THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF ELECTRICAL ENGINEERING

MODELING AN ACTIVE NOISE CANCELLATION SYSTEM IN OPENMODELICA AND
MATLAB

NIKHIL BHAT
SPRING 2017

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Electrical Engineering
with honors in Electrical Engineering

Reviewed and approved* by the following:

Stephen C. Thompson
Professor of Acoustics
Thesis Supervisor

Julio Urbina
Associate Professor of Electrical Engineering
Honors Adviser

* Signatures are on file in the Schreyer Honors College.
ABSTRACT

Noise cancellation is a desirable effect that is achieved by adding a phase reversed copy of undesired noise to a signal that contains this noise, which cancels it through destructive interference. Noise cancellation has proven to be popular in both consumer headphones as well as industrial and office work spaces. Both styles of noise cancellation operate in a similar way: a microphone senses the noise source from a very small distance away and a speaker at some distance away replays the sound with magnitude and phase adjusted so that the sum of these sound pressure levels cancels or attenuates the overall noise.

This study aims to develop a model for this noise cancellation effect in OpenModelica, an open-source modeling software for electrical and mechanical components based on the Modelica language. This model will provide a more granular look at the dynamics of this phenomenon in two dimensions and will also be more accessible to the public. The in-depth look will be provided by the expanded nature of the speaker and microphone models, which display individual circuit components with easily customizable parameters in lieu of the equivalent circuits which dominate in other models. The model will be more accessible because it is being created in a free, open-source software (OpenModelica) as opposed to a more popular Modelica environment, such as Dymola, which can cost thousands of dollars. The result of this study will provide speaker and microphone designers with the tools they need to understand current methodologies and subsequently spur innovation.
# TABLE OF CONTENTS

LIST OF FIGURES ..............................................................................................................iv

LIST OF EQUATIONS ........................................................................................................v

ACKNOWLEDGEMENTS ..................................................................................................vi

Chapter 1 Introduction and Background ........................................................................ 1
  1-1: Active Noise Cancellation ....................................................................................... 1
  1-2: Modelica .................................................................................................................. 3
  1-3: OpenModelica ......................................................................................................... 3
  1-4: Thiele-Small Parameters ......................................................................................... 4

Chapter 2 Methods ............................................................................................................. 5
  2-1: Loudspeaker Model ............................................................................................... 5
  2-2: Microphone Model ............................................................................................... 14
  2-3: Noise Cancellation ............................................................................................... 20

Chapter 3 Results ............................................................................................................. 22
  3-1: Loudspeaker Model ............................................................................................... 22
  3-2: Microphone Model ............................................................................................... 26

Chapter 4 Discussion ....................................................................................................... 28

Chapter 5 Conclusions .................................................................................................... 29

Appendix A MATLAB Code .......................................................................................... 31
  noisecancel4.m ............................................................................................................. 31
  whiteNoiseGenerator.m .............................................................................................. 33

BIBLIOGRAPHY ............................................................................................................. 35

ACADEMIC VITA ............................................................................................................ 36
LIST OF FIGURES

Figure 1-1: Destructive Interference through Phase Inversion .................................. 2
Figure 2-1: Full Loudspeaker Model ............................................................................. 6
Figure 2-2: Speaker Cutaway Visualization [6]............................................................... 6
Figure 2-3: Electrical Domain of the Loudspeaker Model ........................................... 7
Figure 2-4: The Speaker Coil Component in the Loudspeaker Model ......................... 8
Figure 2-5: OpenModelica Code for Speaker Coil Component ................................... 10
Figure 2-6: Mechanical Domain of Loudspeaker Model ............................................. 11
Figure 2-7: Acoustical Domain of the Loudspeaker Model ......................................... 12
Figure 2-8: Full Microphone Model ............................................................................. 15
Figure 2-9: Acoustical Domain of the Microphone Model .......................................... 16
Figure 2-10: Mechanical Domain of the Microphone Model ....................................... 17
Figure 2-11: Electrical Domain of Microphone Model ............................................... 18
Figure 2-12: Charging Portion of the Electrical Domain of Microphone Model .......... 19
Figure 3-1: Input Impedance Curve from Dayton Audio Speaker Specification [3] ... 22
Figure 3-2: Input Impedance Curve for Loudspeaker Model from 20-5000Hz .......... 23
Figure 3-3: Output for Voltage from signalCurrent Component in Loudspeaker Model Full View (left): chirp from 20-500Hz | Zoomed View (right): chirp from 40-100 Hz 24
Figure 3-4: Input Impedance (Output Voltage of signalCurrent Component) from 500-5000Hz .................................................................................................................. 25
Figure 3-5: Output on Voltage Sensor in Microphone Model ...................................... 26
Figure 3-6: Zoomed-in View of Steady State Portion .................................................. 27
LIST OF EQUATIONS

Equation 1: Force on a Wire in a Magnetic Field ......................................................... 9
Equation 2: Voltage Generated in a Moving Wire ......................................................... 9
Equation 3: Coulomb's Law .......................................................................................... 18
Equation 4: Microphone Sensor Differential Voltage Equation ..................................... 19
Equation 5: Ohm's Law with Impedance ...................................................................... 23
Equation 6: Computations for Resonant Frequency ...................................................... 24
Equation 7: Inductive Impedance ................................................................................. 25
ACKNOWLEDGEMENTS

There are several people that I would like to thank for making this thesis possible. I would first like to thank Professor Stephen C. Thompson for his careful direction and invaluable technical mentorship throughout the research process. I would also like to thank Dr. Julio V. Urbina for his professional guidance as an honors adviser. Lastly, I would like to extend my sincerest thanks to my family, friends, and professors for their continued support of my academic and professional endeavors.
Chapter 1
Introduction and Background

1-1: Active Noise Cancellation

Active noise cancellation is a method of noise reduction where undesirable sounds are attenuated by adding a copy of the sound that is designed to cancel the original. The reduction is due to destructive interference, which occurs when adding two waves results in values that are less than the values in either wave alone, or is zero. While adding sound in order to reduce sound may seem counterintuitive, this method of noise cancellation is extremely effective. The cancellation works through a principle called phase inversion. The second wave, which is being used for cancellation, is the same wave, but shifted by $\pi$. A shift of $\pi$ aligns the second wave in such a way that the peaks of one wave line up with the valleys of the other, and vice versa, so that the sum of these waves cancels to zero or some other negligible value. Since the waveforms have to be lined up precisely in order to cancel, active noise cancellation works best with signals that exhibit some level of periodicity, since these are more predictable than patterns of random noise. This concept is illustrated in Figure 1-1.
Active noise cancellation has a broad spectrum of potential uses. Most commonly, it is used in consumer headphones to allow for a better auditory experience. Active noise cancellation is also used in factories, office spaces, and in planes and other transport vehicles to reduce ambient noise. As an example, consider a noise-cancelling headset. A headset of this type operates by placing a microphone in the ear cup of the headphones that “listens” to external sounds. A circuit within takes this noise and makes a wave of equal frequency and amplitude that is 180 degrees out of phase. This cancelling noise is then played through the headphones at the same time as the regular noise, thus cancelling the unwanted noise and preserving the desired audio signal. This project will deal with modeling a speaker and a microphone for use in a similar application in OpenModelica, and subsequently creating a control system between the two to achieve maximum noise reduction.
It is important to note that the noise cancellation that is being used in this paper is a very simplified version that operates exclusively in 2 dimensions. When creating noise cancellation in a large space, it is essential to identify ways to achieve similar levels of noise reduction at different points in a three-dimensional space. In the interest of reducing computational time and to better fit the expositional nature of this project, noise cancellation is only being performed in one direction (the line perpendicular to the face plane of the speaker).

1-2: Modelica

Modelica is “an object-oriented, equation-based programming language, oriented toward computational applications with high complexity requiring high performance” [1]. In simpler language, Modelica is based on equations rather than variable assignments. Therefore, a system must be fully defined (an equal number of equations and unknowns) in order for the program to be run. Modelica allows the user to model a system in multiple domains, which makes it ideal for this project since the microphone and loudspeaker models contain electrical, mechanical, and acoustical components. In this way, Modelica significantly improves upon previous modeling languages such as SPICE, which were restricted to a single domain.

1-3: OpenModelica

OpenModelica is an open-source environment for Modelica. OpenModelica was created by and is regularly updated by the Open Source Modelica Consortium [2]. The impetus for using OpenModelica over industry-standard Modelica environments like Dymola is primarily the cost. OpenModelica is free and has similar functionality to Dymola, whereas Dymola can cost up to
$5,900 in its standard version [3]. In addition, using an open-source program provides more
versatility in component development. The Acoustical Component library developed by Stephen
C. Thompson proves invaluable in modeling the acoustical components of loudspeakers and
microphones in conjunction with the electrical and mechanical workings of these devices.

1-4: Thiele-Small Parameters

Thiele-Small (T/S) parameters are a set of measurable electromechanical performance
properties that are used as a basis for loudspeaker driver design. The Thiele-Small parameters
can be used to calculate an even more useful set of analog circuit component values that allow
the performance of the speaker driver to be calculated. In most cases, driver designers will
provide the customer a list of both the Thiele-Small parameters and the analog circuit modeling
values, and the customer will choose the driver whose performance best fits the specifications of
the speaker system that they are designing. The required analog circuit parameters include the
input impedance of the speaker driver \((Z)\), the electrical resistance of the voice coil \((R_e)\), the
inductance of the voice coil \((L_e)\), the resonance frequency of the driver \((F_s)\), the “motor strength”
of the speaker \((Bl)\), the mechanical compliance, resistance and mass of the voice coil
\((C_{ms}, R_{ms}, M_{ms})\), and other parameters that fully encompass the electrical, mechanical, and
acoustical requirements of the speaker driver. The parameters for the loudspeaker model here
were taken from the ‘Dayton Audio ND91-8 3-1/2” Aluminum Cone Full-Range Driver 8 Ohm’
speaker specification sheet on the Dayton Audio website [4].
Chapter 2

Methods

As stated above, Modelica provides multidomain modeling capabilities. Each domain (electrical, mechanical, acoustical) is modeled using two types of variables: potential variables and flow variables [1]. Modelica uses Kirchhoff’s laws to check that systems are fully defined before running calculations. Potential variables follow Kirchhoff’s voltage law, which states that the sum of the voltage sources in a closed loop is equal to the sum of the voltage drops in that loop. Flow variables follow Kirchhoff’s current law, which states that at any node, the sum of currents flowing into a node is equal to the sum of the currents flowing out of that node [5]. In the electrical domain, these variables are voltage (potential) and current (flow). In the mechanical domain, these variables are force (potential) and velocity (flow). In the acoustical domain, these variables are pressure (potential) and volume velocity (flow). Understanding the similarities between the three domains elucidates how components are able to convert between physical quantities in different domains.

2-1: Loudspeaker Model

The full loudspeaker model is shown in Figure 2-1. The following discussion will explain the fundamentals of loudspeaker operation, and then split the model into the electrical, mechanical, and acoustical domains and provide explanations for each of these.
The general principles of loudspeaker operation rely on the concepts of electromagnetism. The visible part of the speaker is the speaker cone, or diaphragm, which is pictured at the top of Figure 2-2.
The diaphragm is a thin, cone-shaped piece of material that generates the audible vibrations of the speaker. Behind the diaphragm, a coil of wire is wrapped around a cylindrical piece of metal called the former, which is attached to the diaphragm. An audio signal is sent through the wire as an electrical current, which induces a magnetic field perpendicular to the velocity of the current. A permanent magnet is located under the voice coil, which is used to direct this magnetic field through the voice coil. A force is then induced perpendicular to both the current velocity and the magnetic field, which moves the voice coil forward along its fixed axis. As the electrical signal goes from positive to negative, the direction of the force reverses in response to the reversal of the magnetic field direction, which creates the visible push and pull in the speaker cone. As the voice coil moves, the diaphragm moves (since it is attached to the former), thus creating the audible sound [7].

The electrical system in Figure 2-3 is the first domain in the loudspeaker model, which represents the input audio signal.

For testing purposes, a chirp signal (a signal that changes frequency as a function of time) was generated using a component from Stephen Thompson’s Modelica Acoustics Library,
intended to simulate an audio signal and its varying frequencies over time. This chirp signal, which creates a sequence of number values (known as “Real” values in OpenModelica) to represent the time-varying sinusoid, had to be tied to electrical quantities within the Modelica language to have physical significance as an electrical current. A “signalCurrent” component from OpenModelica was used to perform this conversion. An inductor is placed between the voltage source and the “speakerCoil” component to represent the inductance of the voice coil ($L_e$) in mH. The inductance value is normally measured and reported at 1000 Hz.

The next component is the “speakerCoil” component in Figure 2-4, which was created expressly for this model.

![Figure 2-4: The Speaker Coil Component in the Loudspeaker Model](image)

The solenoid with its moving charges (current) and the permanent magnet together comprise a component of the speaker known as the “coil” of the speaker. At a high level, the coil transforms electrical energy (from the audio signal) to mechanical energy (the movement of the piston). As a result, it is a transformation component, later shown to be a type of gyrator. A gyrator is typically an electrical component that converts voltage to current and vice versa. In general, a gyrator can convert any set of units to any other set of units based on some constant ratio between the two quantities, often denoted by “n” in the equation. In order to convert
between the electrical domain and the mechanical domain, a component was necessary to convert voltage to current’s mechanical analogue (velocity) and to convert current to voltage’s mechanical analogue (force). A special component had to be created that simulated this transformation. The component has a positive and negative electrical port on the input side, and a single mechanical flange on the output side. The electrical ports and the mechanical flange are related mathematically through Equation 1 and Equation 2.

\[ F = BlI \]

Equation 1: Force on a Wire in a Magnetic Field

Current is being converted to force using Equation 1, where \( F \) is force, \( Bl \) is a constant generally used by speaker manufacturers to indicate the motor strength of the speaker (how strong of a current is required to push the speaker cone forward), and \( I \) is the current from the audio signal.

\[ V = Bl\nu \]

Equation 2: Voltage Generated in a Moving Wire

Velocity is being converted to voltage in Equation 2, where \( V \) is voltage, \( Bl \) is the same constant from above, and \( \nu \) is velocity. These equations in conjunction describe the transformations made from the electrical domain to the mechanical domain. Current is proportional to force, and voltage is proportional to velocity. In OpenModelica, several other equations were needed to fully define the system at hand.
The in-detail description of this code block is designed to provide the reader with an understanding of the syntax and flow of Modelica code that can be extrapolated to other blocks of code that will be presented in this paper. Descriptions of subsequent blocks of code will not have this level of granularity.

The code begins by importing the Modelica SIunits library so that variables that represent Voltage, Force, Current, Position, and Resistance are defined as physical quantities and treated as such, rather than just numbers. The first two equations, \( V = \text{pin}_p.v - \text{pin}_n.v \) and \( 0 = \text{pin}_p.i + \text{pin}_n.i \), define the relationship between the positive and negative pins of the electrical side of the “speakerCoil” component. The first equation states that the voltage across the component is the voltage difference between the positive pin and the negative pin. The second equation shows that the positive and negative currents sum to 0, indicating that all current that enters the positive pin exits the negative pin, as specified by Kirchhoff’s Current Law [5]. \( I = \text{pin}_p.i \) states that the current is being defined as the current entering the positive pin. The next two equations simply
redefine an internal parameter of the flange so that “flange_a” does not have to be re-written every time that the force and position of the speaker coil flange is referenced. The next two equations are the equations described above for the force and the voltage. The voltage across the component is the sum of the current through the component multiplied by the impedance of the component (which is a result of the inherent impedance of the wire used for the coil) and $Bl \ast \text{der}(s)$, which is the speaker’s $Bl$ property multiplied by the derivative of the position of the flange (which is the flange velocity). The final equation is for the force, which is found by $BlI$ and is described above.

![Diagram](image)

**Figure 2-6: Mechanical Domain of Loudspeaker Model**

Since the electrical signal is transformed into mechanical motion, the mechanical section of the model must be explored. The first component is a mass, which is meant to symbolize the voice coil, which functions as a type of piston inside the speaker. The next component is a spring-mass damper, which represents the spring force and stiffness of the cone that is being pulled back and forth by the coil force. The “fixed” component represents a position of zero velocity, analogous to the electrical ground.
The acoustical domain of the model uses pressure (force divided by area) and volume velocity (particle velocity multiplied by area) to model the performance. The acoustical section consists of an acoustic area transformer, an acoustic compliance, an acoustic ground, and a baffled piston. The acoustic area transformer represents the area of the speaker cone, or how much area the vibrations generated by the mechanical section have to distribute through in order to radiate outward from the speaker cone. The area transformer converts the mechanical force and velocity at a mechanical port to the pressure and volume velocity at its acoustical port. The acoustic compliance is the amount of pressure required to compress a certain volume of air by a differential amount. In this case, it represents the airflow in and around the speaker cone. When the speaker cone moves inward, it creates a momentary vacuum that is quickly filled by the air that is surrounding the face of the speaker. When the speaker cone needs to move outward again, it requires a certain amount of force to push the air in front of it forward and outward and radiate the sound into the space. The amount of force, or pressure, that is required to compress the air in front of the cone as it moves forward is the acoustic compliance. The back of the enclosure is analogous to a spring/capacitor in this case, because it has a certain amount of acoustical “stiffness”. Without an enclosure for the speaker, the sound would be radiated out into the far-
field infinitely in both the positive and negative directions. When viewed from a distance, these positive and negative sounds would be mostly cancelled. The back of the enclosure allows the negative sound pressure waves to be contained, thus allowing the positive sound pressure waves to radiate out into the far-field without cancellation. The acoustic ground is analogous to the electrical ground. The baffled piston represents the radiation impedance, which is a ratio of the force exerted on a medium to the velocity of the object that is applying that force. This component is meant to simulate the actual sound waves that are being generated, which are more accurately quantified as the far-field sound pressure at some distance from the cone.

There were a few important design decisions made during the model creation process that affect the way that users interact with the model and how certain parameters must be defined in order for proper operation. The first design decision was where to place the speaker coil’s impedance. The coil has its own impedance due to the inherent resistance of the wire that is used, and this impedance could either be within the component or before the component. If the user wished to be able to measure the amount of heat generated by the coil as it converts electrical to mechanical energy, the impedance would be placed before the component. Since this is not an essential parameter for the scope of this project, the resistance was chosen to be a parameter of the electromechanical gyrator (“speakerCoil”) component. This design decision means that the “speakerCoil” component actually represents the whole coil: taking in the current and accounting for the wire impedance, using the coil as a solenoid, and generating a force and velocity through the movement of the mechanical flange that powers the mechanical section of this loudspeaker. By making the electromechanical interface a fully self-contained component, the analysis of the circuit is simplified.
The second important design decision was choosing to include two separate parameters for $B$ (magnetic flux density) and $l$ (length) instead of one parameter for $Bl$. Realistically, speaker manufacturers will only provide the user a $Bl$ value because they do not want to reveal too much about the magnetic design that makes their speakers unique. In the case of this model, either $B$ or $l$ could be set to 1 and the other value could be set to the value of $Bl$ and the same results would be achieved. However, it is important to note that these parameters are usually together. The speaker manufacturers will provide the individual $B$ and $l$ values to the speaker component manufacturers, so this granularity in the model is to satisfy both segments of potential users.

2-2: Microphone Model

The microphone model in Figure 2-8 was created and tested entirely by Stephen Thompson. The explanation of this model is provided here for the reader’s convenience. The author of this paper does not take credit for this section of the work.
At a high level, the microphone function is the inverse of the loudspeaker function. A loudspeaker takes an electrical signal at the input, translates it into mechanical vibration in the mechanical section, and subsequently generates sound pressure waves from the acoustic section. Conversely, a microphone takes in sound pressure waves from some sound, translates it to mechanical motion of a thin membrane, which then creates an electrical signal output. More specifically, a microphone is comprised of one very thin membrane of metal or polymer (called the diaphragm) in front and one rigid piece of metal in back (called the back plate) that lie parallel to one another, at a very close distance. As the sound pressure waves generated by a noise reach the thin membrane in front, mechanical vibrations are created in that piece of metal that are proportional to the incident pressure on the diaphragm. Since the two pieces of metal are so close together, there is a capacitance between them that varies as the vibrations of the thin
membrane of metal move the pieces of metal closer together and farther apart. This variable capacitance creates a variable voltage that manifests itself as the characteristic audio waveform.

Figure 2-9: Acoustical Domain of the Microphone Model

The microphone model starts in the acoustical domain, shown in Figure 2-9. A chirp signal (sinusoid with same amplitude but variable frequency over time) is used to represent the sound pressure waves that are incident on the diaphragm of the microphone. This “Chirp” component was taken from Stephen Thompson’s OpenModelica Acoustics Library. The “Chirp” component generates a series of numbers to represent the sinusoid. However, just as in the loudspeaker model above, these numbers must be converted into numbers that OpenModelica can recognize as physical quantities. In this case, the numbers are being converted to pressure waves. The acoustical domain has been split into two sections: the acoustical activity on the back of the diaphragm and the acoustical activity on the front of the diaphragm. In the section representing the back of the diaphragm, the component labeled “acoResistor” represents $R_d$, the air flow resistance between the diaphragm and the back plate. The component labeled “acoResistor1” represents $R_{leak}$, the leak resistance, which provides barometric pressure
equalization through a small hole in the back volume of the diaphragm [8]. The component labeled “acoCompliance” is the acoustical compliance of the back of the diaphragm, representing the spring force or “stiffness” of the air in the region behind the diaphragm. The baffled piston component, which is at the top right of Figure 2-9 and resembles a dining table, is the sole component in the section representing the front of the diaphragm. The microphone operates through sound pressure waves incident on this radiation impedance. The Acoustic Area Transformer ("acoAreaTransformer" in the diagram) is used to couple the acoustic domain with the mechanical domain, where the “turns ratio” of this transformer is equal to the area of the microphone’s diaphragm [8].

The mechanical section of the microphone model is shown in Figure 2-10 below. The mass represents the mounting of the thin metallic diaphragm, and the spring damper represents the assumption of a fixed mechanical compliance $C_m$ and fixed mechanical resistance $R_m$ [8].

Figure 2-10: Mechanical Domain of the Microphone Model
The electrostatic gap component implements the equations for both the mechanical force and the voltage that is represented as a variable capacitance across the gap between the thin diaphragm and the thicker piece of metal behind it. The equation for the force is given by Coulomb’s Law in Equation 3.

\[ F = \frac{q^2}{2\varepsilon_0 A} \]

**Equation 3: Coulomb's Law**

The voltage equation is described further in Equation 4 below. With these two equations in place, the electrostatic gap component is able to illustrate the transfer of energy between the electrical and mechanical domains [8].

![Figure 2-11: Electrical Domain of Microphone Model](image)

Figure 2-11 shows the electrical domain of the microphone model. The reading on the “voltageSensor” component over the resistor is the final output being measured (y). This resistor/voltage sensor area represents the output that would go into an amplifier before being used. This section starts with the voltage on the electrostatic gap of the microphone. The large
resistor value represents the extremely large input impedance necessary for the op-amp. The capacitor is a decoupling capacitor, which is used to block the large DC voltage coming from the charging section of the electrical circuit. Looking next at Figure 2-12, the ramp voltage being used for the charge creates a large transient right from the outset that eventually settles out.

![Figure 2-12: Charging Portion of the Electrical Domain of Microphone Model](image.png)

The ramp voltage is being used to pre-charge all the components, such as the capacitor, prior to model operation through a very large resistor. This is necessary because microphones operate off of small, differential distance changes that translate to small voltage changes. This voltage change is quantified by Equation 4.

\[
V = \frac{q}{C} = \frac{q}{\varepsilon_0 A} = \frac{ql}{\varepsilon_0 A} = \frac{q(l_0 + \delta l)}{\varepsilon_0 A} = \frac{ql_0}{\varepsilon_0 A} + \frac{q\delta l}{\varepsilon_0 A}
\]

Equation 4: Microphone Sensor Differential Voltage Equation

The voltage consists of a fixed term that is related to the static displacement of the diaphragm due to electrostatic attraction (the \(l_0\) term), and a differential distance term (the \(\delta l\) term).
term). If the components are not pre-charged, then the $q$ in the fixed distance term $\frac{ql_0}{\varepsilon_0 A}$ will be 0. 

Since the differential term is based off of small differentials in the fixed distance term, the differential term would also go to 0, and no voltage would be seen on the output. In order to remedy this, the components are pre-charged so the fixed distance term holds some baseline value, from which the differential term can vary slightly. This voltage equation is one of two equations that must be implemented for the electrostatic gap component, the square that contains the green capacitor symbol in Figure 2-10.

2-3: Noise Cancellation

The noise signal was generated in MATLAB and then imported into OpenModelica, which was done because OpenModelica generates discrete step noise which does not appropriately emulate physical noise (periodic signals, which were used to calibrate the shifting algorithm described below, were generated in OpenModelica with no issue). The noise was generated using the code in a file called whiteNoiseGenerator.m (See Appendix A). This noise was imported as a MAT file into OpenModelica using the combiTimeTable block (See Appendix A - Figure A-1).

The input signal and the loudspeaker output were then exported from OpenModelica as a MAT file and imported again into MATLAB using a file from the MATLAB File Exchange called modelicaImport. This program organizes the MAT file into a MATLAB structure that emulates the output organization structure of OpenModelica (i.e. organized by block and the inputs and outputs associated with each block). This allows for a more logical analogue between MATLAB and OpenModelica.
The noise cancellation was achieved by summing the original signal with a phase-inverted version of the output of the loudspeaker. Since the input signal travels through the condenser microphone and loudspeaker models prior to being output from the loudspeaker, there is a time-shift in the signal that must be accounted for. This time-shift is frequency dependent, since the wavelength varies inversely as frequency and therefore the number of samples by which the output signal must be shifted changes. The program uses a “for” loop to test various shifts at .005 second increments between 0 and 20. The signals are summed at each of these shifts, and the optimal shift is defined as the shift that produces the smallest sum between the signals. The final lines of code identify the minimum and maximum attenuation of the summed signal as compared to the input signal by performing a correlation between the two signals.
Chapter 3

Results

3-1: Loudspeaker Model

Speaker parameters for this simulation were taken from a Dayton Audio loudspeaker specification that is referenced above in the “Thiele-Small Parameters” section of the introduction of this paper. The driver specification also provided a characteristic for the desired input impedance of a speaker driver with the given parameters, shown below in Figure 3-1. These curves represent the envelopes of the peaks of the input impedance of the speaker.

![Figure 3-1: Input Impedance Curve from Dayton Audio Speaker Specification [3]](image)

The blue input impedance curve has a peak around 70 Hz, which is the speaker driver’s resonant frequency. The resonant frequency is important because the far field response of a moving coil speaker is generally flat above resonance, and therefore the speaker is generally
operated at a point above this frequency. A plot of the speaker model’s input impedance curve is shown in Figure 3-2 below.

![Input Impedance Curve for Loudspeaker Model from 20-5000Hz](image)

**Figure 3-2: Input Impedance Curve for Loudspeaker Model from 20-5000Hz**

OpenModelica plots data in the time domain and the desired output on the impedance curve is in the frequency domain, so the chirp signal is used to vary the frequency over time. Since frequency and voltage are both varying with respect to time, Figure 3-2 is essentially a plot of voltage versus frequency. The equation to get input impedance is Equation 5.

\[ Z = \frac{V}{I} \]

**Equation 5: Ohm's Law with Impedance**

Since the input current is constant and set to unity (a value of 1) for this calculation, impedance in the above equation is voltage divided by some constant. Therefore, Figure 3-2 is a plot of input impedance versus frequency (on a normal frequency scale, as opposed to the logarithmic scale in Figure 3-1).

In order to find the input impedance at a given frequency, one must find the frequency at a given time. The chirp signal is increasing in frequency at a steady rate, in a specific range, over a set amount of time, so it is essentially a line with a constant slope. The frequency at any specific time can be found by finding the time of interest as a percentage of the total time,
multiplying it by the frequency range, and adding this value to the bottom bound of the frequency range. The center of the peak is at 1.03 seconds. The chirp varies from 20-500 Hz over a period of 10 seconds.

\[
Frequency \ Range \ = \ 500 \ Hz \ - \ 20 \ Hz \ = \ 480 \ Hz
\]

\[
Fractional \ Time \ = \ \frac{1.03}{10} \ = \ .103
\]

\[
Resonant \ Frequency \ = \ .103 \ * \ 480 \ + \ 20 \ Hz \ = \ 69.44 \ Hz
\]

**Equation 6: Computations for Resonant Frequency**

According to Equation 6, the resonant frequency is 69.44 Hz, which corresponds very closely to the previous estimation of 70 Hz for the resonant frequency of the input impedance in the speaker specification.

![Figure 3-3: Output for Voltage from signalCurrent Component in Loudspeaker Model](image)

**Full View (left): chirp from 20-500Hz | Zoomed View (right): chirp from 40-100 Hz**
Figure 3-3 shows a zoomed-in view of the resonant frequency in the loudspeaker model’s input impedance curve, because the full speaker impedance characteristic did not accurately represent the location of the resonant frequency peak. Making the chirp signal sweep more slowly across this range allows the resonant frequency peak to be more defined. The input impedance curve also starts to rise continuously after 1 kHz. This can be seen more easily in Figure 3-4, a zoomed-in view of the input impedance from 500-5000Hz.

![Input Impedance Curve](image)

**Figure 3-4: Input Impedance (Output Voltage of signalCurrent Component) from 500-5000Hz**

This is because the inductive impedance of the coil is directly proportional to the frequency based on Equation 7.

\[ Z_L = j\omega L \]

**Equation 7: Inductive Impedance**

As the frequency rises, the inductive impedance quickly rises until it becomes higher than the resistive impedance of the wire. Thus, the rise of the input impedance curve after 1 kHz corresponds to the inductive impedance as it starts to increase faster than the resistive impedance. In the speaker characteristic in Figure 3-1, this rise would be proportional to \( j\omega L \) if it were not for the presence of eddy currents in the magnetic structure of the speaker. These eddy currents make the curve look more like \( \sqrt{j\omega L} \). The input impedance curve of the speaker model
does not include eddy currents, so the rise is closer to $j\omega L$ in Figure 3-4. The loudspeaker model accurately duplicates the necessary input impedance characteristic based on the provided inputs, so it is a viable model for consistent use.

3-2: Microphone Model

The peak to the left is the large transient that is caused when the ramp voltage source turns on. As the components in the electrical section slowly charge to full capacity, the transient attenuates until the system is in a steady state. The rise and fall of the transient takes approximately three seconds, which is the time delay that is built into the chirp signal before it begins transmitting the signal (the simulation runs for 13 seconds in Figure 3-5). Once the system is in steady state, the chirp signal activates and creates a system that is at approximately constant amplitude over time, except for at the resonant frequency of the system.
Figure 3-6 shows an enlarged view of the flat, steady-state section in the graph above. The amplitude is fairly constant throughout, but has a peak around 17 kHz, based on Equation 6. Thus, the resonant frequency of this condenser microphone is around 17 kHz. A major limitation with OpenModelica is that it can only plot data in the time domain, which means that specific frequency data cannot be found without a creative method of plotting and calculations, such as those above.
Chapter 4

Discussion

The microphone and speaker models are good representations of ideal microphones and
speakers based on the comparisons between the expected and measured results above. The
microphone model output fits the characteristic for a microphone that has a resonant frequency at
17kHz. The loudspeaker models output characteristic matches the input impedance curve from
the Dayton Audio specification sheet, with a peak at around 70Hz.

The noise cancellation program provided between -18 and -170dB of attenuation for
Gaussian noise. When very little is known about the periodicity of the input signal (as is the case
with noise), the expected level of attenuation is between 15 and 20dB [9], which this program
satisfies. When the signal is periodic, such as with a sinusoid of a single frequency, the noise
cancellation is close to 99.84% (e.g. 2.5V was attenuated to approximately .004V), which is
unsurprising considering that the signal is very predictable. Achieving such high levels of
attenuation in the noise was much more difficult due to the inherent frequency responses of the
microphone and speaker models. With sinusoids of single frequencies, the amplitude of the wave
will remain the same even if a certain frequency is attenuated since that frequency is being
uniformly attenuated throughout. The Gaussian noise is comprised of various frequencies and
amplitudes, so the frequency response of the speaker and microphone models will affect different
sections of the signal in a unique way. Thus, the wave shape in the output signal is slightly
altered, which does not allow the signals to cancel in a uniform fashion even if the alignment is
correct. This accounts for the discrepancy in level of attenuation between the periodic signal and
the noise.
Chapter 5

Conclusions

The results fit the specifications which were laid out at the beginning of this project. A microphone and loudspeaker model were created in OpenModelica with the electrical, mechanical and acoustical components laid out in an accessible fashion. Using passive noise cancellation, more than 18dB of attenuation could be achieved in random noise sources, and more than 99% of the noise could be attenuated in periodic signals. To improve this level of attenuation, active noise cancellation methods such as adaptive filtering must be used. Adaptive filtering involves using a feedback or feedforward loop to consistently update a transfer function between the input and output to optimize the amount of attenuation on each loop through the control system. This is a challenging optimizing problem, and one which is outside the scope of the undergraduate project detailed here.

Overall, this project has been instrumental in providing accurate and accessible models of speakers and microphones in open-source software. Speaker and microphone manufacturers, designers, and the common person alike will benefit from having models that allow for a component-by-component examination of the operation of these systems. Altering the parameters to fit various speaker and microphone specifications is quite simple with the OpenModelica platform. The option for a link between MATLAB and OpenModelica that has been documented in this paper allows for the use of two extremely powerful tools in conjunction to provide users who have a proficiency in MATLAB to take advantage of these models as well. It is the author’s hope that future users of these models can learn from the insights provided here.
and expand upon them by creating an adaptive filtering algorithm that better attenuates noise for practical applications, and that these models will continue to be used for years to come as an educational and scientific tool for speaker and microphone design.
% clear
% clc

data = modelicaImport('MData_res.mat');

sampleStart = 260000;
sampleStep = 50;
sampleEnd = 1200000;
time = data.time(sampleStart:sampleStep:sampleEnd);

% the fshift program is from:
% https://www.mathworks.com/matlabcentral/fileexchange/7886-fshift
% It allows for the non-integer time-shifting of signal.
% The peak in the beginning is b/c fshift works like CIRCHIFT, so the
% overflow at the end is attached to the beginning of the array.

speakerOut = data.loudspeaker21.y(sampleStart:sampleStep:sampleEnd);

inputShortened = data.combiTimeTable1.y(sampleStart:sampleStep:sampleEnd)';
speakerGainArray = zeros(length(speakerOut),1);
for i=1:length(speakerOut)
    speakerGainArray(i) = speakerOut(i)/inputShortened(i);
end
speakerGain = abs(median(speakerGainArray));
inputProcessed = speakerGain *
data.combiTimeTable1.y(sampleStart:sampleStep:sampleEnd )';

noiseCancelShiftBaseline =
max(abs(inputProcessed+(speakerOut)));
for noiseCancelTimeShift = 0:.005:20
    speakerOutShifted =
    fshift(speakerOut,noiseCancelTimeShift);
    noiseCancelSignalSum =
    inputProcessed+(speakerOutShifted);
    if max(abs(noiseCancelSignalSum(1000:2000))) <
    noiseCancelShiftBaseline
        noiseCancelShiftBaseline =
        max(abs(noiseCancelSignalSum(1000:2000)));
        optimalNoiseCancelTimeShift =
        noiseCancelTimeShift;
    end
end

speakerOutShifted = fshift(speakerOut,
optimalNoiseCancelTimeShift);

inputProcessed =
inputProcessed(1:length(inputProcessed)-1);
speakerOutShifted =
speakerOutShifted(1:length(speakerOutShifted)-1);
speakerGainArray =
speakerGainArray(1:length(speakerGainArray)-1);

noiseCancelSignalSum =
inputProcessed+(speakerOutShifted);

sprintf('The optimal shift is %g. The max of the sum is
%g.', optimalNoiseCancelTimeShift,
max(noiseCancelSignalSum(5000:length(noiseCancelSignalSum))))
figure(3)
subplot(2,1,1)
hold on
plot(time(1:length(noiseCancelSignalSum)),
inputProcessed(1:length(noiseCancelSignalSum)),'g');
plot(time(1:length(noiseCancelSignalSum)), -1*speakerOutShifted(1:length(noiseCancelSignalSum)), 'b');
title('Noise Cancel Input & Output')
subplot(2,1,2)
plot(time(1:length(noiseCancelSignalSum)),
inputProcessed(1:length(noiseCancelSignalSum)),'g');
hold on
plot(time(1:length(noiseCancelSignalSum)),
noiseCancelSignalSum(1:length(noiseCancelSignalSum)), 'r');
title('Noise Cancel Signal Sum')

%Code from:
https://www.mathworks.com/matlabcentral/newsreader/view_thread/346767
%calculates decibel reduction
[a_corr, lag_a] = xcorr(inputProcessed,
inputProcessed);
[x_corr, lag_x] = xcorr(noiseCancelSignalSum,
inputProcessed);

dB = 20*log10(abs(x_corr) / max(abs(a_corr))));

minAttenuation = max(dB)
maxAttenuation = min(dB)

whiteNoiseGenerator.m

time = linspace(0,4.9923,26001)';
y = randn(26001,1);
whiteNoise = [time y];
% xlswrite('inputNoise.xlsx',whiteNoise);
save('tables.mat', 'whiteNoise');

Figure A-1: combiTimeTable Parameter Values
BIBLIOGRAPHY


ACADEMIC VITA

Nikhil Bhat

Education

The Pennsylvania State University, 2017
Schreyer Honors College
B.S. Electrical Engineering
Minors in Mathematics and Music Technology

Work Experience

Boeing Commercial Airplanes: Avionics – Flight Displays Intern
  o Increased avionics test efficiency by 90% by automating and optimizing the procedures with a variant of C
  o Developed system-level software requirements for the 777X
  o Modernized testing by developing models in MATLAB/Simulink

Siemens: Engineering Leadership Development Intern
  o Decreased testing time by 95% by designing a streamlined switch/breaker testing center
  o Interfaced with key engineers to facilitate iterative software improvement for next generation arc fault interruption device

Professional Memberships

IEEE
Eta Kappa Nu