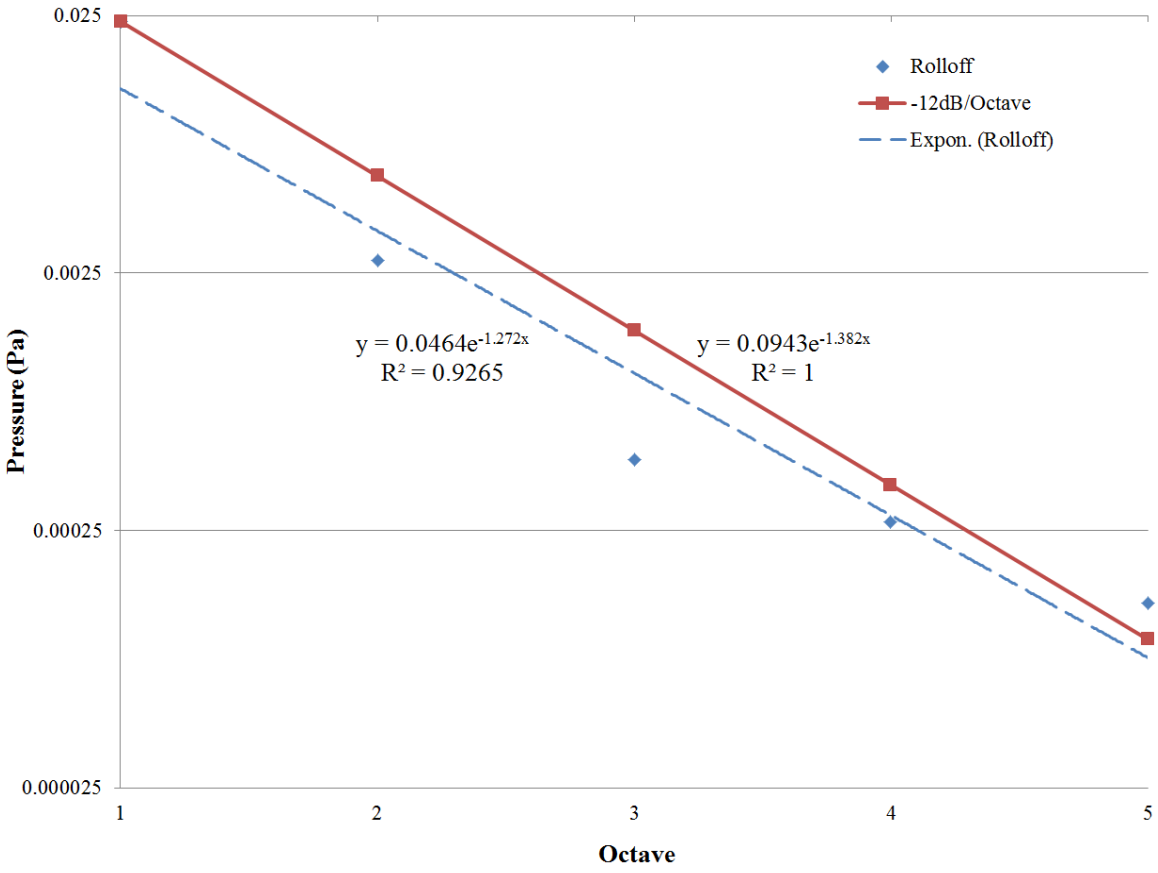


Figure 7-4: Simulation rolloff vs. typical rolloff model

Two of the major complaints with classic piston-driven electrolarynxes are the difficulty and range of frequency control and the self-noise generated by the devices. Given the method used to produce the F_0 in our simulation, we anticipate that nearly linear control over the normal range of fundamental frequencies of men, women, and children should be possible. Even with low-grade ultrasonic transducers a range of 1 kHz should be possible, which exceeds any documented vocal fundamental frequency we have seen.

To examine the self-noise, we moved the microphone 50 mm below the transducers (the "Alternate Microphone" location in Figure 7-1) and noted that the recorded waveform was 7.78 dB below the F_0. For reference, we recorded the self-noise of a Servox® Inton as 82 ± 3.7 dBA at 50 mm. Furthermore, as the primary waveforms

are above human hearing, most communications devices will filter such noise as part of their standard digitization process. We believe our device's lower acoustic leakage will offer a major advantage over electrolarynxes in use today.

7.4. Error Analysis

An important part of any experiment or simulation is gaining as complete an understanding of the sources of uncertainty and bounds for the error. Our simulation is fairly complex, and it is difficult to know exactly how the simulation results will compare with a real world device. However, assuming our material values are correct, we can get an upper bound on error introduced in the simulation itself.

As discussed in Section 6.3, we verified that our spatial step δ is over 8 times smaller than our smallest wavelength, and we calculated a time step of 122.5 ns that is less than the maximum calculated from the Courant number. These two checks should ensure that we have a stable convergent simulation.

Using equation 4.6, we can solve for the phase error given the spatial step, highest frequency of interest, and a distance or amount of simulated time. Our worst case comes from our ultrasonic sources, so we take the frequency f = 40 kHz. The wavelength of our source depends on the material's speed of sound; in order to estimate for the entire model, we take a value for generalized soft tissue of v = 1540 m/s. The number of points per wavelength becomes $N_{PPW} = v / (f \cdot \delta) = 38.5$. If we assume soft tissue has an attenuation of 0.5 dB/cm/MHz, then after a distance of D = 192 cm, the source signal will be reduced by 24 dB (94%). We can calculate the amount of time required for the signal to propagate over a distance as $D / v \approx 1.25$ ms. The number of cycles can now be calculated as $P = t \cdot f \approx 50$. Finally, we can solve for e_φ = 1 milliradian.

In Equation 4.11, we derived an alternate estimate of the phase error. Using the same soft tissue assumptions, frequency and distance as above, and noting that we can calculate the wavenumber k = 163.2, we can calculate |Error| = 1.2 milliradians. So both our error estimations, calculated via different methods, agree to a high degree.

A phase error of 1 milliradian means our output pressures will be multiplied by a factor of ± 0.001. A worst case where the errors on all our signals line up means an error of only about ± 0.01 dB in our simulation output.

To ground our theoretical error estimations, we ran another simulation with the same configuration as in Section 7.2 except we removed the AustinMan voxels leaving only air in the entire domain and disabled Source 2 leaving only the 40 kHz output. We ran the simulation for over 240K iterations and then ran a 2048 sample FFT on a subset of the output. We then generated an analytical model of a 40 kHz sine wave and ran the same type of FFT. Using the cross-correlation, we calculated R = 0.998 indicating that the two outputs are very similar. The major difference is that our analytical model assumes no dispersion or attenuation and the simulation has a dispersion error and includes attenuation even though it is minimal in air (ultrasonic frequencies are attenuated due to CO_2). Overall, the result of this comparison gives us great confidence in the simulator.

7.5. Tissue Property Comparison

As discussed in Section 6.1, many researchers simulate using the general *soft tissue* values only. In order to see what difference this makes to the simulation, we ran a comparison against one of our full material type runs. Starting with all parameters identical to the previous run, we modified each material in the full material file and set

density to 1000 kg/m^3, speed of sound to 1540 m/s, and attenuation to 0.5 dB/cm/MHz. One of the benefits using the soft tissue values, is that the maximum time step can be increased. From equation 4.7, we can calculate a new Δt as 375 ns, so our sampling frequency is 2.667 MHz. As before, we round up to a multiple of 96 kHz, and select a final sampling frequency of 2.688 MHz. Unfortunately, this time step is not stable because the small bones in the nasal passage surrounded by air caused instabilities. Ultimately, we used a sampling frequency of 3.84 MHz which gives Δt equal to 260 ns; this still represents more than a 2x speed up in the simulation.

After running for over 50 ms of simulated time, we exported the waveform and analyzed the demodulated signal. Our output at the microphone is 54.85 dB SPL at 400 Hz which is almost 6.5 dB lower than the full tissue simulation. Also of note, the median roll off rate is −3.6 dB/octave with a standard deviation of 13.9 dB over the first five harmonics; not nearly as clean a fit as before.

To hypothesize why, we looked at the primary signal path which includes in order: skin, fat, muscle, blood vessel, cartilage, trachea, and finally internal air (which is unchanged). Of these, fat becomes denser and faster; all other tissues become less dense and slower. It appears that the fat layer just beneath the skin is having the most impact on the results. It would be interesting to extend the simulator to allow samples to be taken at each tissue boundary and quantitatively confirm this hypothesis.

8. Anisotropic Viscoelastic Simulation

One of our hypotheses is that the acoustic impedance mismatch at the boundary between the trachea epithelial lining and the vocal tract, as shown in Figure 2-2 and Figure 2-3, is not a concern. The mismatch will result in 99.9% reflection of ultrasound energy; however, it is this reflection that results in a significant transfer of energy across the boundary. As the wave is reflected, it exerts pressure on the cell walls causing them to vibrate; essentially we have a mechanical coupling which causes the epithelial lining to act as a speaker oscillating at the beat frequency and thus transferring the fundamental frequency, F_o, into the vocal tract and enabling speech.

This coupling depends on nonlinear effects that have thus far been ignored in our linearized simulation. In this section, we develop an anisotropic viscoelastic model that incorporates these effects and simulates the mechanical coupling. We would have preferred to create a simpler constitutive model such as the Mooney-Rivlin but lacked a characterization to completely describe the tissue; a neo-Hookean model could be implemented by utilizing a subset of the tissue characterization available, however the complexity to add anisotropic behavior was negligible compared to the viscoelastic behavior. Also, the time required to run each simulation to compare different models is always a factor. In the end, we chose the model we did as the only tracheal tissue characterization we could find had data specifically for this model and it therefore minimized any assumptions on our part.

In the rest of this chapter, we will create a framework for calculating the nonlinear deformation of a finite volume. Next we will use a strain-energy density function to characterize the tracheal muscle tissue and link the deformation model to an applied

stress. Using the radiation force generated by the reflecting wave against the boundary wall, we link the FDTD model to a finite element model (FEM) where the applied stress can be used to calculate the deformation. Finally, we tightly couple the two models by utilizing the fact that tissue is almost completely incompressible while the air within the vocal tract is not; we therefore calculate the density change within the internal air computational cells created by the deformed muscle at each time step and use that new value in the FDTD model. This series of steps links the incoming pressure waves to an expansion/contraction of the boundary to a pressure change within the vocal tract which should accurately represent the viscoelastic mechanical coupling.

For further information, see Sharipov, Kelly, Banks et al., and Zienkiewicz and Taylor for a review of tensor calculus, solid mechanics, viscoelasticity, and FEM. [96] [97] [98] [99]

8.1. Isothermal nonlinear mechanical boundary coupling using a continuum model

At the tracheal wall, we assume the computational cells of our model make up independent continuum volumes. In an isothermal purely mechanical model, equilibrium is given by the principle of virtual work: [100]

$$\int_{V_o} S{:}\,\delta E - \left(B - \rho_o \ddot{U}\right) \cdot \delta U \, dV - \int_{S_o} \bar{T}_{(N)} \cdot \delta U \, dS = 0 \qquad 8.1$$

in the Lagrangian or material description where:

S is the second Piola-Kirchhoff (PK2) stress tensor,
E is the Green-Lagrange strain tensor,
B is the reference body force,
$\rho_o\ddot{U}$ is the inertial force per unit reference volume,
$\bar{T}_{(N)}$ is the first Piola-Kirchoff traction vector per unit reference surface area with normal N and the bar indicates this is a function of the deformation,

δU is the first variation of the displacement field U,
A:**B** represents double contraction of tensor **A** with tensor **B**,
V_o is the reference body volume, and
S_o is the boundary of the reference body.

Solving equation 8.1 for the displacement field at each time step will allow us to calculate the body deformation. We begin with some further definitions. For further information, see Banks et al. [101]

X is the position of a point P in the reference configuration,
x is the position of a point P in the current configuration

$$U = x - X \tag{8.2}$$

is the displacement vector.

$$\boldsymbol{F} = \frac{dx}{dX} \tag{8.3}$$

is the configuration or deformation gradient tensor,

$$\boldsymbol{C} = F^T F \tag{8.4}$$

is the symmetric positive definite right Cauchy-Green deformation tensor, and

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}) \tag{8.5}$$

is the Green-Lagrange strain tensor, where **I** is the second-order identity tensor.

Using these definitions, we can define the PK2 stress as: [100]

$$\boldsymbol{S} = J\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-T} \tag{8.6}$$

where J is the determinant of **F**, and **σ** is the Cauchy stress tensor in the spatial configuration.

We can also express the PK2 stress in terms of the gradient of a scalar-valued tensor function **ψ** where **ψ** is a Helmholtz free energy function: [100]

$$\boldsymbol{S} = 2\frac{\partial \boldsymbol{\Psi}(\boldsymbol{C})}{\partial \boldsymbol{C}} \tag{8.7}$$

When $\boldsymbol{\psi}$ is a function of a stress tensor only, it is referred to as the strain-energy density function (SDEF) and is defined per unit reference volume.

Trabelis et al. describe just such SEDFs for tracheal cartilage and smooth muscle. [102] In this analysis, we are only interested in the tracheal smooth muscle. Their histology of the muscle shows two orthogonal fiber groups which they characterize using a Holzapfel SEDF. The Holzapfel SDEF uses a decoupled representation which separates the volumetric and isochoric contributions found in many biological tissues. [103]

$$\boldsymbol{\Psi}(\boldsymbol{C}, \boldsymbol{A_0}, \boldsymbol{B_0}) = \boldsymbol{\Psi}_{vol}(J) + \boldsymbol{\Psi}_{iso}(\bar{\boldsymbol{C}}, \boldsymbol{A_0}, \boldsymbol{B_0}) \qquad 8.8$$

where $\boldsymbol{\psi}_{vol}$ represents the volumetric contribution which is assumed to be elastic, and $\boldsymbol{\psi}_{iso}$ represents the isochoric contribution and utilizes the modified right Cauchy-Green tensor given by $\mathbf{C}=J^{2/3}\ \bar{\mathbf{C}}$ which isolates the dilatational and distortional deformations. $\mathbf{A}_0$ and $\mathbf{B}_0$ are structural tensors based on the unit vector fiber directions: $a_0 \otimes a_0$ and $b_0 \otimes b_0$. Since a_0 and b_0 are orthogonal, the tissue is orthotropic in the reference configuration.

The characterization by Trabelis et al. is missing the volumetric contribution, but a later paper by Malvè et al. from the same research group includes all components: [104]

$$\begin{aligned} \boldsymbol{\Psi}_{vol} &= \frac{1}{D}(J-1)^2 \\ \boldsymbol{\Psi}_{iso} &= C_{10}(\bar{I}_1 - 3) + \frac{k_1}{2k_2}\left(e^{k_2(\bar{I}_{4a}-1)^2} - 1\right) + \frac{k_3}{2k_4}\left(e^{k_4(\bar{I}_{4b}-1)^2} - 1\right) \end{aligned} \qquad 8.9$$

where the first expression in $\boldsymbol{\psi}_{iso}$ represents the standard neo-Hookean model incorporating the isotropic behavior and the remaining expressions capture the

anisotropic fiber behaviors. The values of the constants are shown in Table 8-1. The modified invariants are given by: [103]

$$\bar{I}_1 = tr\,\bar{\boldsymbol{C}} \qquad \bar{I}_{4a} = \bar{\boldsymbol{C}}:\boldsymbol{A}_0 \quad \bar{I}_{4b} = \bar{\boldsymbol{C}}:\boldsymbol{B}_0 \tag{8.10}$$

where $\bar{I}_1$ is the first modified invariant and $\bar{I}_{4a}$ and $\bar{I}_{4b}$ are the fourth modified invariants with respect to $\mathbf{A}_0$ and $\mathbf{B}_0$. Since $\bar{\mathbf{C}}$, $\mathbf{A}_0$ and $\mathbf{B}_0$ are all symmetric, the double contractions in the fourth modified invariants can be performed in either direction, and this is often seen in the literature.

$\mathbf{C_{10}}$ (kPa)	$\mathbf{k_1}$ (kPa)	$\mathbf{k_2}$	$\mathbf{k_3}$ (kPa)	$\mathbf{k_4}$	$\mathbf{D}$ (kPa)
0.877	0.154	34.157	0.347	13.889	8.108

Table 8-1: SDEF constants [102] [104]

Using equations 8.8 and 8.9, we can now derive an equation for the PK2 stress as defined in equation 8.7. For an in-depth derivation, see Gasser and Holzapfel. [105] First, we note that since the Holzapfel SDEF uses a decoupled representation, the resulting stress equation will also have a decoupled representation.

$$\boldsymbol{S} = 2\frac{\partial \boldsymbol{\Psi}(\boldsymbol{C})}{\partial \boldsymbol{C}} = \mathbf{S}_{vol} + \mathbf{S}_{iso} \tag{8.11}$$

Starting with the volumetric part, we have:

$$\mathbf{S}_{vol} = 2\frac{\partial \boldsymbol{\Psi}_{vol}(J)}{\partial \boldsymbol{C}} = Jp\boldsymbol{C}^{-1} = J\left(\frac{\partial \boldsymbol{\Psi}_{vol}(J)}{\partial J}\right)\boldsymbol{C}^{-1} = \frac{2}{D}J(J-1)^2\boldsymbol{C}^{-1} \tag{8.12}$$

where p is the hydrostatic pressure. From equation 8.9, we see that p = 2(J-1)/D. We continue with the isochoric part:

$$\mathbf{S}_{iso} = 2\frac{\partial \boldsymbol{\Psi}(\bar{\boldsymbol{C}})}{\partial \boldsymbol{C}} = J^{-2/3}\left(\mathbb{I} - \frac{1}{3}\boldsymbol{C}^{-1}\otimes\boldsymbol{C}\right):\bar{\boldsymbol{S}} = J^{-2/3}Dev\,\bar{\boldsymbol{S}} = J^{-2/3}\mathbb{P}:\bar{\boldsymbol{S}} \tag{8.13}$$

where $\mathbb{I}$ is the fourth-order identity tensor, $\mathbb{P}$ is the fourth-order projection tensor, and *Dev* is the deviatoric operator defined as: *Dev* (●) = (●) − ⅓ [● : **C**] $\mathbf{C}^{-1}$. [105] Now we can define the modified PK2 stress, $\overline{\mathbf{S}}$, as:

$$\overline{\boldsymbol{S}} = 2\frac{\partial \boldsymbol{\Psi}_{iso}(\overline{\boldsymbol{C}})}{\partial \overline{\boldsymbol{C}}} = 2\frac{\partial \boldsymbol{\Psi}_{iso}(\overline{I}_1, \overline{I}_{4a}, \overline{I}_{4b})}{\partial \overline{\boldsymbol{C}}} = 2\sum_x \frac{\partial \boldsymbol{\Psi}_{iso}}{\partial I_x}\frac{\partial \overline{I}_x}{\partial \overline{\boldsymbol{C}}} \qquad 8.14$$

where x ∈ { 1, 4a, 4b }, $\partial\boldsymbol{\Psi}_{iso}/\partial I_x$ is a scalar, and $\partial\overline{I}_x/\partial\overline{\mathbf{C}}$ is a second-order tensor. Using the right-hand side of equation 8.14 and the modified invariants from equation 8.10, we can derive each modified PK2 stress component independently:

$$\begin{aligned}
\overline{\boldsymbol{S}}_1 &= 2\frac{\partial \boldsymbol{\Psi}_{iso}}{\partial I_1}\frac{\partial \overline{I}_1}{\partial \overline{\boldsymbol{C}}} = 2\frac{\partial \boldsymbol{\Psi}_{iso}}{\partial I_1}\boldsymbol{I} \\
\overline{\boldsymbol{S}}_{4a} &= 2\frac{\partial \boldsymbol{\Psi}_{iso}}{\partial I_{4a}}\frac{\partial \overline{I}_{4a}}{\partial \overline{\boldsymbol{C}}} = 2\frac{\partial \boldsymbol{\Psi}_{iso}}{\partial I_{4a}}a_0 \otimes a_0 \\
\overline{\boldsymbol{S}}_{4b} &= 2\frac{\partial \boldsymbol{\Psi}_{iso}}{\partial I_{4b}}\frac{\partial \overline{I}_{4b}}{\partial \overline{\boldsymbol{C}}} = 2\frac{\partial \boldsymbol{\Psi}_{iso}}{\partial I_{4b}}b_0 \otimes b_0
\end{aligned} \qquad 8.15$$

Substituting these back into equation 8.13, we can derive each isochoric PK2 stress component:

$$\begin{aligned}
\boldsymbol{S}_{iso_1} &= 2J^{-2/3}\frac{\partial \boldsymbol{\Psi}_{iso}}{\partial \overline{I}_1} Dev\left(\frac{\partial \overline{I}_1}{\partial \overline{\boldsymbol{C}}}\right) = 2J^{-2/3}C_{10}\left(\boldsymbol{I} - \frac{\overline{I}_1}{3}\overline{\boldsymbol{C}}^{-1}\right) \\
\boldsymbol{S}_{iso_{4a}} &= 2J^{-2/3}\frac{\partial \boldsymbol{\Psi}_{iso}}{\partial \overline{I}_{4a}} Dev(\boldsymbol{A}_0) = 2J^{-2/3}k_1\left(\overline{I}_{4a} - 1\right)e^{k_2\left(\overline{I}_{4a}-1\right)^2}\left(\boldsymbol{A}_0 - \frac{\overline{I}_{4a}}{3}\overline{\boldsymbol{C}}^{-1}\right) \\
\boldsymbol{S}_{iso_{4b}} &= 2J^{-2/3}\frac{\partial \boldsymbol{\Psi}_{iso}}{\partial \overline{I}_{4b}} Dev(\boldsymbol{B}_0) = 2J^{-2/3}k_3\left(\overline{I}_{4b} - 1\right)e^{k_4\left(\overline{I}_{4b}-1\right)^2}\left(\boldsymbol{B}_0 - \frac{\overline{I}_{4b}}{3}\overline{\boldsymbol{C}}^{-1}\right)
\end{aligned} \qquad 8.16$$

And finally summing the isochoric components leads to:

$$\begin{aligned}
\mathbf{S}_{iso} = 2J^{-2/3}\Bigg[&C_{10}\left(\boldsymbol{I} - \frac{\overline{I}_1}{3}\overline{\boldsymbol{C}}^{-1}\right) + k_1\left(\overline{I}_{4a} - 1\right)e^{k_2\left(\overline{I}_{4a}-1\right)^2}\left(\boldsymbol{A}_0 - \frac{\overline{I}_{4a}}{3}\overline{\boldsymbol{C}}^{-1}\right) \\
&+ k_3\left(\overline{I}_{4b} - 1\right)e^{k_4\left(\overline{I}_{4b}-1\right)^2}\left(\boldsymbol{B}_0 - \frac{\overline{I}_{4b}}{3}\overline{\boldsymbol{C}}^{-1}\right)\Bigg]
\end{aligned} \qquad 8.17$$

Equations 8.12 and 8.17 provide a complete characterization of the PK2 stress for the tracheal muscle.

8.2. Finite Element Method

FEM is a popular and mature way to model complex geometries and numerically simulate mathematical models. Like FDTD, FEM discretizes the domain into a mesh; each of these subdomains is called an element. Each element has associated with it a set of equations that describe its state and evolution. After each time step, the elements can be recombined to provide a representation of the whole domain.

Holzapfel and Gasser utilized the Finite Element Analysis Program (FEAP) by the University of California at Berkley to demonstrate their anisotropic viscoelastic algorithm. [103] [105] Malvè et al. used ABAQUS® to demonstrate theirs. [104] Both these packages are commercial products and too large and complicated to be integrated with our simulator. We want a solution that can be integrated directly in CUDA so we do not suffer the performance penalties of transferring data from device to host and back during each time step.

While there are many excellent open source FEM packages, there are only a few that offer a nonlinear anisotropic viscoelastic solver implemented in CUDA. Taylor et al. developed just such an implementation. [106] Their group is working to create real-time haptic surgical simulations that include accurate tissue deformation. Their work extends the total Lagrangian explicit dynamic (TLED) finite element algorithm presented by Miller et al. and uses a basic neo-Hookean SDEP. [107] Later work by Taylor et al. adds in viscoelastic effects; this is the solution we ultimately adopted. [108] Comas et al. integrated the algorithm into the Simulation Open Framework Architecture (SOFA) open

source project which is designed to showcase real-time solutions with a focus on medical simulation. [109]

We started by retrieving the SOFA package and compiling a 64-bit application with Visual Studio 2010. The download is quite manageable at just over 3 MB; however, once built, the total size of all files is over 12 GB. This is a large framework, and while there is an excellent website with a detailed online manual, wiki, and many user guides and tutorials, there is no online search feature which makes locating specific information within the project difficult. Using Google, we were able to locate the TLED implementation and get one of the demos running. Unfortunately, a number of complications occurred. First, many of the included examples are missing required files; it appears that as the framework has evolved, some files have been removed that are required by the TLED demos (such as the liver object file). But as we were able to locate a basic cube demo that would execute, we were able to evaluate the hexahedron based solution that interested us most. We were able to get the viscoelastic feature to run immediately, but every time we attempted to enable the anisotropic feature, the model would simply disintegrate. After reviewing the code, we discovered three bugs; one of which prevented the anisotropic feature for working. First, we noticed two memory leaks during setup; these are only used once during initialization and the memory lost is minor so they were not a major issue, but should still be corrected. For the other, we found that at some point during refactoring, someone incorrectly optimized the transfer of anisotropic properties from host to device. The GPU expected a four tuple of single precision floating point values for each anisotropic preferred direction as it stored these directly into a texture which must be either a one, two or four tuple; the host side sent

only three and therefore the host properties were misaligned and read back as garbage. Once we corrected this bug, the anisotropic code functioned as expected.

In the SOFA model, Comas et al. only used a single anisotropic fiber; however, extending this to two fiber directions is simple, and since the characterization from Trabelsi et al. involves two orthogonal fibers, the calculation may be speeded up over a more general solution should speed be more important than generality.

The TLED algorithm solves equation 8.1 at each time step by using the central difference method as described in equation 4.1 to calculate the next displacement based on displacements at time t and t-Δt: [106]

$$U_{t+\Delta t} = A(R_t - F_t) + BU_t + CU_{t-\Delta t} \tag{8.18}$$

where R are externally applied loads, F are nodal forces, and A, B, and C are precomputed constants given by: [106]

$$A = \frac{1}{\frac{D}{2\Delta t} + \frac{M}{\Delta t^2}}$$
$$B = \frac{2M}{\Delta t^2} A \tag{8.19}$$
$$C = \frac{D}{2\Delta t} A - \frac{B}{2}$$

where D is the damping matrix and M is a lumped mass approximation. The damping matrix is an artificial construct present to ensure model stability; however, it is not required when viscoelastic effects are included as energy dissipation is then included at a constitutive level.

Like in the FDTD method, the central differences method used for explicit time integration has a maximum time step Δt to ensure stability.

$$\Delta t_{cr} = \frac{L_c}{c} \tag{8.20}$$

where L_c is the smallest element length which for an eight-node hexahedral element is V/A_{max} where V is the element volume and A_{max} is the area of the element's largest face. [107] And c is the dilatational wave speed of the material given by:

$$c = \sqrt{\frac{E(1-\upsilon)}{\rho(1+\upsilon)(1-2\upsilon)}} = \sqrt{\frac{\lambda+2\mu}{\rho}} = \sqrt{\frac{2\mu(1-\upsilon)}{\rho(1-2\upsilon)}} \tag{8.21}$$

where E is Young's modulus, υ is Poisson's ratio, λ is Lamé's first parameter, μ is the shear modulus and ρ is the element density. Strictly speaking, equation 8.21 holds only for homogeneous isotropic linear elastic materials; however, we can use it to estimate our maximum time step by noting that in the neo-Hookean model, the shear modulus equals twice parameter C_{10} from Table 8-1 and the bulk modulus is twice parameter D. From the shear, μ, and bulk moduli, κ, we can calculate the Poisson ratio as $\nu = (3\kappa - 2\mu) / [2(3\kappa + \mu)] = 0.45$. However, as the dilatational wave speed decreases as ν increases towards 0.5 (perfectly incompressible), we use 0.49 which is a typical tissue value representing a nearly incompressible material to calculate the minimum t_{cr}. Using these E and ν along with a volume of 1 mm^3 and area of 1 mm^2 in the reference configuration and the density of tracheal tissue from Table 6-1, we can estimate the maximum time step as 110 μs. This value is much larger than the step size used for the FDTD simulation which will drive both calculations in a tightly coupled solution.

At each time step, the TLED algorithm follows the following steps: [108]

1. Apply loads and boundary conditions.
2. Calculate the deformation gradient using equation 8.3.
3. Calculate the PK2 stress.

4. Calculate the linear strain-displacement matrix.
5. Calculate the element nodal forces.
6. Calculate the displacement using equation 8.18.

Step one will be discussed in the next subsection as the loads are generated by our FDTD simulation. Viscoelastic effects are applied to the PK2 stress in step three; these effects present as time-dependent strain which results in uneven loading and unloading, creep and relaxation, and stiffness affected by rate of loading. Taylor et al. model this as a Prony series which they linearize to create an update equation for each time step: [108]

$$\boldsymbol{\Upsilon}_n = \left(\frac{\tau + \Delta t}{\tau \Delta t}\right)\left(\frac{\alpha \overline{\boldsymbol{S}}}{\tau} + \frac{\boldsymbol{\Upsilon}_{n-1}}{\Delta t}\right) \qquad 8.22$$

where α and τ are constants representing the tissue. As we were unable to find specific values for the trachea, we used the liver values provided by Taylor et al. of α = 0.5 and τ = 0.58. We compute the final PK2 stress using equations 8.12, 8.17, and 8.22:

$$\boldsymbol{S} = \mathbf{S}_{vol} + \mathbf{S}_{iso} - \boldsymbol{\Upsilon}_n \qquad 8.23$$

Steps four and five provide the speed of the TLED algorithm; rather than computing the stiffness matrix at each time step, TLED uses the fact that the nodal forces in step five can be calculated using the strain-displacement matrix from step four which can be calculated based on precomputed element shape spatial derivatives and the deformation gradient from equation 8.3: [106]

$$B = B_0 \boldsymbol{F} \qquad 8.24$$

where B is the strain-displacement matrix which depends on the reference configuration strain-displacement matrix B_0 which can be precomputed using the element

shape spatial derivatives. [110] The element nodal forces in an eight-node hexahedral element can then be calculated as: [108]

$$\tilde{f} = 8B\check{\mathbf{S}}detJ \qquad 8.25$$

where $\check{\mathbf{S}}$ is the vector form of the stress tensor **S**, and J is the precomputed element Jacobian matrix.

8.3. Defining the FEM Mesh

Before we begin to mesh the AustinMan voxels for the finite element model, it is important to define how each element will be laid out and to label faces and nodes. Figure 8-1 shows the faces labeled in large bold numerals; these start at the back and proceed counter-clockwise and top to bottom. Similarly, nodes begin at the top back right and increase counter-clockwise top to bottom.

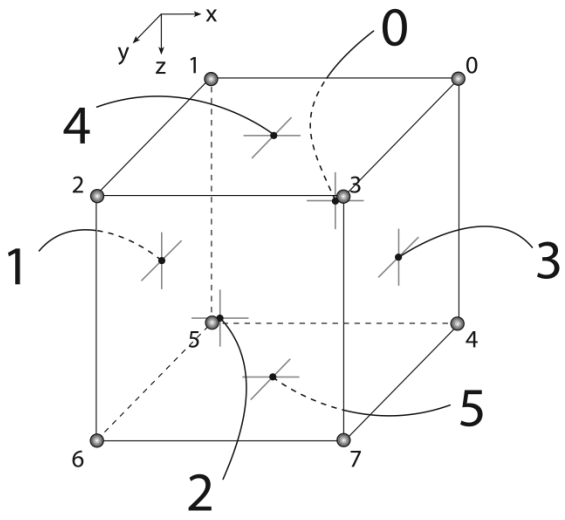


Figure 8-1: Element face and node layout

We assume all elements within our FEM mesh will have this orientation; nodal forces and anisotropic fiber directions will be rotated to correctly orient them to this reference orientation. Using the reference orientation requires less device memory and

allows better optimization of the TLED algorithm. To see how this reference orientation will work, imagine a two-dimensional transverse slice; each voxel is bordered by eight other voxels as shown in Figure 8-2. If we consider that any bordering voxel can be classified as either internal air or tissue, there are a total of 256 possible configurations. However, we are interested only in voxels that are made of tracheal tissue and that have at least one face bordering a voxel of internal air. Furthermore, given that this is a model of a real human being, all voxel tissues will be connected to the body; there will be no free-floating voxels surrounded on all sides by internal air.

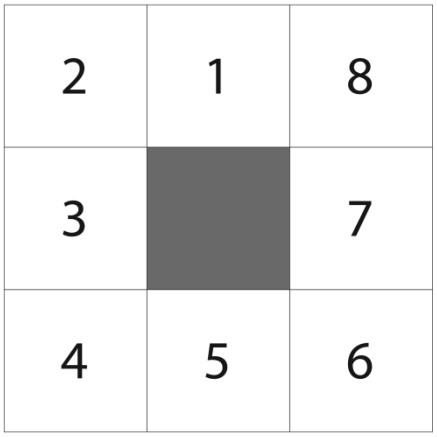


Figure 8-2: Neighboring voxels from a Transverse slice

Given these constraints, we can reduce the number of possible configurations to just 16. To calculate the state of a particular voxel, we use the face numbering from Figure 8-1. Starting with a state value of zero, we logical OR one left shifted by the face number for each of the perpendicular faces coinciding with neighbors 1, 3, 5, 7. This provides a set of active faces for calculating nodal forces. Furthermore, by grouping these 16 configurations by primary face, we can represent these configurations through four simple rules (*numbering intentional*):

0. Neighbor 5 must not be internal air; neighbor 1 must be internal air; Rule 3 must not be true. *(internal air neighbors: 1 or 1 and 3 or 1 and 3 and 7)*

1. Neighbor 7 must not be internal air; neighbor 3 must be internal air; Rule 0 must not be true. *(internal air neighbors: 3 or 3 and 5 or 1 and 3 and 5)*

2. Neighbor 1 must not be internal air; neighbor 5 must be internal air; Rule 1 must not be true. *(internal air neighbors: 5 or 5 and 7 or 3 and 5 and 7)*

3. Neighbor 3 must not be internal air; neighbor 7 must be internal air; Rule 2 must not be true. *(internal air neighbors: 7 or 1 and 7 or 1 and 5 and 7)*

Only one rule may be active at a time and an error is generated when none, all, or only opposing neighbors are active. The rule numbering allows easy indexing into arrays that identify the primary tissue voxel face as well as the opposing internal air voxel face as shown in Figure 8-1.

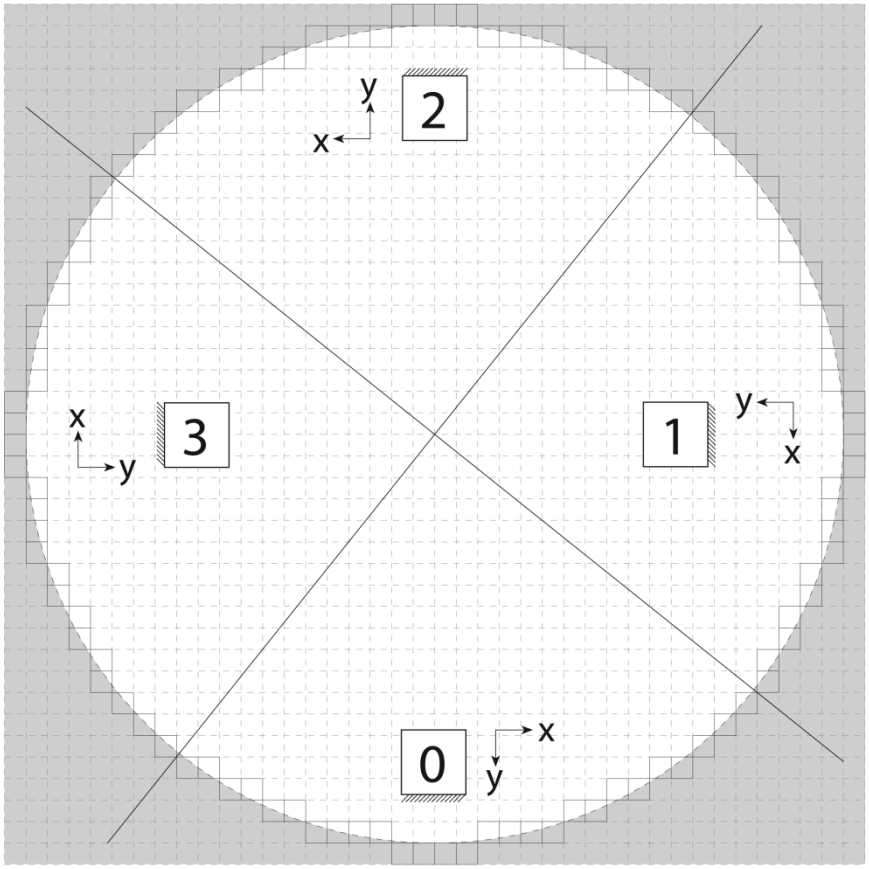


Figure 8-3: Example voxel orientation and FEM mesh

Figure 8-3 shows an example 2D mesh; the dashed circle marks a perfect circle which has been discretized using Bresenham's circle algorithm without aliasing as indicated by the square outlines which are considered completely tracheal tissue. The shaded region indicates tissue while the white interior area indicates internal air; dotted horizontal and vertical lines show the transverse boundaries of each voxel. The figure has been separated into segments indicating each rule, the large boxes show the voxel orientation with the fixed edge and the axis rotation required for nodal forces and anisotropic fiber orientation. This is a simplified example; in a real trachea, the boundary would not be circular and therefore the segments would not necessarily be diagonal nor equal.

In the AustinMan voxels, there are 3,162 voxels that match one of the four rules; these voxels make up the FEM mesh. There are two different ways to implement the mesh; one in which each element is independent of all others and one in which neighboring elements are connected and interact with each other. We have implemented the former as we view this as the worst case scenario representing the least amount of mechanical coupling; for large deformations, a connected mesh with hourglass control may be required for stability.

8.4. FDTD to FEM and Back Again

In Section 4.1, we outlined the procedure for calculating the FDTD; in Section 6.3, we explained that the calculation is split into two kernels – in the first, we calculate velocities and in the second pressures. In order to strongly couple the FDTD and FEM simulations, we will insert a new kernel between these two existing kernels. The TLED FEM kernel uses the updated velocities and current pressures to calculate the nodal forces

and then calculates each element's deformation using equation 8.1; we use the deformation to update the internal air densities and run the FDTD pressure kernel to calculate the updated pressures. The process then repeats for the next time step.

To calculate the nodal forces, we first calculate the acoustic radiation pressure: [111]

$$|p_{rad}| = \frac{\langle I \rangle}{c_0} = \frac{p^2}{2\rho_0 c_0^2} \qquad 8.26$$

where p_{rad} is the radiation pressure, ‹I› is the time average acoustic intensity, p is the incident acoustic pressure, ρ_0 is the material density, and c_0 is the speed of sound in the material. In the case of a high impedance boundary, nearly 100% of the wave energy will be reflected back and equation 8.26 is multiplied by 2R/100 from equation 3.33. We can calculate the force by multiplying by the incident surface area: [112]

$$|F| = \frac{Rp^2\delta^2}{100\rho_0 c_0^2} \qquad 8.27$$

where δ is the spatial step size. Finally, we can calculate the instantaneous radiation force incident on a particular element face using: [113]

$$F = \frac{Rpv\delta^2}{100c_0} \cdot \hat{n} \qquad 8.28$$

where p and v are the incident pressure and velocity respectively and $\hat{n}$ is the unit normal vector to the element face calculated using the node definitions from Figure 8-1. Equation 8.28 provides the final value necessary to calculate equation 8.1.

Since tracheal muscle is basically incompressible, we assume that faces not bordering internal air are fixed. And since air is compressible, any tissue displacement will result in a volume change within the neighboring internal air voxels. The deformed

tissue element is unlikely to be rectangular, therefore we use algorithm 14 from Grady to compute the deformed volume. [114] After computing the new volume for each element, we walk through the list of adjacent internal air elements and calculate their new volumes taking into account all connected tissue elements. Since the time step is very small, we assume that these volume changes cannot be equalized instantaneously and therefore update the internal air element's density. This final step completes the FEM kernel and the FDTD simulation resumes, calculating the updated pressure using the new densities.

8.5. Discussion

This part of our research was by far the most complicated and difficult. We had almost given up on a viscoelastic model until we found the trachea characterization by Trabelis et al. We then had to learn the basics of tensors, review nonlinear deformations, find open source implementations and integrate one into our code, test and run the simulations.

We ran the simulation using the same parameters as discussed in Section 7.2. See Figure 7-1 for the location of the sources and microphone. Table 8-2 shows a comparison between FDTD only simulations and those using the coupled FDTD and FEM simulation. The first row shows a comparison between the simulation presented in Section 7; the remaining rows show additional simulations run with inputs of 39900 Hz and 41000 Hz at three different locations within the vocal tract.

Beat Frequency (Hz)	Y Location (mm)	FDTD (dB)	FDTD+FEM (dB)
400	179	61.47	58.40
200	172	62.37	62.28
200	179	46.94	46.87
200	183	58.86	58.91

Table 8-2: Results for FDTD and FDTD+FEM simulations

The simulation results do not show a major difference between the fourth-order linear model and the anisotropic viscoelastic model. We were expecting to see higher outputs using the coupled simulation than the FDTD alone as the coupled simulation explicitly models the mechanical coupling that we can see and feel when observing the device or a classic electrolarynx on a simple distilled water phantom. We can feel the interior surface vibrate which strongly implies a mechanical coupling.

To understand why the coupled simulation did not show a major difference with the linearized simulation, we must consider the many assumptions made in creating the simulation. Our FEM model considers only the first layer of tracheal computation cells. Each computational cell is 1 mm^3 which could reasonably contain anywhere between 250K to 2M biological cells. [115] The AustinMan voxels only characterize the gross tissues; we assumed that the airway boundary is made of tracheal muscle only but in reality there are many tissue types including: mucosa, submucosa, and adventitia membrane, all of which are highly nonlinear and viscoelastic. [116] Furthermore, there is also a layer of liquid coating the airway as well as cilia. [117] Any of these layers or structures could be important in the mechanical coupling.

We also used a disconnected mesh as it presents a worst-case model; a connected mesh would have resulted in a larger deformation and perhaps that is a better model of reality. Going a step further and modelling several tissue layers in more detail and fully connecting them into a single mesh would require substantial computational resources but would most likely result in a more accurate model which may show the effects we were seeking.

On a positive note, it may be that the mechanical coupling is in fact captured by the fourth-order linear model and that the agreement between the two result sets is a validation of both models.

9. Designing a Physical Device

The goal of the early device is not to build the smallest, fastest or most efficient device, but rather to have a device we can easily monitor, repair, and extend. Defining our goals is important as it leads to very different choices in the design phase than if we were building a commercial device. The first step is to determine how we will generate our ultrasonic waves; once we have a transducer, we will know how much power and processing is required and what general size the board will be. The next step is to build test circuits and eventually a robust device that can be handled without worrying about loose connections. Once the design has been validated, we can perform some basic tests and finally see if we can actually generate sound from ultrasound through a phantom.

9.1. Transducer selection and resonant frequency identification

There are a number of low-frequency ultrasound (LFUS) transducers on the market for use in alarms, garage door openers, cleaning equipment, sensing and distance calculation. Many of the transducers are for receiving only, and many are in the upper end of the LFUS spectrum. At the low end, 20-30 kHz, sound may be audible to young children and household pets, so we ruled these out. Above 100 kHz would require faster electronics, which while possible, would add time and expense to creating the device; in addition, higher speed digital signals are more susceptible to analog noise than lower speed signals. Therefore we concentrated on the 30 – 100 kHz range. There are several possibilities, and a few examples are show in Figure 9-1. Most are of the type on the far left of the figure; these are designed specifically for use in air and have a mesh covering that would cause poor energy transfer into the body. The weather resistant models shown in the center and right side of the figure offer an excellent option. Ultimately, we

selected the Panasonic EFRTQB40K5 due to its shape, 1 kHz bandwidth, and weatherproof design. Vasilescu et al. used this same transducer in the MIT/CSIRO underwater sensor network and reported good acoustic coupling with water once the driving frequency was lowered to account for the greater density of water. [118] Once sealed and immersed in water, they could send modulated signals over 15 meters with a 20 V_{PP} input.

Figure 9-1: Sample ultrasonic transducers

The transducer has a nominal frequency of 40.0 ± 1.0 kHz, SPL of 105 dB at 30 cm, minimum bandwidth of 1.0 kHz, a maximum input of 20 V_{RMS} (≈ 56.6 V_{PP}), and cost about $20 each. Each unit is a PZT bimorph covered in an aluminum shell which is coated with an electrically insulating paint and joined to the base with latex foam. The only downside for our application is the units transmit almost uniformly over a hemisphere; we would prefer a more focused wave to reduce leakage.

To find the resonant frequency of each transducer, we connected it to a function generator and placed it in a sealed box whose interior was covered with wedge foam acoustic panels. A cardioid condenser microphone on a desk stand was also placed inside the box and located 6 inches from the front plane of the transducer. The microphone was connected to an Edirol UA-25 digitizer set to 24-bit samples at 96 kHz. Adobe Audition was used to record the output, and a VC3165 high resolution counter was used to verify the frequency readings.

The function generator was set to 5 V_{RMS}. The frequency was then swept from 30 – 50 kHz. Each test was run multiple times using both Sine and Square wave outputs. As the input frequency approaches the resonant frequency, the SPL rises and then drops off as the frequency goes past. Table 9-1 shows the results for all transducers.

Transducer	Avg. Resonant Frequency (Hz)	Relative Output (dB)
1	38672.5 ± 0	-1.0
2	38720.0 ± 0.5	-1.5
3	38594.0 ± 247.7	-4.0
4	38625.0 ± 0	-1.0
5	39844.0 ± 0	-5.0

Table 9-1: Transducer Resonant Frequency

The relative output is based on setting the Edirol's output to a constant level and then comparing the strength of each transducer's output. It helps pair transducers with similar output amplitudes. We also noticed that the highest output occurs when driving with a square wave; the square wave does not appear to have any negative impact on the output waveform.

9.2. Circuit design

Using a breadboard, we started with a single transducer and a MOSFET. We input a square wave from the signal generator into the MOSFET and monitored the acoustic output. But driving only one side of the transducer would not allow maximum power; to avoid having multiple power supplies, we opted for an H-bridge configuration which would allow us to switch a single power source between each terminal creating twice the voltage peak-to-peak.

An easy configuration for an H-bridge driven by a microprocessor is to use P channel MOSFETS for the high side as N channel MOSFETS require about 12V above the source to turn off once in the "on" state. Unfortunately, P channel MOSFETS have a

higher “on” resistance which means they waste more power as heat. A better solution is to use N channel MOSFETS and a driver circuit to provide the extra voltage necessary to turn them off. As H-bridges are a common configuration for bi-directional motor control in robotics, ICs exist that contain all the necessary circuitry to provide the extra voltage (charge pump) as well as protection to ensure that both channels are not open at the same time during switching (shoot through). All that is required are some appropriately sized capacitors and diodes for the charge pump circuitry and a resistor for the shoot through delay. To protect each MOSFET, we added a low valued resistor and a diode. This configuration allowed us to use a single 12 V power source and generate 24 V_{PP} across the transducer. Adding an inverter allowed control from a single clock input.

With the analog section working, we began designing the digital control section. We wanted a flexible system that would allow us to test various scenarios. As such we added communications systems for both USB and serial. We also wanted a way to observe and record the acoustic output, so we added a ceramic microphone (linear up to 22 kHz but sensitive all the way to about 60 kHz) and a two stage amplifier that allows the microprocessor to read the signal. As the microprocessor has limited memory and flash storage, we added a SD card port which can be used for recording as well as reading complex input signals should we need that functionality.

For the microprocessor, we opted for an Atmel ATmega328P which has 32 KB of flash, 1 KB of EEPROM, and 2 KB of RAM. It also has a high-resolution 16-bit timer, PWM, A/D converter, and can run at up to 20 MHz.

To isolate the digital and analog sections, we use a separate power supply for the high-voltage analog section (up to 30 V) and for the digital section (up to 20 V). The

high-voltage section is fed directly to the MOSFETs and to a 12 V voltage regulator to supply the H-bridge drivers. The digital voltage can come from an external input or from the USB port and is selectable with a jumper.

The circuit schematic was created with Cadence Capture and is shown in Figure 9-2.

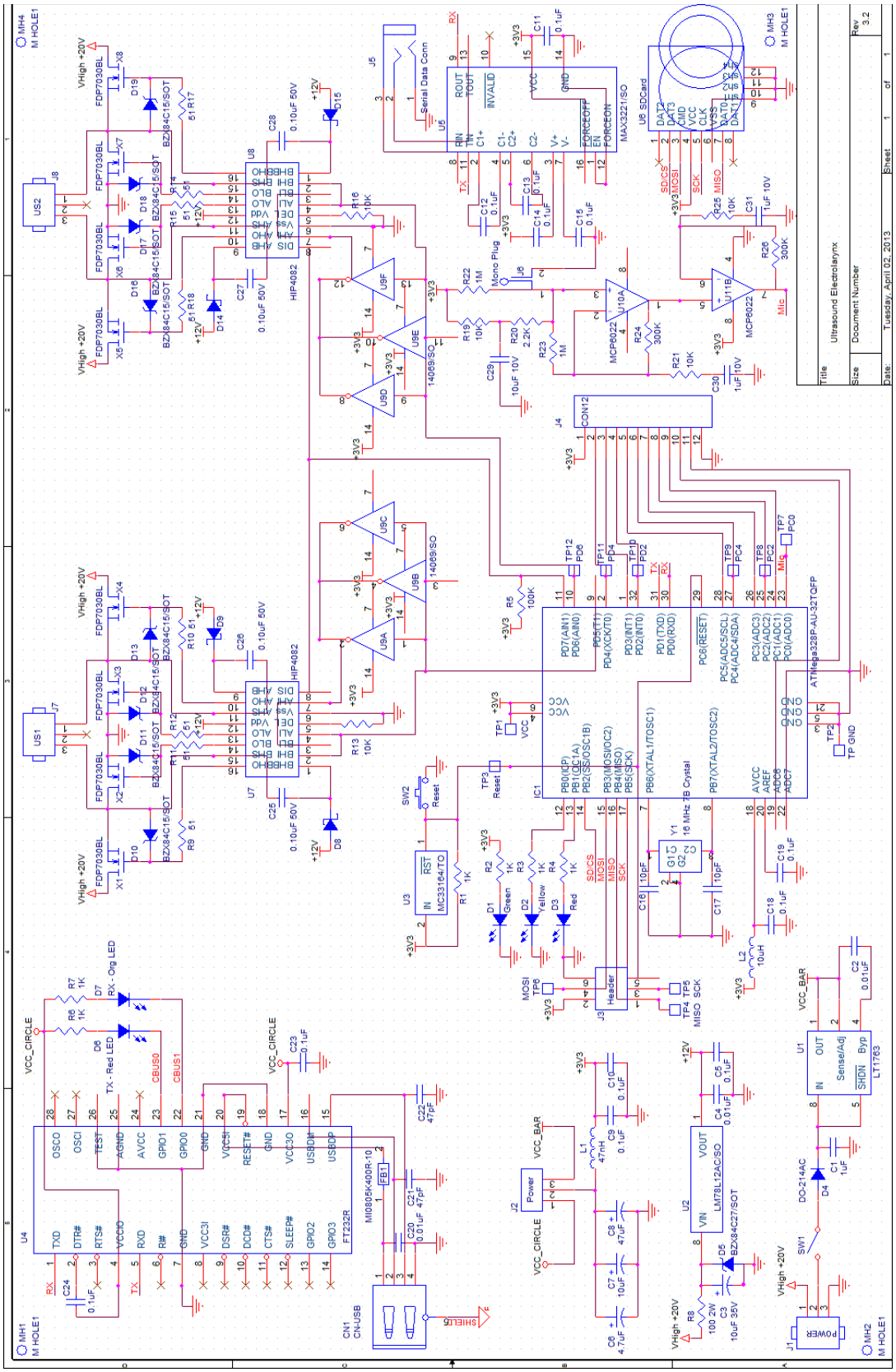


Figure 9-2: Device Schematic

9.3. PCB fabrication and assembly

Once we had the schematic laid out, we needed to identify and source appropriate components. This step takes considerable time as there are many options each with its own tradeoffs in terms of functionality, availability and price. For the MOSFETs we went with the Fairchild 7030L due to their high voltage and current capability, low $R_{DS(ON)}$ and high switching rate. To control the gates, we selected the Intersil HIP4082 as it worked in the voltage ranges we were targeting and offered the switching speeds we required. We chose the FTDI FT232R for the USB chip as it has a solid host side driver and integrates directly with our microprocessor using an internal serial converter; it also provides a 3.3V regulated output generated from the USB port. Where possible we selected surface mount components, opting for through-hole only for some of the higher current analog components. Table 9-2 shows the Bill of Materials.

Using Cadence Allegro, we created packages for each of the parts which did not have standard surface mount profiles. This is a long and laborious process which involves finding and interpreting each part's specification sheet and carefully creating the proper footprints. Once completed, we were able to place the components on the board outline. The outline itself was initially set quite large, then as component placement was determined, shrunk back down to as small an area as possible. We placed all analog components at one end of the board and used a copper pour on the bottom to create a ground plane. After all the components were placed and oriented, we manually routed all the critical traces first and then optimized remaining traces to achieve a two layer board with minimal vias.

To achieve the most robust system, we separated the analog and digital sections with a ground break, used wide power and ground traces, took care not to introduce any ground loops, minimized inductance, used a range of power capacitors to ensure a steady clean power supply, and placed bypass capacitors as closely as possible to their components.

The top, bottom, and silkscreen Gerber images are shown at actual size in Figure 9-3.

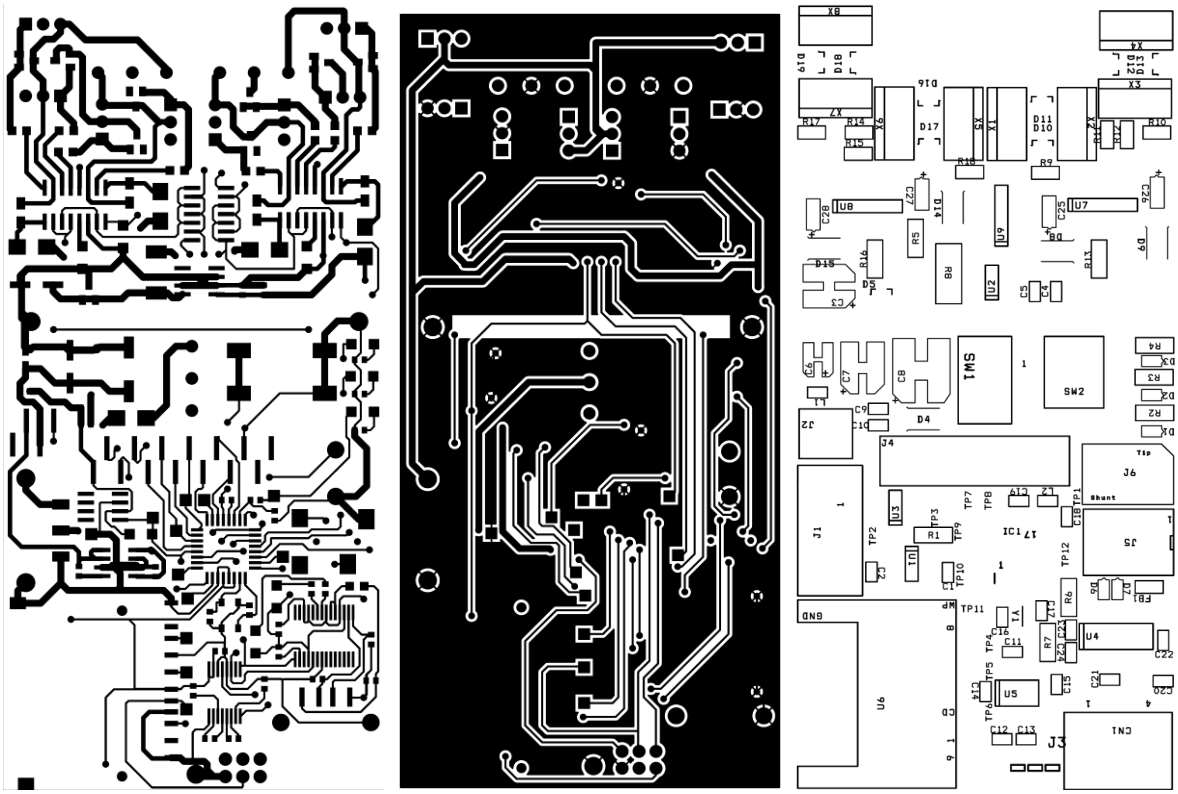


Figure 9-3: PCB layout

The Gerber files were sent to a fabrication facility which sent back completed FR2 PCBs measuring 5.5 cm x 11.5 cm x 2 mm. We manually applied solder paste to each pad and placed the components using tweezers and a magnifying headband. When

ready, the loaded board was carefully placed in a reflow oven with a profile set for the specific board components. Once cooled, the board was ready for inspection.

Qty	Reference	Manufacturer	Mfg Part #	Description	Package
8	X1 - X8	Fairchild Semiconductor	FDP7030BL	MOSFET N-CH 30V 60A	TO-220
1	SW1	NKK Switches	M2012ES1W03	SWITCH TOGGLE SPDT 6A PCB 5PCS	Through Hole
2	J7,J8	TE Connectivity	284093-3	TERM BLOCK 3POS TOP ENTRY 5MM	Through Hole
1	IC1	Atmel	ATMEGA328P-AU	MCU AVR 32K FLASH 32TQFP	32-TQFP
1	U4	FTDI, Future Technology Devices International Ltd	FT232RL-REEL	IC USB FS SERIAL UART 28-SSOP	28-SSOP
1	FB1	Laird-Signal Integrity Products	MI0805K400R-10	FERRITE 1.5A 40 OHM 0805 SMD	0805 SMD
1	U1	Linear Technology	LT1763CS8-3.3#PBF	IC REG LDO 3.3V .5A 8-SOIC	8-SOIC
1	U3	ON Semiconductor	MC33164D-3G	IC SENSING CIRCUIT UV 3V 8-SOIC	8-SOIC
1	J6	Switchcraft Inc.	MDSL2A	CONN JACK PHONE MINI 2POS	DNP
1	U5	Texas Instruments	SN75C3221DBR	IC RS232 DRVR/RCVR 1-CH 16-SSOP	16-SSOP
1	Y1	TXC CORPORATION	7B-16.000MEEQ-T	CRYSTAL 16.000MHZ 10PF SMD	4-SMD
1	U6	AVX Corp/Kyocera Corp	145638009211859+	CONN MEMORY CARD NORMAL TOP PCB	
1	U9	ON Semiconductor	MC14069UBDG	IC INVERTER HEX P/N 14-SOIC	14-SOIC
2	U7,U8	Intersil	HIP4082IBZ	IC DRIVER FET H-BRIDGE 16SOIC	16-SOIC
1	U2	Texas Instruments	LM78L12ACM/NOPB	IC REG LDO 12V .1A 8-SOIC	8-SOIC
1	D5	NXP Semiconductors	BZX84-A27,215	DIODE VREG 250MW 27V SOT23	SOT23
8	D10,D11,D12,D13,D16,D17,D18,D19	NXP Semiconductors	BZX84-A15,215	DIODE VREG 15V 250MW SOT23	SOT23
4	D8,D9,D14,D15	Diodes Inc	B160-13-F	DIODE SCHOTTKY 60V 1A SMA	DO-214AC
6	R1,R2,R3,R4,R6,R7	Stackpole Electronics Inc	RMCF1206FT1K00	RES 1K OHM 1/4W 1% 1206 SMD	1206 SMD
8	R9,R10,R11,R12,R14,R15,R17,R18	Rohm Semiconductor	ESR10EZPJ510	RES 51 OHM .4W 5% 0805 SMD	0805 SMD
12	C5,C9,C10,C11,C12,C13,C14,C15,C18,C19,C23,C24	TDK Corporation	C1608X7R1E104K080AA	CAP CER 0.1UF 25V 10% X7R 0603	0603 SMD
1	C1	Yageo	CC0603KRX5R8BB105	CAP CER 1UF 25V 10% X5R 0603	0603 SMD
3	C2,C4,C20	TDK Corporation	C1608X7R1E103K080AA	CAP CER 10000PF 25V 10% X7R 0603	0603 SMD
2	C16,C17	Kemet	C0603C100K3GACTU	CAP CER 10PF 25V 10% NP0 0603	0603 SMD
4	C25,C26,C27,C28	Panasonic Electronic Components	EEE-HA1HR22AR	CAP ALUM 0.22UF 50V 20% SMD	Radial, Can - SMD
2	C21,C22	Kemet	C0603C470J3GACTU	CAP CER 47PF 25V 5% NP0 0603	0603 SMD
2	C3,C7	Kemet	EDK106M050A9GAA	CAP ALUM 10UF 50V 20% SMD	Radial, Can - SMD
1	C6	Cornell Dubilier Electronics	AVE475M25B12T-F	CAP ALUM 4.7UF 25V 20% SMD	Radial, Can - SMD
1	C8	Cornell Dubilier Electronics	AVEK476M50F24T-F	CAP ALUM 47UF 50V 20% SMD	Radial, Can - SMD
1	L1	TDK Corporation	MLG1608B47NJ	INDUCTOR MULTILAYER 47NH 0603	0603 SMD
1	L2	TDK Corporation	MLF1608E100K	INDUCTOR MULTILAYER 10UH 0603	0603 SMD
1	D4	Fairchild Semiconductor	S1K	DIODE GP 800V 1A DO214AC	DO-214AC
1	J5	CUI Inc	SJ-2523-SMT-TR	CONN AUDIO JACK 2.5MM STEREO SMD	Surface Mount RA
1	D1	Panasonic Electronic Components	LP L296-J2L2-25-Z	LED SMARTLED 560NM PURE GRN 0603	0603 SMD
2	D3,D6	OSRAM Opto Semiconductors Inc	LS L296-P2Q2-1-Z	LED SMARTLED RED 633NM 0603	0603 SMD
1	D7	OSRAM Opto Semiconductors Inc	LO L296-Q2S1-24-Z	LED SMARTLED ORANGE 606NM 060	0603 SMD
1	D2	OSRAM Opto Semiconductors Inc	LY L296-P1R2-26-Z	LED SMARTLED 587NM YLW 0603 SMD	0603 SMD
1	CN1	Assmann WSW Components	AU-Y1006-2-R	CONN USB 2.0 R/A FMAL TYPE-A SMD	Surface Mount RA
1	J4	3M	951114-2530-AR-PR	CONN HEADER 14POS 2MM VERT SMD	header
1	J1	Phoenix Contact	1727243	CONN TERM BLOCK 3POS 3.81MM SMD	header
1	J2	3M	951104-2530-AR-PR	CONN HEADER 4POS 2MM VERT SMD	header
1	SW2	C&K Components	PWR70Q1S	SWITCH PUSH SPST-NO 2A 125V	switch
1	R8	Bourns Inc.	CRM2512-JW-101ELF	RES 100 OHM 2W 5% 2512 SMD	2512 SMD
3	R5,R13,R16	Panasonic Electronic Components	ERJ-8ENF1203V	RES 120K OHM 1/4W 1% 1206 SMD	1206 SMD

Table 9-2: Bill of Materials

9.4. Validating the device

We began by examining the board under a bright light with a 10x magnification headband for any unconnected pins, bridges (cross connected adjacent pins), solder balls or spatters, tombstones (lifted components), and grainy/cold joints. With manually applied solder paste, there is always some combination of these defects; they are easily corrected with a reflow station, soldering iron, and solder braid. Once the surface mount components were visually inspected, we soldered the eleven through-hole components. The assembled board is shown in Figure 9-4.

Using printouts of the schematic and Gerber files, we checked all power and ground connections using the continuity setting on a digital multi-meter. Finally a pair of scanning tweezers was used to check cumulative resistance, capacitance and inductance values for each interconnected line. Once everything checked out, we prepared to power up the device.

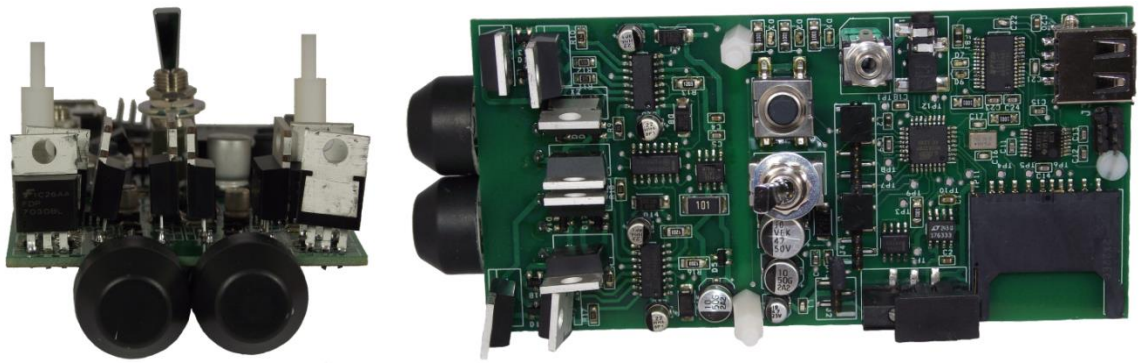


Figure 9-4: Photos of the device

The first power on test was for the digital circuitry only. We began with a 5V input to see if the power on LED would light up. When everything checked out, we connected a JTAG connector to the microprocessor and read the chip model and status. Atmel AVR microprocessors start in an on-chip clock mode which means they are

running quite slowly; the ICE programmer must be set to a very slow communication rate and then the chip can be configured to use the external crystal oscillator. Once that configuration was done, we flashed a simple program to blink all the LEDs and verified that they operated correctly. Next we uploaded a communications program which allows us to talk to the board via the USB or serial ports.

Once proper communication was established with the board, we uploaded a program that utilized the 16-bit timer to control the ultrasonic transducers. Timing for each transducer can be set by selecting left or right and then entering a 4 digit hexadecimal number representing the "on" time of the left H-bridge high side. At 16 MHz, each clock tick is 62.5 ns and we have an output resolution of approximately 200±40 Hz. For example, if we enter an input of 0x00DE, our output should be $222 \cdot 2 \cdot 62.5\ \text{ns} = 27.75\ \mu\text{s}$ or 36036 Hz.

To verify proper function of the control lines, we connected a Saleae Logic16 to the digital output lines and entered 0x00DE (36036 Hz) for the left transducer and 0x00B9 (43243 Hz) for the right. Figure 9-5 shows the oscilloscope timing; the output signals on average are correct, but due to chaining a single 16-bit timer, there is some occasional timing jitter that we need to minimize through careful programming.

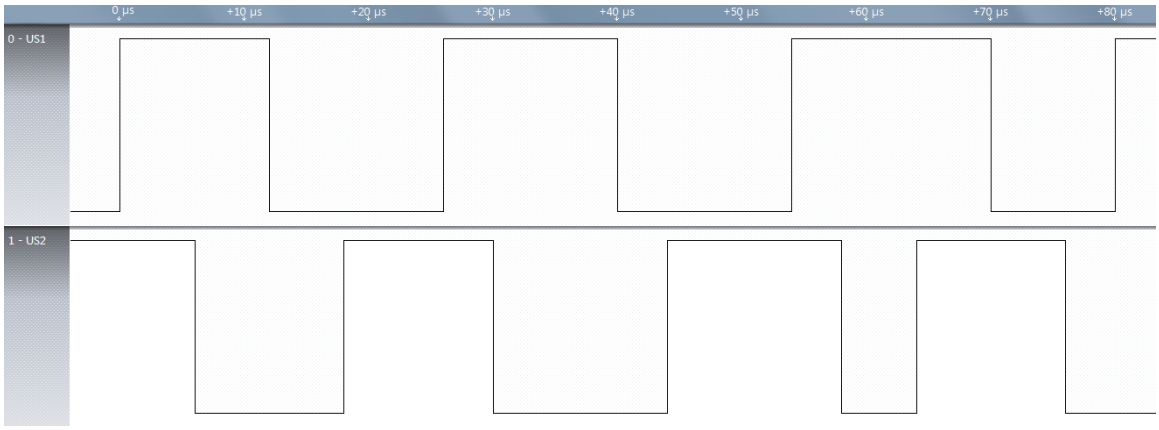


Figure 9-5: Output timing

Now that we had confidence in the digital circuitry, we connected the analog circuitry to a minimal 12.5 V input. As the digital oscilloscope is only rated to 6 V, we used a HP 1741A to observe the 25 V_{PP} MOSFET outputs. Once we verified that the analog output was following the digital control signals, we were finally ready to connect the ultrasound transducers.

9.5. Preliminary Results

Using the 36036 Hz and 43243 Hz settings, we used the UA-25 and Adobe Audition to create a spectrogram of the device output. Figure 9-6 shows the spectrogram where the two signals can be seen as bright lines (high content) against a white background (low content). This test was not run in the sound box, however, there wasn't much interference at these frequencies.

Figure 9-6: Device output

As a prototype phantom, we constructed a trachea out of corrugated polyethylene with an inner diameter of 24.5 mm which corresponds well to the average size diameter of the male trachea (27 ± 3 mm x 25 ± 3 mm). For soft tissue, we used a latex torus filled with distilled water. In a review of tissue substitutes, Culjat et al. indicate that water is a common phantom material but suffers from the speed of sound having a strong temperature dependence and a low attenuation coefficient. [119] Furthermore, as Fatemi et al. indicate, the latex has minimal effect on the ultrasonic wave and the main effect will be from distilled water which has a density of 993.3 kg/m^3 and a speed of sound of 1524 m/s at 37°C (body temperature) which is very near the typical soft tissue values of 1000 kg/m^3 and 1540 m/s as shown in Table 6-1. [70] The thickness of the torus around the opening was carefully constrained and measured at 21.55 mm using a set of Duratool DC150 calipers. This is a good average value considering our measurements of the AustinMan ranged from 18 – 24 mm ($z \in [251,264]$). Figure 9-7 shows the phantom. Figure 9-8 shows the device in its testing configuration with the transducers pressed firmly against the phantom and causing deformation of the material as would happen with soft tissue.

Figure 9-7: Trachea phantom

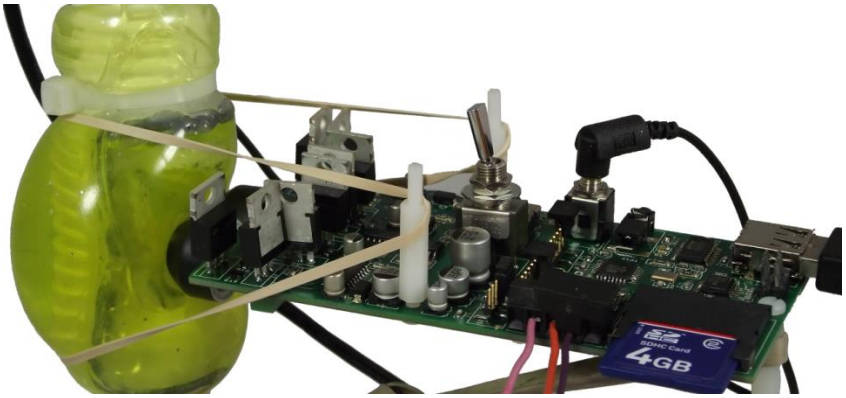


Figure 9-8: Testing configuration

As a proof of concept, we setup the phantom with a condenser microphone inside, connected the device to 12.5 V, selected the 36036 Hz and 43243 Hz output frequencies and recorded using the UA-25 and Adobe Audition. As can be seen in Figure 9-9, as the device nears the phantom, the output signals are seen, once the device is in contact with the phantom, the interior surface is mechanically coupled to the difference wave and begins to vibrate which gives rise to the lower intensity signal around 8 kHz. Based on the inputs, we expected to see a 7207 Hz wave and we measured at 7260 Hz a difference of only 53 Hz.

Figure 9-9: Initial phantom test

10. Testing the Physical Device

Now that we have a verified working device, we can begin to optimize it and run full tests with the phantom in our anechoic sound box.

10.1. Optimizing the Embedded Code

Given a microprocessor like the Atmel ATmega328P, there are many ways to implement the embedded timing algorithm. In this section, we will examine some of the algorithms we conceived and tested. Our evaluation method includes writing the code, compiling using the ImageCraft ICCAVR version 7.23 IDE, examining the generated assembly code and counting the instruction cycles, and finally uploading the code and evaluating the resulting timing signals using a Saleae Logic16 and a VC3165 high resolution counter. [120] Instruction timing is calculated using the Atmel timing guide which includes timing for all instructions; careful attention is required to ensure that the correct cycle count is selected as the same instruction may have different counts depending on the address mode, length of operands, and branching. [121]

We have developed our algorithms to generate square waves with a 50% duty cycle which results in stable signals; however, our timing resolution is not as good as we would like due to the frequency of the generated signal and the system clock frequency. Since our transducers have maximum output around 40 KHz, most of our output signals will be within a few hundred hertz of that center frequency. With a 16 MHz clock, a 40 KHz signal takes only 400 cycles; with a 50% duty cycle this equates to 200 on cycles and 200 off. A 40.1 KHz signal requires 399 cycles which cannot be generated with a 50% duty cycle. For this reason, we developed some algorithm variations which allow an extra cycle to be added to the off half; we refer to this capability as dynamic sub-cycle.

Most of the algorithms require the output frequencies to be specified at compile time; we also developed a set that allow this to be done via the serial/USB link. These algorithms have a postfix of *var*. A summary of each algorithm and the resulting timing is shown in Table 10-1.

Algorithm	Methods	Timing (Cycles)
1	16-bit timer + static flag + AND	68 / 69
2	16-bit timer + re-ordered	62 – 65
3	No interrupts	55 / 56
4	16-bit timer + dedicated register + XOR & check	43 / 44
5	16-bit timer + dedicated register + direct check + sub-cycle	33 / 34 | 34 / 35
5 var	16-bit timer + dedicated register + direct check	33 / 34
6	16-bit timer + HW toggle + sub-cycle	29 / 29 | 32 / 33
6 var	16-bit timer + HW toggle	29
7 var	Independent 8-bit timers + HW toggle + NEG + dynamic sub-cycle	19

Table 10-1: Embedded Algorithm Timing

Algorithm 1 was our first timing algorithm; hand optimized using integer C best practices. It uses the output compare feature of the 16-bit timer in Normal mode. In each interrupt service routine (ISR), a static variable keeps track of whether the signal is on or off and manually sets the output pin. As both routines are tied to the same timer, the output compare register (A or B) is updated in each ISR. The algorithm takes 68 cycles for on and 69 for off due to the timing difference when the conditional branch is taken and not taken.

For Algorithm 2, we analyzed the assembly of Algorithm 1 and simply re-ordered some of the instructions to maximize register reuse. This re-ordering results in timing between 62-65 cycles depending on branching.

Rather than use interrupt routines, we created Algorithm 3 to run continuously in the main() function. We thought we might be able to get more stable timing without

chaining effects. However, even with a highly optimized routine, each complete cycle took between 55-56 cycles; this results in significant waveform errors when the rising/falling edges overlap. We determined that this type of implementation will never exceed the advantages of using the built-in hardware capabilities.

We store the timing count required for the on/off duty cycle in a global word memory location; accessing the value requires two byte reads each requiring two cycles. A good way to reduce this overhead is to store these values in dedicated global registers. ICCAVR allows 4 byte registers to be reserved for global variables; these registers can be used in any legal combination of bytes and/or words. For Algorithm 4, we store our duty cycle in two global word registers. In order to determine whether we are in the on or off part of the duty cycle, we examined several options including addition, bitwise ANDs, Exclusive ORs, and flag variables; the EOR instruction has the lowest cycle count. We flip the flag variable, turn off the output line and the conditionally enable it based on the flag. These optimizations reduced the cycle count to 43 – 44 cycles depending on the branch condition.

When calling an ISR, the ICCAVR compiler automatically stores any registers that are modified in the routine on the stack and restores them on exit. Pragmas are available for preventing this behavior for normal functions, but it is not possible to prevent this for ISRs even when we know those registers do not need to be preserved. Therefore, a key optimization is reducing register use inside the ISRs. Algorithm 5 uses built-in hardware instructions to read the pin value itself which also allowed us to remove the local static variable inside each ISR. We also implemented compile time conditional statements which allow handling sub-cycle timing. When sub-cycle timing is not

activated, the routine takes 33-34 cycles; when activated it takes 34-35 cycles. Algorithm 5 var implements variable inputs using the same setup as Algorithm 5 but without the sub-cycle capability and therefore has only the 33-34 cycle timing.

For Algorithm 6, we take advantage of the timing hardware to automatically toggle the output pin. This ensures that our output is completely regular as long as in the worst case where both ISRs must be run concurrently their combined timing is less than half the duty cycle. As our typical duty cycle is 400 cycles, each ISR must be less than 50 cycles. Using the hardware toggle allows us to reduce the ISR timing to just 29 cycles without sub-cycling and 32-33 cycles when sub-cycling is active. In Algorithm 6 var, we remove the sub-cycling capability and add in variable inputs and the timing is always 29 cycles.

All the previous algorithms used the 16-bit timer; however, the microprocessor also has an 8-bit timer. The 8-bit timer only allows on/off duty cycles from 0 – 255 cycles; however, for our use this is acceptable as we are mainly interested in counts from 196 – 204. Using 8-bit timers means our variables now take half the cycles to load and we can use the global registers for four byte variables instead of two word variables. We use the extra global registers to implement dynamic sub-cycling; once the user has input a new frequency, the input routine calculates the on/off timing count and determines if an extra off cycle will result in more accurate timing. We analyzed multiple implementations and found a simple solution using the two's complement negate instruction which allows alternately adding and subtracting one from the cycle count. As we are still using the hardware output pin toggle feature, the ISR now requires no branches and always takes 19 cycles. This algorithm works correctly for signals from

31.3 KHz to 421 KHz. This is the algorithm we have used in our experiments. A sample of the output timing from the Saleae Logic16 is shown in Figure 10-1 for output frequencies 40100.25 Hz (with sub-cycling – on = 199 cycles / off = 200 cycles) and $40404.\overline{04}$ Hz (no sub-cycling – on/off = 198 cycles); the VC3165 shows the output as 40100.0 Hz (0.0% error) and 40403.8 Hz (0.000595% error).

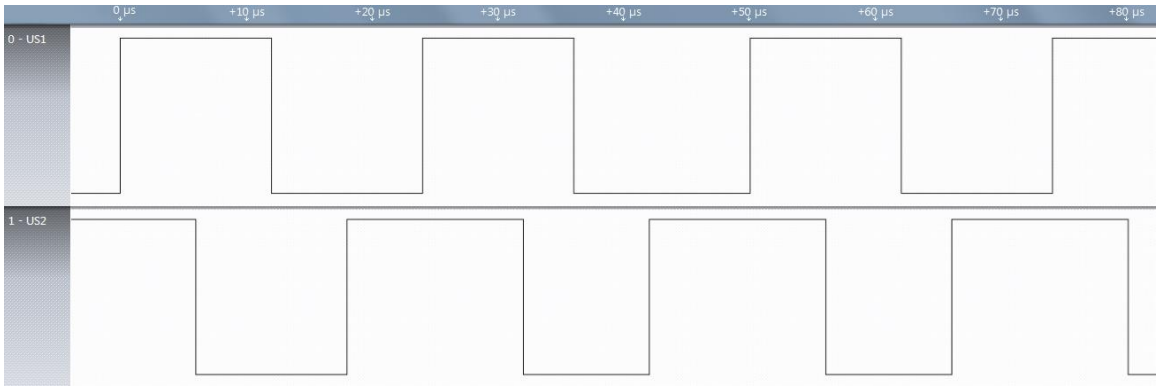


Figure 10-1: Timing Output for Algorithm 7 var with US1 = 40.1 KHz and US2 = 40.4 KHz

After completing the algorithm development and analysis, there are a few things we will do differently in future hardware. Although the ATmega328P can handle an external timing signal up to 20 MHz, we opted for a 16 MHz timer as we had used this configuration in the past and thought it would be easier to make work reliably on our prototype PCB. Changing to the 20 MHz signal requires no changes to the circuit and will provide a better timing signal with each clock tick being 50 ns rather than the 62.5 ns we have with the 16 MHz clock. And although we have been able to eliminate almost all issues caused by chaining the timer, if we modified the circuit, we could implement two independent timing signals each with a dedicated timer. In this configuration, we could use the Clear Timer on Compare (CTC) mode on each timer allowing timing to be completely generated in hardware. Such an implementation would remove any

restrictions on the upper bound of the signal frequency and should support 8-10 MHz depending on the system clock frequency.

10.2. Experimental Setup

During our preliminary tests, we noted that our equipment registered quite a bit of noise from 0 – 1 KHz. In order to ensure the most accurate readings, we created an anechoic chamber. Starting with a 2' x 2' x 2' corrugated cardboard box, we installed 2' x 2' x 2" wedge foam acoustic panels with a noise reduction coefficient of 0.80 in an alternating pattern covering all internal surfaces. In addition, we covered the outside with 4mm EP0M sound dampening acoustic foam. Figure 10-2 shows the inside of the box.

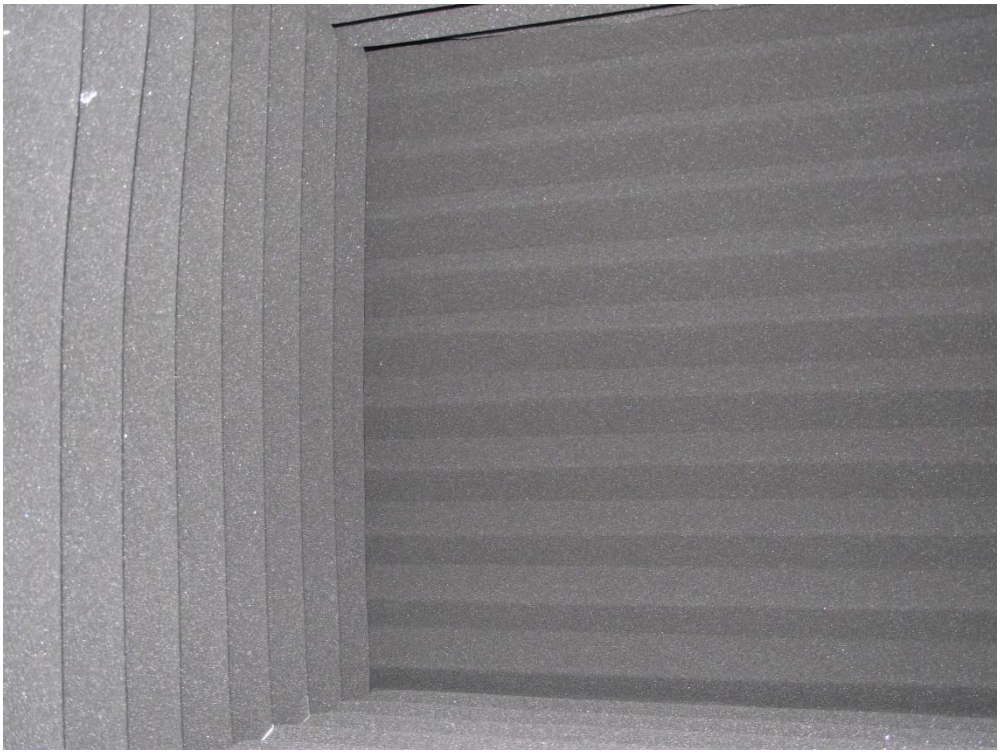


Figure 10-2: Anechoic Chamber

We used a Sper Scientific 850014 sound meter to measure the dBA and dBC levels within the box. Even with all the dampening material, the dBC level, which has a

flatter weighting scale in the low frequencies, was still too high. Low frequency noise is very difficult to stop and travels well through any solid surface. As the box was sitting on a table, the low frequency noise was travelling through the building and table and into the box. To minimize this, we placed the box on top of two foam pillows. This drastically reduced the dBC level. Even so, we were still picking up noise in the 0 – 200 Hz range. We went through the lab and shutdown all electronics; one surprising source of noise was the Halogen lights. A more difficult source was the HVAC. Eventually, we were able to have the HVAC shut off, turned off the lights, and shutdown all devices with fans such as our desktop computers. Using a fanless battery powered system and LED lamps, we were able to reduce the noise to a level that the sound meter could no longer measure – the specifications indicate a minimum of 30 dBA and 35 dBC; however, we were able to get readings down to about 25 dBA and dBC before the unit read "Und." The meter is extremely sensitive; with the meter enclosed in the sealed box, it is able to register an idling truck over two blocks away. For this reason, we ran our experiments late at night when traffic was very light.

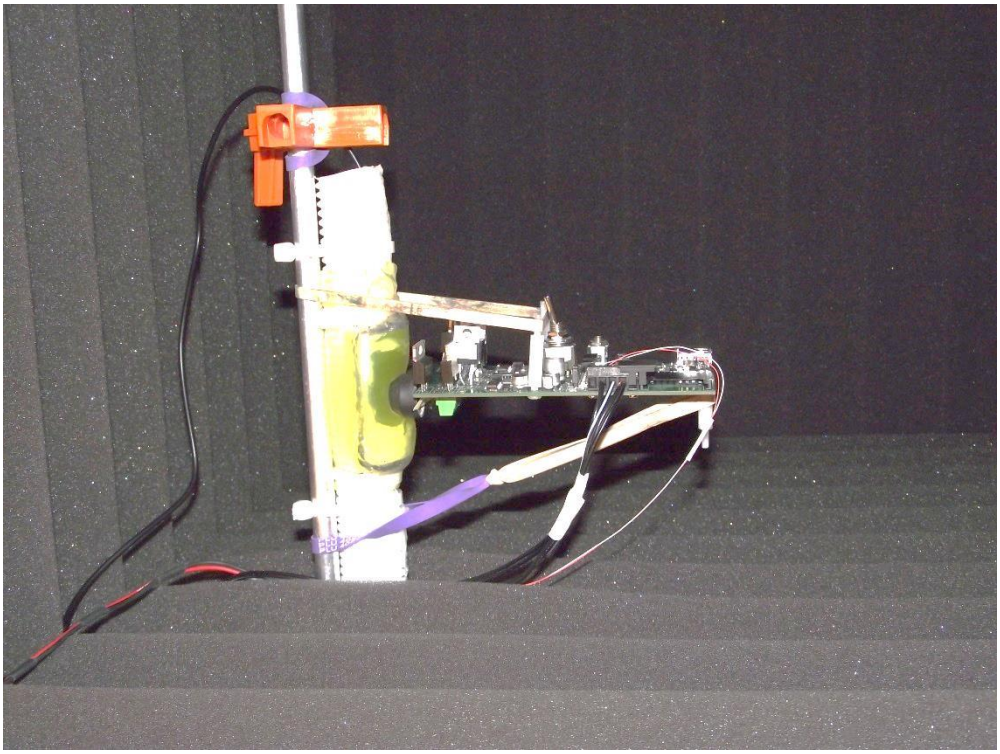


Figure 10-3: Phantom and Device in the Anechoic Chamber

Figure 10-3 shows the device attached to the phantom and mounted in the box. The construction of our phantom was described in Section 9.5; no changes were made for these experiments.

10.3. Results

In each of our experiments, we used two microphones, one located near the center of the phantom (but without touching the tubing or latex torus) and the other located outside the tube at the top opening. The microphones connect to the left and right channels of the UA-25 which feeds into the computer and are recorded using Adobe Audition. Without a calibrated signal source there is no way to compare our experiments to our simulations. For this reason, the sound meter was also used to take readings in both dBA and dBC modes; the sound meter was set to Slow (1 s) mode for all readings.

[122] These readings were measured at a number of input voltages. The results for inputs at 39600 Hz and 40000 Hz are summarized in Table 10-2. The results for inputs at 39900 Hz and 40100 Hz are summarized in Table 10-3.

Voltage (V)	Vpp	dBA	dBC
12.5	25.0	52.5	54.0
15.0	30.0	58.4	59.2
20.0	40.0	63.2	63.9
25.0	50.0	66.0	66.8
26.3	52.6	66.2	67.3
28.4	56.8	66.4	67.6
30.0	60.0	66.8	68.0

Table 10-2: Experimental Results for 39.6 KHz and 40.0 KHz

Voltage (V)	Vpp	dBA	dBC
12.5	25.0	34.4	45.8
15.0	30.0	36.2	46.9
20.0	40.0	46.2	50.2
25.0	50.0	62.5	62.9
26.3	52.6	62.8	63.4
28.4	56.8	63.7	64.8
30.0	60.0	63.8	65.1

Table 10-3: Experimental Results for 39.9 KHz and 40.1 KHz

Voltage readings all have an error of ± 0.1 V; the dBA and dBC readings all have an error of ± 1.5 dB.

10.4. Discussion

The Panasonic transducers are rated at 105 dB SPL @ 30 cm and 20 V_{RMS} with a reference of $2x10^{-5}$ Pa. 20 V_{RMS} is equivalent to 56.577 V_{PP}. Our dBC readings are higher than the dBA readings; this is expected as the dBA specification applies lower weighting coefficients to frequencies below 1 KHz while the dBC specification applies nearly flat weighting for all frequencies.

It is interesting to note that the readings for the 400 Hz beat frequency are consistently higher than those for the 200 Hz beat frequency. This discrepancy does not appear to be due to the frequency weighting as it occurs for both dBA and dBC. During experimentation we noted that pressing the device too firmly against the phantom yielded poor results. This effect is due to the construction of the transducer and how the shell resonates; a different design would be required to eliminate this effect. However, the output frequency for the individual transducers is nearly the same in both tests. This leads us to speculate that the mechanical coupling at the boundary is frequency dependent. This view is strengthened by the preliminary results where our beat frequency was about 7 KHz and showed none of these effects. We expected lower frequencies to couple better than higher frequencies. It is also possible that the latex has a frequency dependent response that is more pronounced at lower frequencies. Further research will be required to isolate the true cause and find a mitigating solution.

Another curious observation is that as we have increased the accuracy of our output frequencies, the magnitude of the output has gone down slightly. This effect has also been noted in the simulations where the highest output occurs during periods of transition or frequency change. It will be worth investigating the use of non-regular driving frequencies in future experiments to see if we can quantify and model this effect. And while these signals may result in higher output magnitudes, they may also lessen the linear control the current configuration offers. This would be similar to the tradeoffs made for classic piston driven electrolarynxes where the piston was disconnected from the strike plate to generate a higher output SPL but resulted in a nonlinear impulse.

11. Simulation vs. Physical Device

In order to close the loop and compare the results from our simulator and our experiments, we need to be able to compare similar configurations. In this chapter, we develop a digital model of the phantom, run simulations using this model and compare them to the results from the physical device.

11.1. Creating a model of the Phantom

The phantom is a fairly simple structure consisting of corrugated tubing and a water filled torus. We measured the phantom's various dimensions using a set of Duratool DC150 calipers. Using the measurements, we created a parametric model of the phantom using a set of simple geometric operations coded in Perl.

The tubing has a minimum diameter of 24.50 mm and a maximum diameter of 27.25 mm. Since our simulator has a resolution of 1 mm in the x, y, and z dimensions, we used diameters of 24 mm and 28 mm. The corrugation has a peak-to-peak length of 4 mm. The tubing has a length of 192 mm. A rectangular hole of 21 mm by 32 mm was removed from the middle of the tubing. The torus has an outer width of 22 mm and an inner width of 6 mm and a height of 58 mm.

Using these measurements, we can build the model one step at a time. First, we define a volume of 64 mm x 48 mm x 192 mm to hold our model; we fill this volume with air. Next, we create the corrugated tubing using Bresenham's circle algorithm and varying the radius to generate the corrugation. [123] Then we remove a rectangular hole by subtracting a volume of air from the tubing. Finally, we create the torus by intersecting two Bresenham's ellipses and filling the interior of the intersection with water. Figure 11-1 shows the final model from multiple views; the center image shows

the cross-section where the torus intersects the hole and bulges both inwards and outwards constrained by the elasticity of the latex and the polyethylene.

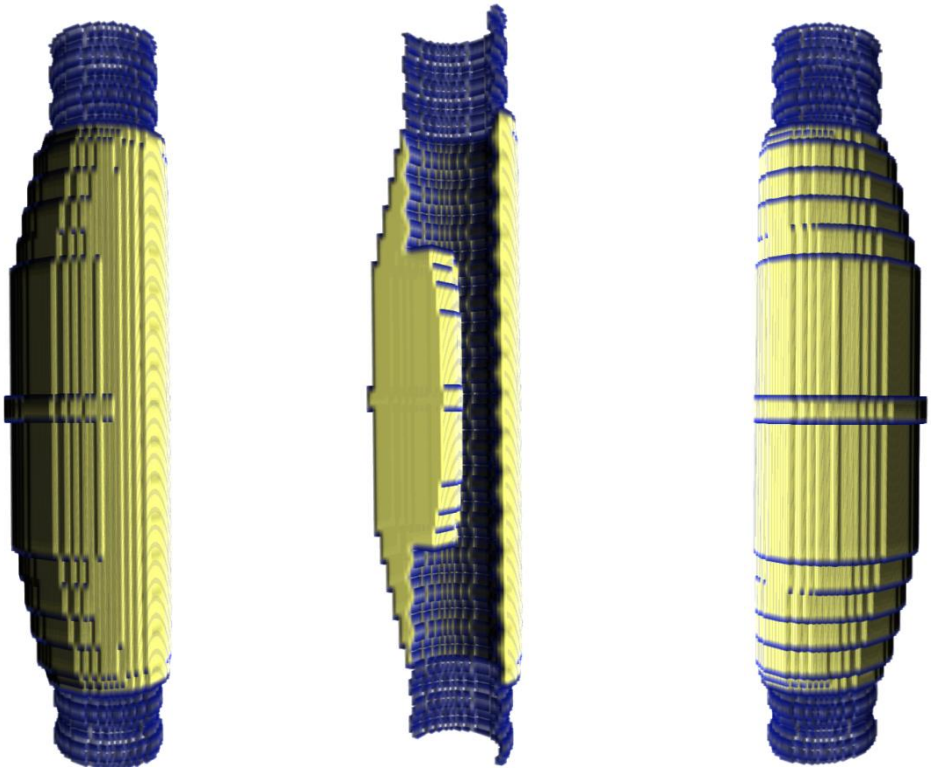


Figure 11-1: Model of the Phantom for Simulation

The material properties used to simulate the phantom are shown in Table 11-1. The source of each property or derivation is shown in the "Reference" column. The "Voxel Count" indicates how many voxels of this material type are present in the model. The phantom consists of a total of 589,824 voxels.

ID	Voxel Count	Name	Density (kg/m^3)	Speed of Sound (m/s)	Attenuation dB/cm/MHz	Reference
0	530,478	Air @ 25°C, 60% Rel. Hum	1.2	347	0.012	[81] ; [81] ; [81]
112	13,312	Polyethylene Split Loom Tubing	917.5	2250	17.37	[81]; avg [81] ; avg [81]
216	46,034	Distilled Water @ 37°C	1000	1520	0.0022	NIST ; (Marczak eqn); [82]

Table 11-1: Material Properties

11.2. Simulating the Phantom

Simulations were run using the linear 3D simulator as described in Section 7. Source 1 is located at (23 mm, 43 mm, 89 mm); source 2 is located at (41 mm, 43 mm, 89 mm). Both sources were modeled at 105 dB. The recording microphone is located at (32 mm, 11 mm, 89 mm); as we will be comparing these results with the experimental results, and we cannot be entirely sure of the exact x, y location of the microphone, we also ran the simulations at the front (32 mm, 6 mm, 89 mm), back (32 mm, 17 mm, 89 mm), left (17 mm, 17 mm, 89 mm), and right (43 mm, 11 mm, 89 mm) of the phantom. Each simulation was run for a minimum of 520K iterations (about 63 ms of simulated time) which models at least 25 complete cycles of the beat frequency at 400 Hz and 12 cycles at 200 Hz. Each run takes about 45 minutes to complete using the NVIDIA GeForce GTX 660 Ti GPU with double precision floating point values.

For the experimental values, we re-ran the experiments. However, unlike the first set of experiments where we sealed the device, sound level meter, and phantom in the sound chamber and ran all tests with that one setup, this time we ran many experiments (28 – 47 individual runs per beat frequency) and physically removed both the device and sound meter from the phantom after every few readings. Everything was then reset, sealed back in the sound chamber, and the readings continued. This allowed us to get a good sampling of different device couplings and meter placements. For these experiments, the device was powered at 30.0 ± 0.1 V.

As discussed in Section 10.2, we used a Sper sound meter to measure the device output in dBC. To calculate an equivalent value for the simulations, we used the algorithm from Appendix C of the ANSI S1.4 specifications. [122] The specification

allows calculating the appropriate weighting factor for any frequency. The standard is for a sampling rate of 44.1 KHz; however, the Sper sound meter only samples at 16 KHz. Therefore, we also used the lower rate for our calculations.

11.3. Results

For the simulations, we used the same source frequencies as we used in the device tests, namely 39600 Hz and 40000 Hz for the 400 Hz beat frequency and 39900 Hz and 41000 Hz for the 200 Hz beat frequency. In addition, we added an additional version of the 200 Hz beat frequency with sources at 39800 Hz and 40000 Hz to see if our earlier results were related to the frequency selection. For each set of frequencies, the results were averaged and the standard error calculated. Table 11-2 shows the results of the simulation at each beat frequency.

Source & Beat Frequencies	Simulation (dBC)	Device (dBC)
40.0 kHz + 39.6 kHz, 400 Hz	68.2 ± 2.1	70.9 ± 0.3
40.1 kHz + 39.9 kHz, 200 Hz	70.1 ± 0.6	66.6 ± 0.8
40.0 kHz + 39.8 kHz, 200 Hz	71.0 ± 0.9	70.9 ± 0.3

Table 11-2: Comparison of Simulation and Device output

11.4. Discussion

Figure 11-2 shows a graph of the dBC values. For the 400 Hz beat frequency, the simulation and device tests are very similar. For the 200 Hz beat frequency centered around 40 kHz, the device tests are below the value of the simulation and have the highest difference even considering the standard error. The second 200 Hz beat frequency with one source at 40 kHz performed much better with nearly perfect agreement between the experimental results and the simulations. As noted in Section

10.4, we are not sure if the lower device readings for the first 200 Hz beat frequency are caused by the choice of transducer which has a nominal center frequency at 40 kHz or a timing issue caused by the timer implementation for this particular set of frequencies.

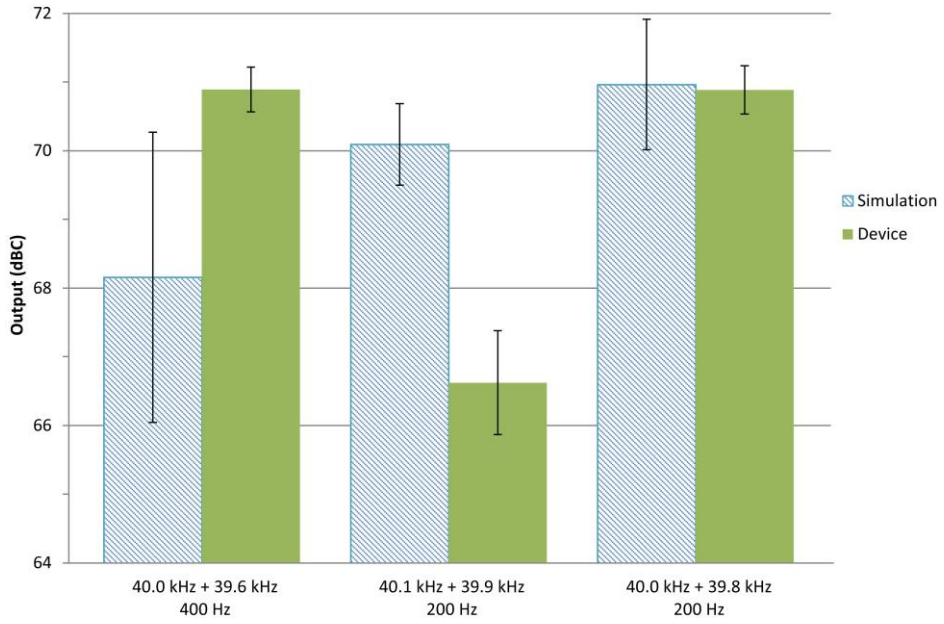


Figure 11-2: Device vs Simulation at multiple frequencies

Another effect we have noticed is that a more irregular output wave generates higher outputs; as we achieved such a regular output wave using version 7 var of our ISR algorithm, we have noticed a drop from our preliminary tests. We have not explored this relationship enough to draw any conclusions, but it will definitely be worth investigating.

It is worth noting that although we didn't achieve the sound levels that we wanted, in the 80 dBC range, we did achieve levels considered very loud. [124] For a first run preliminary device, this is a very encouraging start.

11.5. Sensitivity Analysis

We have strong confidence in the values used for distilled water and air as these are well documented within the temperature and humidity parameters provided.

However, the density and speed of sound parameters for plastics can vary greatly depending on the manufacturer. In order to determine what effect this might have on the results should our values not be completely accurate for the material used, we varied each parameter of the Polyethylene Split Loom Tubing by ±40% and re-ran the simulation at 400 Hz for y = 6mm and 105 dB.

	Density (kg/m^3)	**Speed of Sound (m/s)**	**dBA**	**dBC**
Reference result from Table 11-2	917.5	2250	63.15	67.70
Density down, Speed of Sound down	550.5	1350	63.37	67.86
Density up, Speed of Sound down	1284.5	1350	63.36	67.86
Density down, Speed of Sound up	550.5	3150	67.71	72.08
Density up, Speed of Sound up	1284.5	3150	67.70	72.06

Table 11-3: Polyethylene sensitivity analysis results

Varying the density had a negligible effect as did varying the speed of sound down; however, increasing the speed of sound in the polyethylene did result in higher simulated output levels but only by less than 6.5% and that was by increasing the speed of sound by 40% from 2250 m/s to 3150 m/s which is a fairly large increase. While not exhaustive, this analysis gives us confidence that our simulation outputs are not highly sensitive to changes in the polyethylene parameters. Researchers using similar materials from different manufactures should not see a large deviation from the results provided here due to variations in the polyethylene parameters.

12. Glottal Waveform and Classic EL Comparison

To compare the results from our simulations using the AustinMan model and our experiments with our device and phantom to healthy speakers and existing external voice restoration technologies, we model a glottal waveform and capture data from a classic electrolarynx.

12.1. Experimental Setup

For the comparison, we have four different sources of data: our simulations, our device experiments, a model of a glottal waveform and a classic electrolarynx. The following sub-sections describe the configuration for each source used for collecting data for fundamental frequencies of 100 Hz, 200 Hz and 400 Hz as appropriate.

12.1.1. Device Simulation

Simulations were run using the linear 3D simulator as described in Section 7. In order to better match the simulated glottal waveform and measurements of the classic electrolarynx, we have added additional sampling locations. For each of the beat frequencies, we recorded samples at the following six locations within the AustinMan trachea: (259 mm, 167 mm, 278 mm), (259 mm, 172 mm, 278 mm), (250 mm, 176 mm, 278 mm), (266 mm, 184 mm, 278 mm), (259 mm, 179 mm, 278 mm), (259 mm, 185 mm, 278 mm). To examine the acoustic leakage, we also recorded samples at (259 mm, 252 mm, 278 mm) which is 5 cm from the device/neck interface.

The transducers are located in the same locations as previous simulations; both sources have amplitudes of 105 dB. Source 1 is located at (249 mm, 202 mm, 278 mm) and source 2 at (269 mm, 202 mm, 278 mm). Source 1 is 40 kHz @ 0° phase and source

2 is 90° phase; for 100 Hz, source 2 is 39.9 kHz; for 200 Hz, it is 39.8 kHz; and for 400 Hz, it is 39.6 kHz.

The simulation for each beat frequency was run for a minimum of 520K iterations (about 63 ms of simulated time) which models at least 25 complete cycles of the beat frequency at 400 Hz, 12 cycles at 200 Hz, and 6 cycles at 100 Hz. The results from all recording locations were averaged and the standard error calculated.

12.1.2. Device

Measurements with the device were taken within the anechoic sound chamber as described in Section 10. For the first set of measurements, we set the transducers at 39.9 KHz and 40.1 KHz which resulted in a difference wave of 200 Hz. For the second run, we used input frequencies of 39.6 KHz and 40.0 KHz; which resulted in a difference wave of 400 Hz. As mentioned previously, due to early design decisions we are unable to take measurements at 100 Hz.

12.1.3. Glottal Waveform

There are a number of well-known glottal pulse models including: Rosenberg, Fant, Liljencrants-Fant, and Klatt [125]. Each model uses different parameters to represent the glottal cycle and therefore results in slightly different waveforms. For our purposes, we are interested in the pulse period, and the opening and closing phases; in particular, we want a continuous function with a discontinuous first-derivative as this combination results in the asymptotic decay of 12 dB/octave found in natural speech [126]. One of the simplest models with these properties is the Rosenberg pulse shape type

C [127]. We used an amplitude of 105 dB, an opening time T_P = 40%, and a closing time T_N = 16% for both of the fundamental frequencies.

We are interested in the excitation pulse only, not the vocal tract resonances / formants as these are physiological and for a given position / configuration will be the same irrespective of the excitation [128].

12.1.4. Classic Electrolarynx

For comparison to a typical classic piston electrolarynx, we followed Qi and Weinberg and utilized the popular Servox® Inton. [129] We recorded the device at 96 KHz in Adobe® Audition using an Edirol UA-25 digitizer. Sound level readings were made with a Sper Scientific 850014 sound meter in dBC slow mode.

The device has a limited range of frequencies, approximately 70 – 260 Hz, so we were not able to generate data for comparison with the 400 Hz measurements.

12.2. Results

In Figure 12-1, we have arranged a series of four graphs (A, B, C, D) comparing the synthetic glottal pulse train (dashed green line), the simulated device (dot-dashed blue line), the prototype device (solid purple line), and the electrolarynx (dotted red line) for each fundamental frequency. For each graph, the inputs have been normalized to equal peak amplitudes.

The graphs in 12-1A show the excitation pulse in the time domain. If the pulse generated by the device simulation/prototype was an actual glottal pulse, it would be considered breathy as there would be little glottal closure; however, as the excitation will be imparted mechanically, there will be no breathy effect.

The graphs in 12-1B show the spectrum of each excitation pulse. As noted by Qi and Weinberg, the electrolarynx suffers from a deficit of low-frequency energy which can be seen by the substantial dropoff in magnitude near the fundamental frequency. The graphs in 12-1C show the spectrum as a plot of SPL with a reference of 20 μPa. Both 12-1B and 12-1C show that the spectrum of the electrolarynx does not resemble the glottal pulse, while the device simulation and prototype are very close approximations. To quantize the comparison, we ran a correlation between pairs of the normalized spectral magnitudes from 12-1B; Table 12-1 shows the results which support the conclusion that the simulated device and prototype device generate a spectrum closer to a glottal pulse than does a classic electrolarynx.

Fundamental Frequency	**Sim Glottal Pulse vs. Electrolarynx**	**Sim Glottal Pulse vs. Device Simulation**	**Glottal Pulse vs. Device**
100 Hz	-0.16	0.94	N/A
200 Hz	0.06	0.96	0.97
400 Hz	N/A	0.97	0.95

Table 12-1: Correlation between normalized spectrum magnitudes

12-1D shows the harmonic roll off. As noted earlier, natural speech has a roll off approximated by −12 dB/octave. The electrolarynx increases with frequency; in fact, Weiss *et al.* [95] state that a roll off of −14±2 dB/octave occurs as the frequency decreases, completely opposite of a natural spectrum. Table 12-2 shows the mean and standard deviation for the harmonic roll off.

Fundamental Frequency	**Simulated Glottal Pulse (dBC)**	**Electrolarynx (dBC)**	**Device Simulation (dBC)**	**Device (dBC)**
100 Hz	-11.8 ± 2.5	-5.7 ± 4.5	-9.7 ± 10.7	N/A
200 Hz	-11.7 ± 4.4	2.6 ± 7.4	-10.1 ± 7.9	-12.9 ± 4.5
400 Hz	-12.1 ± 6.7	N/A	-13.4 ± 4.0	-12.5 ± 5.0

Table 12-2: Harmonic roll off

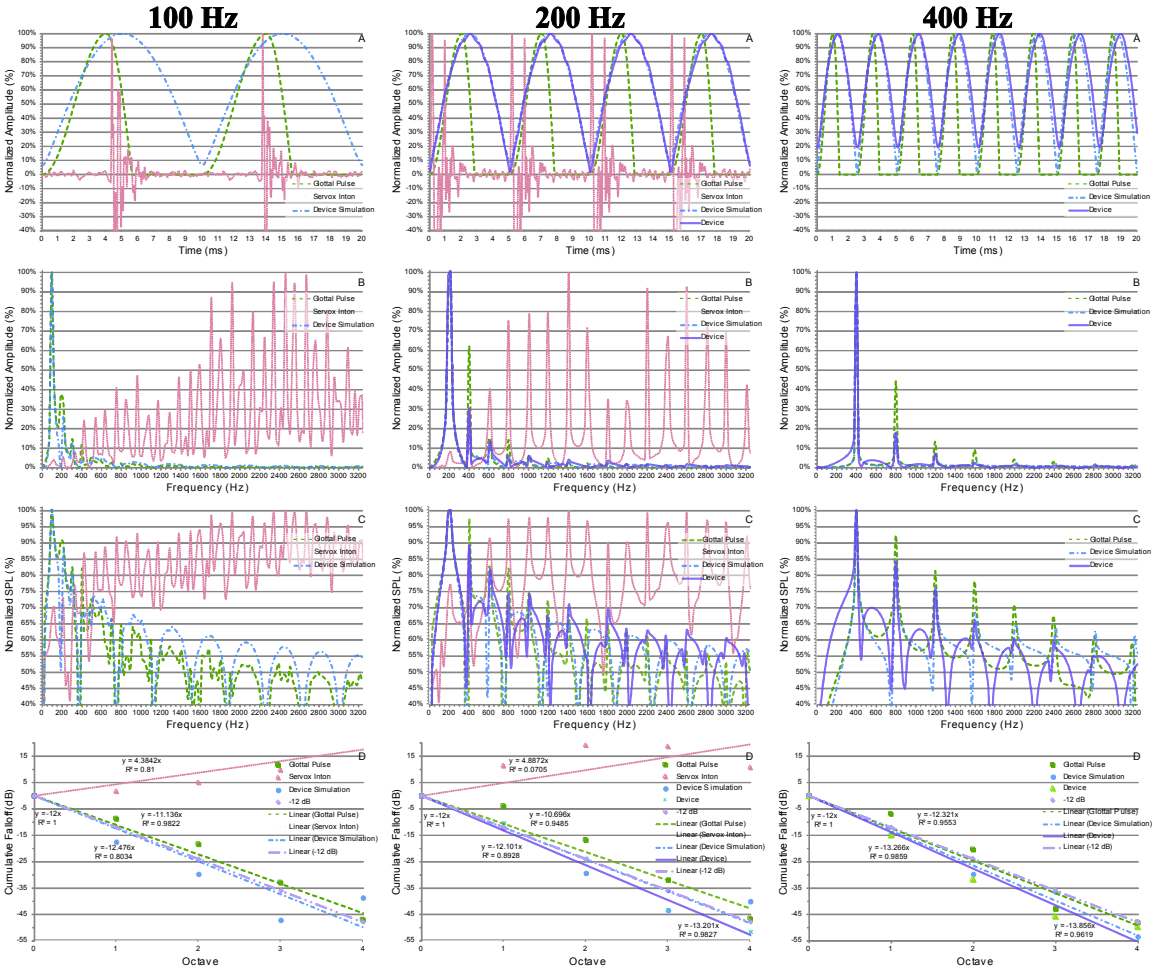


Figure 12-1: Comparison of peak normalized synthetic glottal pulses based on Rosenberg model (dashed green line), beat frequency from simulated device (dot dashed blue line), beat frequency from the prototype device (solid purple line), and a Servox® Inton piston electrolarynx (dotted red line).

A. time domain excitation waveform. B. spectrum, C. SPL spectrum *(ref. 20 µPa)*, D. Roll off per octave

Table 12-3 shows the comparison of self-noise between the electrolarynx and both the simulated and prototype devices at a distance of 5 cm. Lower acoustic leakage will increase listener comprehension. Furthermore, as the primary waveforms of our device are above human hearing, most communications devices will filter such noise as part of their standard digitization process.

Fundamental Frequency	Electrolarynx (dBC)	Device Simulation (dBC)	Device (dBC)
100 Hz	87.6 ± 0.1	63.9 ± 2.0	N/A
200 Hz	85.7 ± 0.1	64.0 ± 1.9	52.8 ± 0.1
400 Hz	N/A	63.0 ± 1.9	51.1 ± 0.1

Table 12-3: Self-noise output magnitudes at 5 cm

12.3. Discussion

Our tests show that a beat frequency can be generated using our technique with spectral characteristics that closely mimic a natural glottal pulse. Furthermore, two of the major complaints with classic piston-driven electrolarynxes are the difficulty and range of frequency control and the self-noise generated by the devices.

Given the method used to produce the fundamental frequency in our device, we anticipate that nearly linear control over the normal range of fundamental frequencies of males, females, and adolescents should be possible. For women in particular, our device's ability to generate even very high fundamental frequencies would allow for a more natural feminine sounding voice.

Our device's method for generating audible sound results in much lower self-noise. Only a small portion of the output will interfere outside the body and result in audible noise; the rest is outside the range of human hearing. Furthermore, unlike a classic electrolarynx where effective shielding is difficult; shielding in an ultrasonic device should result in a significant reduction in radiated noise. Further testing is necessary to test this hypothesis and determine to what extent it may be effective.

13. Conclusion

The novel ultrasonic device presented here offers many advantages over classical piston driven devices. As the input/output relationship is linear, frequency and amplitude control are excellent and offer a wide range which is in stark comparison to the highly nonlinear effects of the struck plate in a classic electrolarynx. And while our initial device was not small, it could be made much smaller by eliminating unnecessary components used for testing and utilizing the smallest trace sizes, multilayer fabrication and double sided assembly. Furthermore, custom designed MEMS transducers would offer a smaller and more efficient emitter. Ultimately, we believe this device if proved safe and effective could benefit those who have lost the ability to speak due to a laryngectomy.

13.1. Future Work

During the development and testing of our prototype, we noted some changes that might improve our system, some features that warrant further investigation, and some new experiments we would like to conduct.

For system changes, we would like to test the device with a higher system clock to improve the output resolution. In future versions, we may even use a different microprocessor with a much higher system clock; a resolution of about 50 Hz should be enough for most proof of concept devices, but to potentially match a patient's original voice, a resolution of 1-5 Hz would most likely be required. When we build another PCB, we will include the ability to select different signal paths to allow using all the hardware timing features which would allow better output signal generation.

We would also like to test a wide range of transducers to see if there are better options already commercially available. We would also like to investigate the time and costs associated with designing and fabricating a custom micro-electromechanical (MEMS) transducer tailored to our specific application. A transducer that does not distort its output when pressure is applied to the surface would improve our system's linearity and potentially increase the output level.

During both simulation and testing, we noticed that our highest output levels are during times when the signals are in flux – i.e. outside the steady state. When the transducers are first enabled, we see a much higher output; similarly, if the transducers are disabled or pulsed, we also see a higher output. As this occurs both in simulation and real-world testing, we feel it warrants further investigation to determine if the output has a useful frequency, if it is controllable, and if the higher levels are sustainable through continuous changes. If pulsing or frequency changes are advantageous, we would then need to determine the relationship between the pulse duration or magnitude of the frequency change and the change in the output.

A bigger project is to create a connected mesh for the viscoelastic FEM model. We still feel this is the best option for capturing and actively visualizing the mechanical coupling that occurs at the tissue air boundary. However, it may be necessary to add in additional tissue layers which requires a finer mesh size and additional tissue characterizations; the finer mesh needs either much longer to run or more computational resources and the tissue characterizations require significant time and resources. To satisfy the additional computational overhead, the current top-of-the-line NVIDIA Tesla K40 should reduce runtime by over half compared to our current hardware; the Tesla has

over twice the number of cores, twice the memory bandwidth, four times the memory and is dedicated to computation. Multiple units can be installed in the same machine and run as one larger unit with minimal synchronization.

For additional experiments, we would like to use professionally calibrated sound equipment to take measurements of the device in a sound studio. High-end equipment allows precise frequency analysis and calibration allows capturing the voltage input / SPL output curve which can then be used directly in the simulator. A device such as the ANSI/IEC Type 1 Casella CEL633C or Larson Davis 824 is capable of recording 33 one-third octave band measurements from 12.5 Hz to 20 KHz; these devices cost over $6,000 each. To directly measure the transducers a meter such as the Svantek SVAN979 with GRAS 40AM microphone would be required.

Our simple phantom has served us well in evaluating the prototype device. However, we are working on a new phantom that uses the AustinMan model which will allow us to compare the device to simulations using the AustinMan voxels. To date we have created the exterior mold which can be 3D printed. A hole in the top allows pouring in an agar or tofu mixture. [119] Figure 13-1 shows the four piece mold. We created the model by exporting the AustinMan voxels as 3D faces in STL format; these were then imported as a mesh into Solidworks. We used the Mesh Prep Wizard to orient, clean, reduce and smooth the mesh. We then used the Curve Wizard to create profile slices. We created a half-face surface using the "Lofted Surface" command and filled the back of the surface using the "Filled Surface" command. We then created a plane at the open side of the surface and mirrored across the plane and knitted the surfaces together. We generated a solid body using the "Thicken Surface" command. Finally, we used the

Mold sub-system to generate the mold, manually splitting the final output into four pieces and adding pegs to allow extracting the solid phantom without damage. We are currently working on creating a model of the vocal tract. As the tract must be a void in the final phantom we are considering using a material we can easily dissolve rather than splitting the phantom; a sugar or frozen sand mixture a probable candidates.

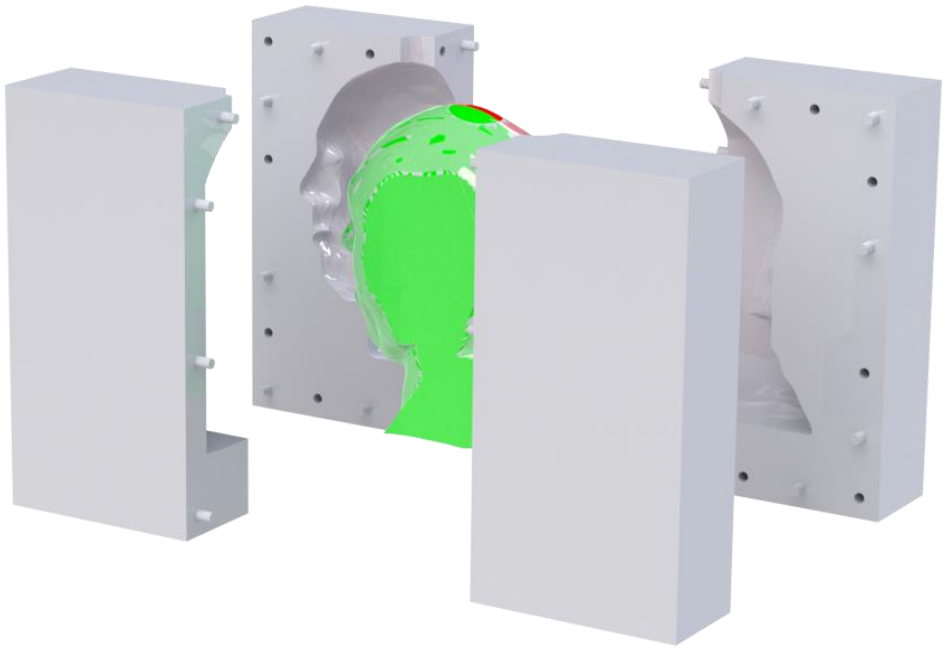


Figure 13-1: Mold for an AustinMan based Phantom

13.2. Our Vision

While effective, the device presented here is just a proof of concept. We envisage a commercial device with a purposely designed and manufactured parametric ultrasound transducer the diameter of a penny or less. We also foresee the day when technology has advanced enough to reliably monitor the laryngeal nerve for actuation input. The device shown in Figure 13-2 has a parametric transducer on one side and a surface EMG sensor on the other; both heads are mounted on flexible arms. One side of the device houses a

capacitive on/off switch and on the other an inductive recharger. The device is worn around the neck like a choker or neckwire necklace.

Figure 13-2: Rendering of future commercial device

The device would be weighted to sit comfortably on the neck and keep the transducers and sensors firmly in place. Should coupling gel be required, it could be stored in a container within the device and automatically applied like an inkjet printer when the device detected poor penetration.

The device would be self-calibrating; when the user activated the device with a specific touch sequence, the device would listen for speech and monitor echoes from the ultrasonic pulses to determine the best steering angles for actuation waveforms.

The unstoppable march of technology assures that the processing power to perform these actions will arrive at some near future date. We hope to see some incarnation of our device helping people live better lives in the near future.

References

[1] P. W. Alberti M.B., "Panel discussion: II. the evolution of laryngology and laryngectomy in the mid-19th century," *Laryngoscope,* vol. 85, no. 2, pp. 288-298, 1975.

[2] S. Bien, A. Rinaldo, C. E. Silver, J. J. Fagan, L. W. Pratt, C. Tarnowska, E. Towpik, N. Weir, B. J. Folz and A. Ferlito, "History of voice rehabilitation following laryngectomy," *Laryngoscope,* vol. 118, no. 3, pp. 453-8, 2008.

[3] G. S. Meltzner, Perceptual and Acoustic Impacts of Aberrant Properties of Electrolaryngeal Speech, Ph.D. dissertation, Health Sciences and Technology Speech and Hearing Biosciences and Technology, MIT, Cambridge, MA, 2003.

[4] M. van der Torn, A sound-producing voice prosthesis, Ph.D. dissertation, Dept. of Otolaryngology / Head and Neck Surgery, Free University, Amsterdam, The Netherlands, 2005.

[5] Sound from ultrasound : the parametric array as an audible sound source, Ph.D. dissertation, Media Arts & Sciences, School of Architecture and Planning, MIT, Cambridge, MA: Massachusetts Institute of Technology, 2002.

[6] W. A. Lea, Ed., Trends in Speech Recognition, Prentice-Hall, 1980, p. 6.

[7] Newell, Allen; et al., Speech Understanding Systems: Final Report of a Study Group, North-Holland Publishing, 1973, p. 50.

[8] A. Chapanis, "Interactive Human Communication," *Scientific American,* vol. 232, no. 3, pp. 36-42, 1975.

[9] P. T. Maddox and L. Davies, "Trends in total laryngectomy in the era of organ preservation: a population-based study.," *Otolaryngol Head Neck Surg.,* vol. 147, no. 1, pp. 85-90, 2012.

[10] I. R. Titze, J. G. Švec and P. S. Popolo, "Vocal Dose Measures: Quantifying Accumulated Vibration Exposure in Vocal Fold Tissues," *J Speech Lang Hear Res.,* vol. 46, no. 4, p. 919–932, 2003.

[11] J. C. Stemple, L. E. Glaze and B. G. Klaben, Clinical Voice Pathology: Theory and Management 3rd edition, San Diego, CA: Singular Publishing Group, 2000.

[12] C. Formby and R. B. Monsen, "Long-term average speech spectra for normal and hearing-imparied adolescents," *J Acoust Soc Am.,* vol. 71, no. 1, pp. 196-202, 1982.

[13] P. M. Mills, Fuzzy Speech Recognition, B.S. thesis, Dept. ECE, University of South Carolina, Columbia, SC, 1994.

[14] H. Gray, F.R.S., Gray's Anatomy, Fifteenth ed., R. Howden, M.A., M.B., C.M. and T. P. Pick, F.R.C.S., Eds., Barnes & Noble Books, 1995, 1901, pp. 939-954.

[15] National Cancer Institute, "SEER Training: Thyroid," U.S. National Institutes of Health, [Online]. Available: http://training.seer.cancer.gov/head-neck/anatomy/thyroid.html.

[16] A. C. Guyton, M.D. and J. E. Hall, Ph.D., Textbook of Medical Physiology, Tenth ed., W.B. Saunders Company, 2000, pp. 440-441.

[17] E. Breatnach, G. C. Abbott and R. G. Fraser, "Dimensions of the Normal Human Trachea," *American Journal of Roentgenology,* vol. 142, no. 5, pp. 903-906, 1984.

[18] C. H. Chen, Ed., Digital Waveform Processing and Recognition, CRC Press, 1982, pp. 50-51.

[19] University of Iowa Health Care, "Total Larynegectomy - Iowa Head and Neck Protocols," [Online]. Available: https://wiki.uiowa.edu/display/protocols/Total+Laryngectomy.

[20] C. Espy-Wilson, V. Chari and C. Huang, "Enhancement of Alaryngeal Speech by Adaptive Filtering," in *Proc. 4th Int'l Conf. on Spoken Language*, vol. 2, pp. 764-767, 1996.

[21] K. Ooe, T. Fukuda and F. Arai, "New Type of Artificial Larynx using PZT Ceramics Vibrator as Sound Source," in *IEEE/ASME Int'l Conf. on Advanced Intelligent Mechatronics*, pp. 114-119, 1999.

[22] S. M. Robertson, J. C. L. Yeo, C. Dunnet, D. Young and K. MacKenzie, "Voice, swallowing, and quality of life after total laryngectomy: results of the west of Scotland laryngectomy audit," *Head Neck,* vol. 34, no. 1, pp. 59-65, 2012.

[23] Servox, "Servox Inton," [Online]. Available: http://www.servona.com/UserFiles/File/PI_SERVOX_INTON_gb.pdf.

[24] D. Alzamora, D. Silage and R. Yantorno, "Implementation of a Software Model of the Human Glottis on a TMS32010 DSP to Drive and Artificial Larynx," in *Proc. IEEE 19th Ann. NE Bioengineering Conf.*, pp. 212-214, 1993.

[25] A. Lobo and M. O'Malley, "Towards a Biomechanical Model of the Larynx.," in *Proc. 4th Int'l Conf. on Spoken Language*, vol. 1, pp. 279-282, 1996.

[26] F. Alipour-Haghighi and I. Titze, "A Finite Element Simulation of Vocal Folds Vibrations," in *Proc. 14th Ann. NE Bioengineering Conf.*, pp. 186-189, 1988.

[27] K. Ooe, T. Fukuda and F. Arai, "Artificial Larynx using PZT Ceramics Vibrator as Sound Source (The Characteristic Improvement of PZT Cermaics Vibrator)," in *Proc. 2000 Int'l Symp. on Micromechatronics and Human Science*, pp. 189-194, 2000.

[28] R. Myrick and R. Yantorno, "Vocal Tract Modeling as Related to the Use of an Artificial Larynx," in *Proc. IEEE 19th Annual NE Bioengineering Conf.*, pp. 75-77, 1993.

[29] C. Ramalingam and R. Kumaresan, "Voiced-Speech Analysis Based on the Residual Interfering Signal Canceler (RISC) Algorithm," in *IEEE Int'l. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 473-476, 1994.

[30] C. Veaux, J. Yamagishi and S. King, "Voice banking and voice reconstruction for MND patients," in *ASSETS '11 Proc. 13th Intl. ACM SIGACCESS Conrf. on Computers and accessibility*, pp. 205-206, Dundee, Scotland, UK, 2011.

[31] A. Mallis, P. D. Goumas, N. S. Mastronikolis, T. Panogeorgou, T. Stathas, K. Prodromaki and T. A. Papadas, "Factors influencing quality of life after total laryngectomy: a study of 92 patients," *Eur Rev Med Pharmacol Sci.,* vol. 15, no. 8, pp. 937-42, 2011.

[32] E. J. O. Ten Hallers, H. A. M. Marres, G. Rakhorst, R. Hagen, A. Staffieri, B. F. A. M. Van Der Laan, E. B. Van Der Houwen and G. J. Verkerke, "Difficulties in the fixation of prostheses for voice rehabilitation after laryngectomy," *Acta Otolaryngol,* vol. 125, no. 8, pp. 804-13, 2005.

[33] H.-S. Choi, Y. J. Park, S. M. Lee and K.-M. Kim, "Functional Characteristics of a New Electrolarynx "Evada" Having a Force Sensing Resistor Sensor," *Journal of Voice,* vol. 15, no. 4, pp. 592-599, 2001.

[34] P. Pandey, S. Bhandarkar, G. Bachher and P. Lehana, "Enhancement of Alaryngeal Speech using Spectral Subtraction," in *14th Int'l Conf. on Digital Signal Processing*, vol. 2, pp. 591-594, 2002.

[35] K. F. Nagle, T. L. Eadie, D. R. Wright and Y. A. Sumidaa, "Effect of Fundamental Frequency on Judgments of Electrolaryngeal Speech," *Am J Speech Lang Pathol,* vol. 21, no. 2, pp. 154-166, 2012.

[36] M. Merlo, G. P. Li and M. Bachman, "A Remotely Powered and Wirelessly Controlled Intraoral Electrolarynx," in *Conf Proc IEEE Eng Med Biol Soc.*, Vancouver, 2008.

[37] F. Ahmadi, I. V. McLoughlin and H. R. Sharifzadeh, "Linear predictive analysis for ultrasonic speech," *Electronics Letters,* vol. 46, no. 6, pp. 387-388, 2010.

[38] J. T. Heaton, E. A. Goldstein, J. B. Kobler, S. M. Zeitels, G. W. Randolph, M. J. Walsh, J. E. Gooey and R. E. Hillman, "Surface electromyographic activity in total laryngectomy patients following laryngeal nerve transfer to neck strap muscles," *Ann Otol Rhinol Laryngol,* vol. 113, no. 9, pp. 754-64, 2004.

[39] E. A. Goldstein, J. T. Heaton, J. B. Kobler, G. B. Stanley and R. E. Hillman, "Design and implementation of a hands-free electrolarynx device controlled by neck strap muscle electromyographic activity," *IEEE Trans Biomed Eng,* vol. 51, no. 2, pp. 325-32, 2004.

[40] C. E. Stepp, J. T. Heaton, R. G. Rolland and R. E. Hillman, "Neck and face surface electromyography for prosthetic voice control after total laryngectomy," *IEEE Trans Neural Syst Rehabil Eng,* vol. 17, no. 2, pp. 146-55, 2009.

[41] J. T. Heaton, M. Robertson and C. Griffin, "Development of a wireless electromyographically controlled electrolarynx voice prosthesis," in *Conf Proc IEEE Eng Med Biol Soc*, 2011.

[42] G. Duncan, B. Yegnanarayana and H. A. Murthy, "A nonparametric method of formant estimation using group delay spectra," in *Proc. ICASSP*, May 1989.

[43] L. C. Wood and D. J. Pearce, "Excitation Synchronous Formant Analysis," in *IEE Proc. Communications, Speech, and Vision*, 1989.

[44] D. Cole, S. Sridharan, M. Moody and S. Geva, "Application of Noise Reduction Techniques for Alaryngeal Speech Enhancement," in *IEEE TECON, Speech and Image Technologies for Computing and Telecommunications*, 1997.

[45] T. Ifukube, "A Neuroscience-Based Design of Intelligent Tools for the Elderly and Disabled," in *Proc. EC/NSF Workshop on Universal Accessibility of Ubiquitous Computing*, 2001.

[46] W. D. O'Brien, Jr., "Fundamentals of Engineering Acoustics Lecture Notes," Univ. of Illinois (UIUC) Bioacoustics Research Laboratory (BRL), 2011.

[47] P. M. Gerhart, R. J. Gross and J. I. Hochstein, Fundamentals of Fluid Mechanics, New York: Addison-Wesley, 1992.

[48] A. R. Rasmussen, M. P. Sørensen, Y. B. Gaididei and P. L. Christiansen, "Analytical and numerical modeling of front propagation and interaction of fronts in nonlinear thermoviscous fluids including dissipation," *arXiv: 0806.0105v2 {physics. flu-dyn},* 2008.

[49] A. R. Rasmussen, Thermoviscous Model Equations in Nonlinear Acoustics: Analytical and Numerical Studies of Shocks and Rarefaction Waves, Ph.D. disseration, Dept. of Mathematics, Technical University of Denmark, Lyngby, Denmark, 2009.

[50] C. M. Fox and L. O. Ramig, "Vocal sound pressure level and self-perception of speech and voice in men and women with idiopathic Parkinson disease," *American Journal of Speech-Language Pathology,* vol. 6, no. 2, p. 85–94, 1997.

[51] Wikipedia, "Diffraction," [Online]. Available: http://en.wikipedia.org/wiki/Diffraction. [Accessed 20 March 2013].

[52] Y. Amirat, V. Choqueuse, M. Benbouzid and S. Turri, "Hilbert Transform-Based Bearing Failure Detection in DFIG-Based Wind Turbines," *International Review of Electrical Engineering,* vol. 6, no. 3, pp. 1249-1256, 2011.

[53] B. Fornberg, A Practical Guide to Pseudospectral Methods, New York: Cambridge University Press, 1998.

[54] J. B. Schneider, "Understanding the Finite-Difference Time-Domain Method," 2010. [Online]. Available: www.eecs.wsu.edu/~schneidj/ufdtd. [Accessed 14 11 2011].

[55] T. K. Katsibas and C. S. Antonopoulos, "Development and Evaluation of a Complete 3D FDTD Computational Algorithm for the Numerical Approximation of Guided Acoustic Wave Propagation in Lossy Media," *Acta Acustica united with Acustica,* vol. 93, no. 6, pp. 861-869, 2007.

[56] C. W. Manry and S. L. Broschat, "FDTD simulations for ultrasound propagation in a 2-D breast model," *Ultrason Imaging,* vol. 18, no. 1, pp. 25-34, 1996.

[57] G. Mur, "Absorbing Boundary Conditions for the Finite-Difference Approximation of the Time-Domain Electromagnetic-Field Equations," *IEEE Trans. Electromagn. Compat,* Vols. EMC-23, no. 4, p. 377–382, 1981.

[58] J.-P. Bérenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys,* vol. 114, no. 2, p. 185–200, 1994.

[59] G. Zheng, A. A. Kishk, A. W. Glisson and A. B. Yakovlev, "Implementation of Mur's absorbing boundaries with periodic structures to speed up the design process using fiite-difference time-domain method," *Progress In Electromagnetics Research,* vol. 58, pp. 101-114, 2006.

[60] L. Ling, L. Ronglin, X. Suming and N. Guangzheng, "A new generalized perfectly matched layer for terminating 3D lossy media," *IEEE Trans. Magn.,* vol. 38, no. 2, pp. 713-716, 2002.

[61] P. G. Petropoulos, "Phase error control for FD-TD methods of second and fourth order accuracy," *IEEE Trans. Antennas Propag.,* vol. 52, no. 6, pp. 859-862, 1994.

[62] A. Taflove, Ed., Advances in Computational Electrodynamics: The Finite-Difference Time-Domain Method, Norwood, MA: Artech House, 1998.

[63] A. J. Hill and M. O. J. Hawksford, "Visualization and analysis tools for low-frequency propagation in a generalized 3D acoustic space," *Journal of the Audio Engineering Society,* vol. 59, no. 5, pp. 321-337, 2011.

[64] K. F. Warnick, "An intuitive error analysis for FDTD and comparison to MoM," *IEEE Antennas Propag. Mag,* vol. 47, no. 6, pp. 111-115, 2005.

[65] J. W. Massey, Creating AustinMan: An Electromagnetic Voxel Model of the Visible Human, B.S. thesis, Dept. ECE, University of Texas at Austin, Austin, TX, 2011.

[66] V. Spitzer, M. J. Ackerman, A. L. Scherzinger and D. Whitlock, "The visible human male: a technical report," *Journal of the American Medical Informatics Association : JAMIA,* vol. 3, no. 2, pp. 118-30, 1996.

[67] G. T. Silva, S. Chen and L. P. Viana, "Parametric amplification of the dynamic radiation force of acoustic waves in fluids," *Phys Rev Lett.,* vol. 96, no. 23, p. 234301, 2006.

[68] S. Bharat and T. Varghesea, "Radiofrequency electrode vibration-induced shear wave imaging for tissue modulus estimation: A simulation study," *J Acoust Soc Am.,* vol. 128, no. 4, p. 1582–1585, 2010.

[69] F. G. Mitri, J. F. Greenleaf and M. Fatemi, "Chirp imaging vibro-acoustography for removing the ultrasound standing wave artifact," *IEEE Trans Med Imaging,* vol. 24, no. 10, pp. 1249-55, 2005.

[70] M. Fatemi and J. F. Greenleaf, "Vibro-acoustography: An imaging modality based on ultrasound-stimulated acoustic emission," *Proc. Natl. Acad. Sci. USA,* vol. 96, p. 6603–6608, 1999.

[71] P. Ji, J. Yang and W.-S. Gan, "The investigation of localized sound generation using two ultrasound beams," *IEEE Trans Ultrason Ferroelectr Freq Control,* vol. 56, no. 6, pp. 1282-7, 2009.

[72] T. Hueber, E.-L. Benaroya, G. Chollet, B. Denby, G. Dreyfus and M. Stone, "Development of a silent speech interface driven by ultrasound and optical images of the tongue and lips," *Journal Speech Communication,* vol. 52, no. 4, pp. 288-300, 2010.

[73] C. Q. Howard, C. H. Hansen and A. C. Zander, "A Review of Current Ultrasound Exposure Limits," *J. Occupational Health and Safety of Australia and New Zealand,* vol. 21, no. 3, pp. 253-257, 2005.

[74] M. Nagel and J. H. Nagel, "Bioeffects and safety of low frequency ultrasound," in *BMES/EMBS Conference*, Atlanta, GA, 1999.

[75] F. Ahmadi, I. V. McLoughlin, S. Chauhan and G. ter-Haar, "Bio-effects and safety of low-intensity, low-frequency ultrasonic exposure," *Prog Biophys Mol Biol.,* vol. 108, no. 3, pp. 119-38, 2012.

[76] G. D. Ludwig, "The Velocity of Sound through Tissues and the Acoustic Impedance of Tissues," *Journal of the Acoustical Society of America,* vol. 22, no. 6, pp. 862-6, 1950.

[77] H. Azhari, Basics of Biomedical Ultrasound for Engineers, John Wiley & Sons, 2010.

[78] F. A. Duck, Physical properties of tissue: a comprehensive reference book, London: Academic Press, 1990.

[79] D. Andreuccetti, R. Fossi and C. Petrucci, "Dielectric Properties of Body Tissues," IFAC-CNR, [Online]. Available: http://niremf.ifac.cnr.it/tissprop/.

[80] P. A. Hasgall, E. Neufeld, M. C. Gosselin, A. Klingenböck and N. Kuster, "IT'IS Database for thermal and electromagnetic parameters of biological tissues," Version 2.2. IT'IS Foundation, 11 July 2012. [Online]. Available: www.itis.ethz.ch/database.

[81] National Physical Laboratory, "Welcome to Kaye and Laby Online," 2012. [Online]. Available: http://www.kayelaby.npl.co.uk/.

[82] W. R. Hendee and E. R. Ritenour, Eds., Medical Imaging Physics, 4th ed., New York: Wiley-Liss, 2002.

[83] J. K. Zuur, S. H. Muller, F. H. C. de Jongh, M. J. van der Horst, M. Shehata, J. van Leeuwen, M. Sinaasappel and F. J. M. Hilgers, "A newly developed tool for intra-tracheal temperature and humidity assessment in laryngectomized individuals: the Airway Climate Explorer (ACE).," *Med Biol Eng Comput.,* vol. 45, no. 8, pp. 737-45, 2007.

[84] Saint-Gobain, "Norprene," 30 April 2009. [Online]. Available: http://www.foams.saint-gobain.eu/uploadedFiles/SGfoamseu/Documents/Norprene_EN.pdf. [Accessed 20 August 2013].

[85] D. King, Development of Renal Phantoms for the Evaluation of Current and Emerging Ultrasound Technology, Ph.D. dissertation, School of Physics, Dublin Institute of Technology, Dublin, 2009.

[86] P. N. Wells and H.-D. Liang, "Medical ultrasound: imaging of soft tissue strain and elasticity," *J R Soc Interface,* vol. 8, no. 64, pp. 1521-49, 2011.

[87] L. Luo, J. Molnar, H. Ding, X. Lv and G. Spengler, "Ultrasound absorption and entropy production in biological tissue: a novel approach to anticancer therapy," *Diagnostic Pathology,* vol. 1, no. 1, p. 35, 2006.

[88] C. M. Sehgal and J. F. Greenleaf, "Ultrasonic absorption and dispersion in biological media: A postulated model," *J. Acoust. Soc. Am.,* vol. 72, no. 6, pp. 1711-1718, 1982.

[89] A. Sawicki, "Adam Sawicki," [Online]. Available: http://www.asawicki.info/. [Accessed 23 November 2011].

[90] B. E. Rogowitz and L. A. Treinish, "How not to lie with visualization," *Comput. Phys.,* vol. 10, no. 3, pp. 268-273, 1996.

[91] O. B. Vikholt, Stående-bølge-problemer i opptaksstudioer kan minskes vha GPUakselerert simuleringsprogram, M.S. thesis, Norges teknisk-naturvitenskapelige universitet, Trondheim, Norway, 2012.

[92] R. F. Coleman, J. H. Mabis and J. K. Hinson, "Fundamental frequency-sound pressure level profiles of adult male and female voices," *J Speech Hear Res.,* vol. 20, no. 2, pp. 197-204, 1977.

[93] A. Schindler, A. Canale, A. L. Cavalot, R. Albera, P. Capaccio, F. Ottaviani and O. Schindler, "Intensity and fundamental frequency control in tracheoesophageal voice," *Acta Otorhinolaryngol Ital.,* vol. 25, no. 4, p. 240–244, 2005.

[94] J. D. Markel and A. H. Gray, Jr., Linear Prediction of Speech, New York: Springer-Verlag, 1976.

[95] M. S. Weiss, G. H. Yeni-Komshian and J. M. Heinz, "Acoustical and perceptual characteristics of speech produced with an electronic artificial larynx," *J Acoust Soc Am,* vol. 65, no. 5, pp. 1298-1308, 1979.

[96] R. A. Sharipov, "Quick Introduction to Tensor Analysis," 2004. [Online]. Available: http://samizdat.mines.edu/tensors/ShR6b.pdf. [Accessed 6 August 2013].

[97] P. A. Kelly, "Solid Mechanics Lecture Notes," 2012. [Online]. Available: http://homepages.engineering.auckland.ac.nz/~pkel015/SolidMechanicsBooks/. [Accessed 30 July 2013].

[98] H. T. Banks, S. Hu and Z. R. Kenz, "A brief review of elasticity and viscoelasticity for solids," *Adv. Appl. Math. Mech.,* vol. 3, no. 1, pp. 1-51, 2011.

[99] O. C. Zienkiewicz and R. L. Taylor, Finite Element Method, Volume 2: Solid Mechanics, Fifth Edition, Woburn, MA: Butterworth-Heinemann, 2000.

[100] A. G. Holzapfel, Nonlinear Solid Mechanics: A Continuum Approach for Engineering, Chichester: John Wiley & Sons, Ltd., 2000.

[101] H. T. Banks, S. Hu and Z. R. Kenz, "A brief review of elasticity and viscoelasticity for solids," *Adv. Appl. Math. Mech.,* vol. 3, no. 1, pp. 1-51, 2011.

[102] O. Trabelsi, A. P. del Palomar, J. L. López-Villalobos, A. Ginel and M. Doblaré, "Experimental characterization and constitutive modeling of the mechanical behavior of the human trachea," *Med Eng Phys,* vol. 32, no. 1, pp. 76-82, 2010.

[103] G. A. Holzapfel and T. C. Gasser, "A viscoelastic model for fiber-reinforced composites at finite strains: Continuum basis, computational aspects and applications," *Comput. Methods Appl. Mech. Engrg.,* vol. 190, no. 34, p. 4379–4403, 2001.

[104] M. Malvè, A. P. del Palomar, S. Chandra, J. L. López-Villalobos, A. Mena, E. A. Finol, A. Ginel and M. Doblaré, "FSI Analysis of a healthy and a stenotic human trachea under impedance-based boundary conditions," *J Biomech Eng.,* vol. 133, no. 2, pp. 021001:1-12, 2011.

[105] T. C. Gasser and G. A. Holzapfel, "A rate-independent elastoplastic constitutive model for biological fiber-reinforced composites at finite strains: continuum basis, algorithmic formulation and finite element implementation," *Computational Mechanics,* vol. 29, no. 4-5, pp. 340-360, 2002.

[106] Z. A. Taylor, M. Cheng and S. Ourselin, "High-speed nonlinear finite element analysis for surgical simulation using graphics processing units," *IEEE Trans Med Imaging,* vol. 27, no. 5, pp. 650-63, 2008.

[107] K. Miller, G. Joldes, D. Lance and A. Wittek, "Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation," *Commun. Numer. Meth. En.,* vol. 23, no. 2, pp. 121-134, 2007.

[108] Z. A. Taylor, O. Comas, M. Cheng, J. Passenger, D. J. Hawkes, D. Atkinson and S. Ourselin, "On modelling of anisotropic viscoelasticity for soft tissue simulation: numerical solution and GPU execution," *Med Image Anal.,* vol. 13, no. 2, pp. 234-44, 2009.

[109] O. Comas, Z. A. Taylor, J. Allard, S. Ourselin, S. Cotin and J. Passenger, "Efficient nonlinear FEM for soft tissue modelling and its GPU implementation within the open source framework SOFA," in *ISBMS*, London, UK, 2008.

[110] G. Dhondt, The Finite Element Method for Three-Dimensional Thermomechanical Applications, Chichester: John Wiley & Sons, Ltd., 2004.

[111] B. Cox, "Acoustics for Ultrasound Imaging," January 2012. [Online]. Available: http://www.ucl.ac.uk/medphys/staff/people/bcox/. [Accessed 10 December 2012].

[112] M. L. Palmeri and K. R. Nightingale, "Acoustic radiation force-based elasticity imaging methods," *Interface Focus,* vol. 1, no. 4, pp. 553-64, 2011.

[113] F. E. Borgnis, "Acoustic radiation pressure of plane compressional waves," *Rev. Mod. Phys.,* vol. 25, p. 653–664, 1953.

[114] J. Grandy, "Efficient computation of volume of hexahedral cells," Lawrence Livermore National Laboratory, Informal report UCRL-ID-128886, 1997.

[115] Harvard University, "BioNumbers - The Database of Useful Biological Numbers," [Online]. Available: http://bionumbers.hms.harvard.edu/. [Accessed 04 August 2013].

[116] Z. Teng, O. Trabelsi, I. Ochoa, J. He, J. H. Gillard and M. Doblare, "Anisotropic material behaviours of soft tissues in human trachea: an experimental study," *J Biomech,* vol. 45, no. 9, pp. 1717-23, 2012.

[117] R. D. Kamm, "Airway wall mechanics," *Annu Rev Biomed Eng.,* vol. 1, pp. 47-72, 1999.

[118] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in *ACM SenSys*, San Diego, CA, 2005.

[119] M. O. Culjat, D. Goldenberg, P. Tewari and R. S. Singh, "A review of tissue substitutes for ultrasound imaging," *Ultrasound Med Biol.,* vol. 36, no. 6, pp. 861-73, 2010.

[120] ImageCraft, "ImageCraft Embedded Systems C Development Tools," [Online]. Available: https://www.imagecraft.com/. [Accessed 2 March 2013].

[121] "Atmel 8-bit AVR Instruction Set," June 2010. [Online]. Available: http://www.atmel.com/images/doc0856.pdf. [Accessed 9 January 2014].

[122] "ANSI S1.4-1983. American national standard specification for sound level meters," American National Standards Institute, Washington, D.C., USA, 1992.

[123] J. D. Foley, A. V. Dam, S. K. Feiner and J. F. Hughes, Computer Graphics: Principles and Practice, 2nd edn. in C, Boston: Addison-Wesley, 1996.

[124] American Speech-Language-Hearing Association, "Noise," [Online]. Available: http://www.asha.org/public/hearing/noise/. [Accessed 5 February 2014].

[125] J. Mariani, Ed., Language and Speech Processing, London, England: ISTE, Ltd., 2009.

[126] W. J. Hardcastle, J. Laver and F. E. Gibbon, Eds., The Handbook of Phonetic Sciences, 2nd ed., Oxford, England: Blackwell Publishing, Ltd., 2010.

[127] A. E. Rosenberg, "Effect of glottal pulse shape on the quality of natural vowel," *J Acoust Soc Am.,* vol. 49, no. 2, p. 583–590, 1971.

[128] J. Walker and P. Murphy, A review of glottal waveform analysis. In: Progress in Nonlinear Speech Processing, Y. Stylianou, M. Faundez-Zanuy and A. Eposito, Eds., Berlin: Springer Verlag, 2007, p. 1–21.

[129] Y. Qi and B. Weinberg, "Low-frequency energy deficit in electrolaryngeal speech," *J Speech Hear Res.,* vol. 34, no. 6, pp. 1250-6, 1991.

[130] N. H. Fletcher, "Adiabatic Assumption for Wave Propagation," *American Journal of Physics,* vol. 42, no. 6, pp. 487-9, 1974.

[131] I. R. Titze and B. H. Story, "Acoustic interactions of the voice source with the lower vocal tract," *J Acoust Soc Am,* vol. 101, no. 4, pp. 2234-43, 1997.

[132] J.-A. Bachorowski, "Vocal expression and perception of emotion," *Curr Dir Psychol Sci,* vol. 8, no. 2, pp. 53-57, 1999.

[133] M. J. Crocker, Ed., Handbook of Acoustics, John Wiley and Sons, Inc., 1998.