

ECE 496 Final Project Proposal

Interactive Noise Cancellation Demonstration System

Prepared by

Team 292

Yao Sun (999123586)

Taylor Straatsma (998986780)

Yuxuan Liang (998081221)

Supervised by

Prof. Khoman Phang

Executive Summary

The following document outlines our interactive noise cancellation demonstration system which will be incorporated into the outreach program at the University of Toronto (UofT) to, not only promote UofT's undergraduate programs in the areas of science, technology, engineering, and mathematics, but also to spark an interest in these fields among high school students. We will create a system, using an UDOO board as our central module, in which high school students will be able to experience, firsthand, the effects of constructive and destructive audio wave interference through the use of a single frequency sine wave emitting from two speakers.

Our system will involve hardware and software components coupled with an audio input/output in order to create an environment in which students can manipulate the sine waves to experience the phenomena mentioned above. This document will also give a more detailed explanation of the technical aspect of the project as well as the functions, objectives, and constraints influencing our design. The responsibility of financing this project will fall equally on all members of the team up to \$100 per person. We do not anticipate the need for additional finances to be requested from the Design Centre.

Table of Contents

Executive Summary

1. Project Description

1.1 Background and Motivation	4
1.2 Project Goal	5
1.3 Project Requirements	6
1.4 Validation and Acceptance Tests	7

2. Technical Design

2.1 Possible Solutions and Design Alternatives	7
2.2 System-level Overview	7
2.3 Module-level Descriptions	10
2.4 Assessment of Proposed Solution	14

3. Work Plan

3.1 Work Breakdown Structure and Gantt Chart	15
3.2 Financial Plan	16
3.3 Feasibility Assessment	17

4. Conclusion

5. References

6. Appendices

Appendix A: Gantt chart	20
Appendix B: Student-supervisor agreement form	21
Appendix C: Report attribution table	22

1. Project Description

1.1 Background and Motivation

In order to educate high school students about what engineering is and what engineers do, the Faculty of Applied Science and Engineering from the University of Toronto (UofT) launched an outreach program in May 2008 [1]. This program was intended to promote UofT's engineering program by recruiting graduate and undergraduate students and create a "Task Force" to introduce students at the secondary education level about research and technologies in electrical engineering at UofT. Students from the Ontario secondary school curriculum currently seeking further education in engineering and the sciences are required to take courses focused on classical mechanics and wave mechanics [2]. Both of these topics can be considered fundamental to UofT's engineering curriculum.

However, the current curriculum does not give students the opportunity to conceptualize the material they are learning [3], as demonstration of wave mechanics is difficult and rarely done as part of the curriculum. Research has shown that merely listening in class accounts for ~5% of the students' retention of the information whereas the use of demonstrations and audio/visual techniques account for 20-30% retention [4]. UofT's outreach program believes this is an opportunity to introduce the role of engineering in designing a demonstration to apply these theoretical concepts.

The concepts of wave mechanics is easily demonstrated through application. To provide students with an exciting and interactive demonstration of constructive and destructive wave interference, we will construct a system for students to experience this phenomenon in two ways: to have the audio automatically cancel at their location, and to do it themselves manually. The students will be exposed to how the team used computer engineering to automate this system, and signal engineering to input and output signals. In our setup, two speakers will be facing the room and emit sine waves intended to interfere with each other at a specific location defined by the microphone. While the students are kept interested by the demonstration, supplementary material

is provided to teach students about the details about the role of UofT's computer and electrical engineering in designing the automatic and manual systems.

1.2 Project Goal

The goal of this project is to demonstrate the concept of constructive and destructive audio wave interference using pure sine waves. Through this demonstration, students will follow a lesson plan resulting in their exposure and education to these concepts. Using a microphone to detect sound at a specific location within a classroom, real time digital signal processing will be applied to the input signal in order to simulate louder or softer noises corresponding to the type of interference mentioned above.

1.3 Project Requirements

The following specifications act as guidelines for our project and final design:

Functions

- The design shall use wave interference algorithms to generate the correct output signals from the input signal.
- The design shall generate two audio sine waves.
- The design shall achieve constructive and destructive interference using sine waves.
- The design shall be able to operate in a classroom setting.
- The design shall have the ability to receive sound via a microphone.
- The design shall process the input signal using digital signal processing techniques and filters.
-
- The design shall have an associated lesson plan to explain applied concepts and the role of engineering in the system.

Objectives

- The design should use physical or touch based interfaces for the user to interact with. This allows the demonstration to be interactive and should aim to further excite, motivate students.
- Because the system is intended to be portable, it should be able to be set up in no more than ten minutes.
- The design should be responsive to changes as close to real time as possible. Therefore we set a maximum response time of three seconds.

Constraints

- The design must use only two pure sine waves at variable frequencies and phases. This is because two wave channels are the minimum required channels needed to accomplish constructive and destructive interference.

1.4 Validation and Acceptance Tests

In order to test our system, we will need to setup two speakers in a classroom at U of T and connect these speakers to a computer and our signal processing board. Next, one student at a time (other students and teacher may stay in the room as well), we will make the necessary changes in our system to allow the student to experience wave interference. They will then get the opportunity to make their own changes to the system by turning knobs and/or pressing buttons. Depending on the locations of the other students in the room, they may experience slight changes in sound level as well.

2. Technical Design

2.1 Possible Solutions and Design Alternatives

Previous work on this project has been done using different approaches. Two such design models are mentioned below:

- A high level software-based approach using a computer running MATLAB and SIMULINK to handle the signal processing and wave interference algorithms to create the nodes. This was the setup used by the previous group and they encountered difficulties in keeping a continuous signal while changing phases.
- A circuit level approach, using the Arduino board, along with a digital signal processing component to manipulate the input signal and generate the desired output signals. This was another approach used by a previous group and they encountered problems trying to automatically generate nodes. The final system could not automatically minimize the sound at a desired location and only had manual controls for changing the phase.

2.2 System-level Overview

To the user, the demonstration begins with a user interface device, from which they can set the demonstration to auto or manual mode. The design is our own approach to the solution and is not based on any progress made by previous groups.

System Flow

Auto mode is designed so that the system can automatically cancel out the sine wave at the location of the microphone. In this mode:

1. The audio receiver program picks up the audio stream from the microphone via the Linux audio libraries and processes it into a sound file.
2. The sound file is picked up every ~0.1 seconds by the audio analyzer program which bandpass filters the audio file at the frequency of the sine wave and passes it to the audio input analyzer.
3. The analyzer determines the next change in phase for one of the speakers and sends the information to the sine wave generator program using another variable or control signal.
4. The sine wave generator generates the change in phase according to the signal and continues to output to the speakers through the Linux sound libraries.

Manual mode allows the user to control the frequency and phase of both signals to manually determine the nodes/antinodes of the system:

1. Manual user interface turns off the audio receiver and the audio analyzer programs via control signals (network protocol or sockets). Also obtains control of sine wave generator via control signals.
2. Sends sine wave generator information about frequency and phase chosen by user.
3. The sine wave generator generates the change in phase according to the signal and continues to output to the speakers through the Linux sound libraries.

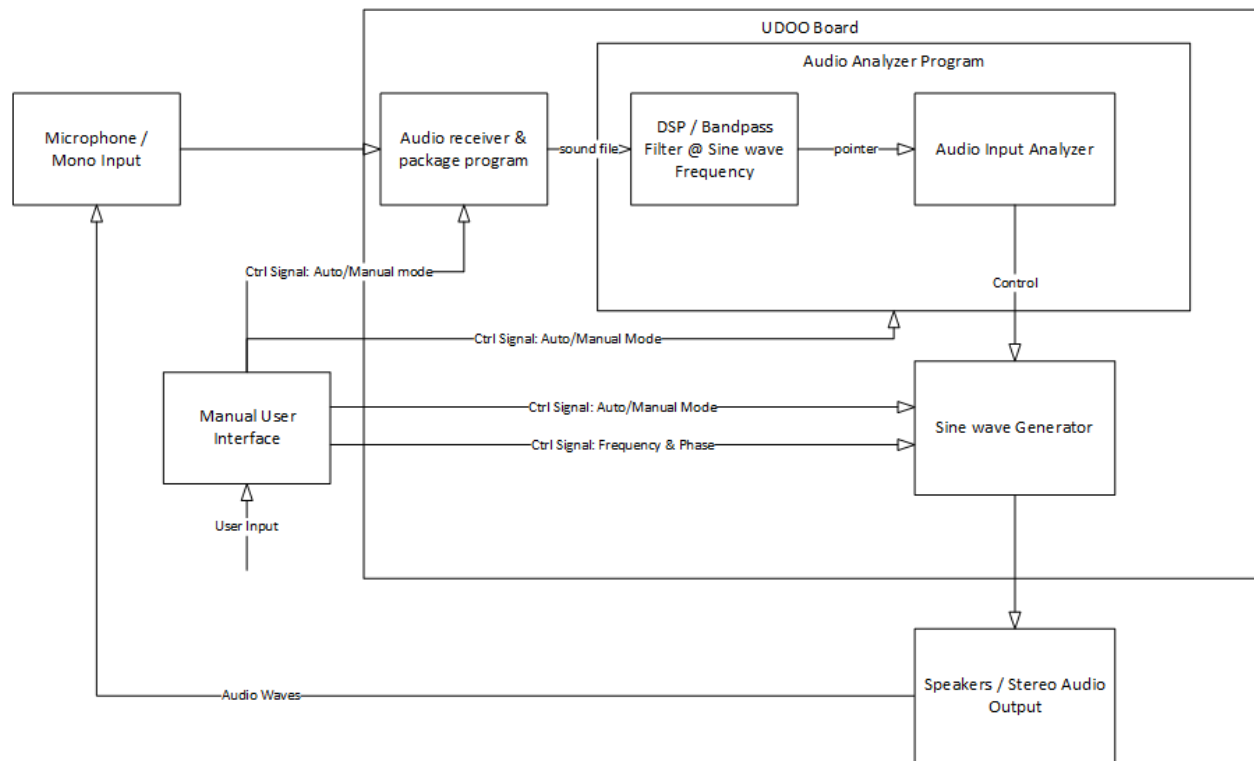


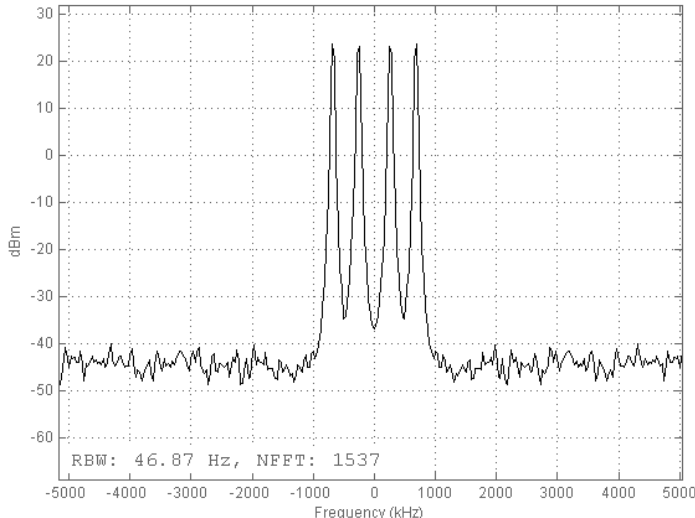
Fig 1. System Level overview of the proposed design.

2.3 Module-level Descriptions

<i>Module</i>	Microphone / Mono Input
<i>Inputs</i>	Stereo sound wave (via 3.5mm audio input jack)
<i>Outputs</i>	Boolean control signal: is_auto = true/false Audio bitstream of analog signal sampled at 48kHz
<i>Functionality</i>	<p>Microphone converts sound waves into an analog electric current to be processed by the A/D converter on the UDOO board and further processed by the Linux operating system. The microphone should be chosen to pick up ranges between 64-23,000Hz (Human hearing range). The team is currently working with ranges between 261 Hz to 445 Hz.</p> <p>If we consider the A/D conversion done to be part of the Audio input, the output is an audio bitstream sampled at 48kHz by the operating system.</p>

<i>Module</i>	Audio Receiver & Package Program
<i>Inputs</i>	Audio bitstream sampled at 48kHz
<i>Outputs</i>	Boolean control signal: is_auto = true/false One second long sound file with appropriately named title in the waveform audio file format (.wav)
<i>Functionality</i>	<p>The audio receiver program serves as a link between the audio bitstream and the audio analyzer program. This is a constraint imposed by the DSP library being used. Due to the inability to handle continuous streams of data, a small portion must first be saved to allow for processing.</p> <p>The role of the package program is to package ~0.1 second of the audio input and place it in the correct location for the audio analyzer program to process. The value of 0.1 was chosen so that the system can quickly sweep the room with different phases to determine the nodes/antinodes of the room.</p>

<i>Module</i>	Audio Analyzer Program / DSP - Bandpass Filter
---------------	--

<i>Inputs</i>	Waveform audio file (.wav)
<i>Outputs</i>	Boolean control signal: is_auto = true/false Variable/pointer to processed audio in memory (n) Variable/pointer to processed audio in previous iteration (n-1)
<i>Functionality</i>	<p>The DSP applies a bandpass filter centered at the frequency of the generated sine wave to remove ambient noise. Give an input signal, the team chose to use the DSPs given biquad filters (a simple filter with two poles and zeros) to filter the input signal.</p> <p>By using MATLAB's Fdatool, the team designed an array of biquad filters connected in series to create higher order systems.</p> <p>We simulated the parameters and using order a 10th order system (5 biquad filters in series) we were able to achieve satisfactory filtering around a 261Hz sine wave:</p> <p>Given the inputs:</p> <ul style="list-style-type: none"> • Desired sine wave at 261 Hz • Undesired sine wave at 630 Hz • White noise across the spectrum <p>Output without 10th order biquad filter:</p>  <p>Fig 2. The simulated input system, with a desired signal and undesired extraneous noise.</p> <p>With 10th order biquad filter:</p>

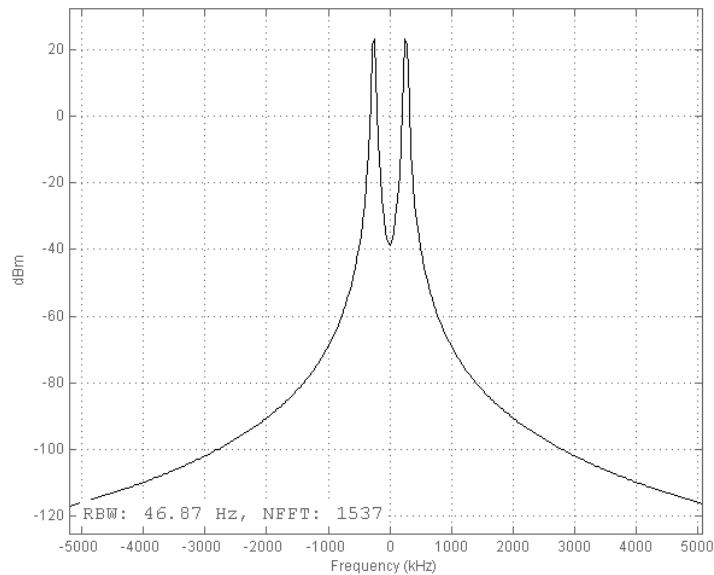
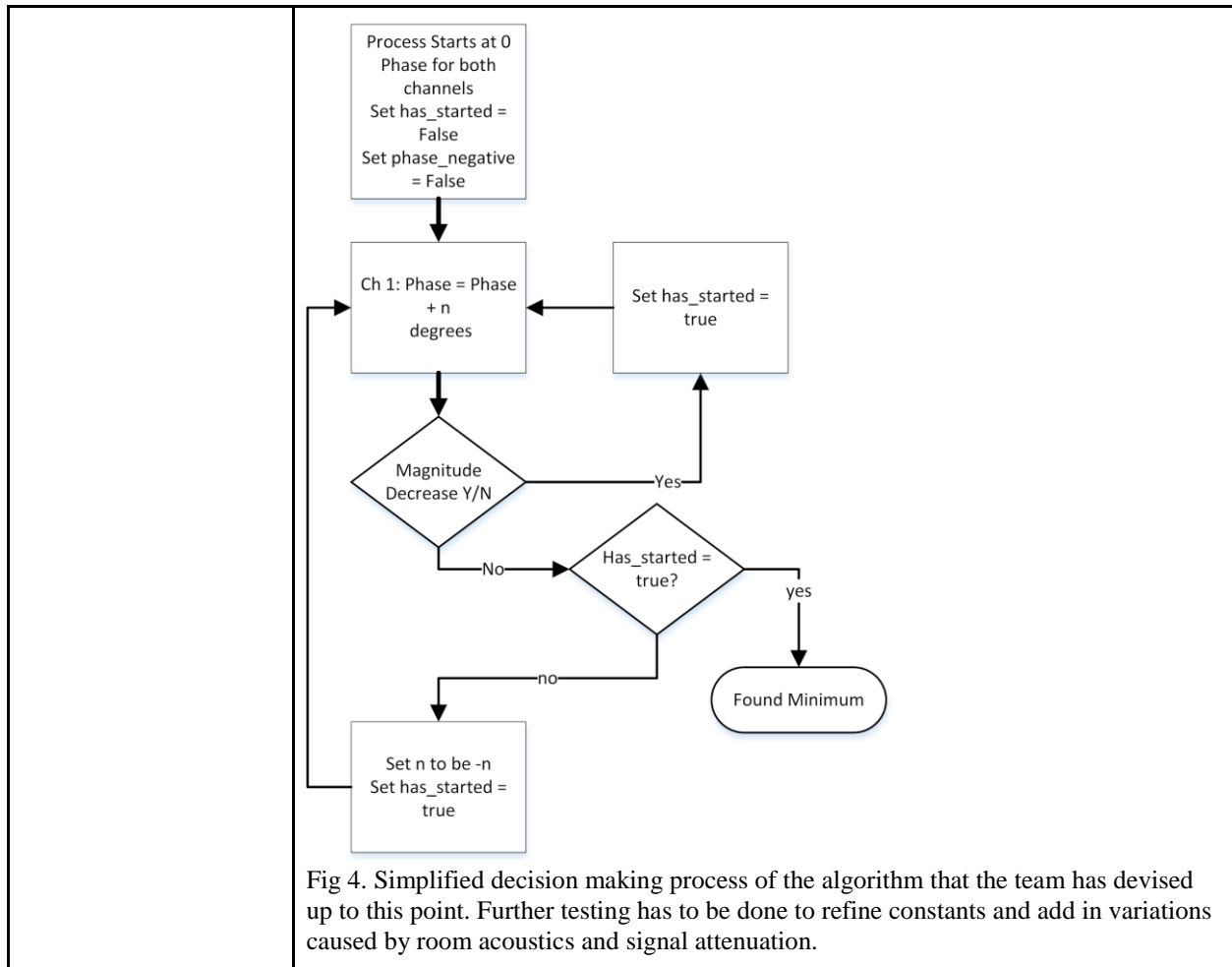


Fig 3. The filtered signal using a 10th order biquad filter array. Note that only the desired peaks are present in the output.

The follow filter can be realized using a DSP library included in PYO, an open source python audio processing library the team has chosen to use.

<i>Module</i>	Audio Analyzer Program / Audio Input Analyzer
<i>Inputs</i>	Boolean control signal: is_auto = true/false Variable/pointer to processed audio
<i>Outputs</i>	Control variable consisting of a dictionary containing: {ch1_mag:<ch1_magnitude>, ch2_mag:<ch2_magnitude>, freq_1:<frequency>,freq_2:<frequency>,phase:<phase>}
<i>Functionality</i>	<p>The audio input analyzer is the control unit of the system. Based on changes in the magnitude and frequency between the last audio sample and the present one, it will:</p> <ol style="list-style-type: none"> 1. Make adjustments to the speaker system <ol style="list-style-type: none"> a. Adjust the magnitude of each speaker individually. b. Adjust the phase of one speaker to create antinodes at certain locations. 2. Remove the previous (n-1) sample, and replace it with the present sample (n). <p>At present, the team devises the following algorithm for determining anti-nodes/nodes at the location of the speaker, presented in the form of a logic chart:</p>



<i>Module</i>	Sine Wave Generator
<i>Inputs</i>	Boolean control signal: is_auto = true/false Control variable consisting of a dictionary containing: {ch1_mag:<ch1_magnitude>, ch2_mag:<ch2_magnitude>, freq:<frequency>,phase:<phase>}
<i>Outputs</i>	2-Channel analog signal (3.5mm audio output jack)
<i>Functionality</i>	<p>The sine wave generator is responsible for playing two sine waves at desired magnitude, phase and frequency. By default, only channel 1 can make changes to phase.</p> <p>The sine wave generator is a simple python program written using libraries provided by PYO, a lower level view reveals that the audio library utilizes various Linux libraries to produce two continuous sine signals of varying magnitude, phase, frequency.</p>

	If we consider the UDOO board D/A converter and the Linux libraries to be part of this module, the output is a two channel analog signal.
--	---

<i>Module</i>	Stereo Speakers
<i>Inputs</i>	2-Channel analog signal (3.5mm audio output jack)
<i>Outputs</i>	Stereo sound wave
<i>Functionality</i>	The Stereo speakers are responsible for emitting the two sine waves according to the signal generator.

2.4 Assessment of Proposed Solution

Our design utilizes off-the-shelf hardware components and our custom program designed to run on the Linux operating system. This allows us to generate output audio waves and utilize DSP filters (such as the biquad filter) to process the input signals using a software based approach. By doing so, we avoid a lot of complications that come with analog to digital (A/D) conversion, and designing DSP filter circuits.

The main faults of the previous implementations we've looked at were:

- MATLAB and SIMULINK were too restrictive about how signal input and output occurred, therefore the group had difficulties in maintaining a continuous sine wave while switching phases.
- The circuit level approach was too difficult to implement. Although altering the sine wave manually worked, implementing an automatic wave interference system was too difficult using Arduino hardware.

Tradeoffs

We address these faults by picking a middle ground between the two:

- We create our own programs running on digital signal processing and audio input/output libraries. This gives us more flexibility.
- We use a higher level approach, letting the off-the-shelf circuitry and operating system handle the analog signals. Additionally, by moving to a processor-based approach we can

leverage existing technologies such as the Python programming language and Linux operating system to speed up our development time.

- Cheaper to produce compared to the system using MATLAB.

However the benefits of our system come at a financial and technical cost:

- The UDOO board, which runs the Linux OS and in turn our DSP and signal generator costs around \$150.00 CDN each.
- Despite avoiding MATLAB in our final project, it is still necessary to prototype our signals and filters in MATLAB. Whereas the higher level approach using matlab allows us to directly use the FIR/IIR filters implemented in Fdatool using Simulink blocks, our approach uses filters that are either given to us by the PYO Audio Processing Libraries or ones coded ourselves in Python. This can be a more complex process than using Simulink blocks.

3. Work Plan

3.1 Work Breakdown Structure (WBS)

Task #	Task	Yao	Taylor	Alex
1	• Psychoacoustic experiments			
1.1	□ Speakers facing eachother		R	
1.2	□ Speakers facing same direction			R
2	• Software/UDOO board configuration			
2.1	□ Setup Linux environment on UDOO board	R		
2.2	□ Program audio input and output component	R		
2.3	□ Program input and output DSP system	R		
3	• Mathematical modelling of setup		R	A
4	• Finalize project proposal document	A	A	A
5	• UDOO board			
5.1	□ Incorporate board into setup	R		
5.2	□ Experiment with board and speakers		R	A
6	• Develop student learning plan			R

7	<ul style="list-style-type: none"> Select speakers and microphone for demonstration 	R	A	
8	<ul style="list-style-type: none"> Finalize final report 	A	A	A
9	<ul style="list-style-type: none"> Design fair and student demonstration 	A	A	A

R = Responsible, A = Assisting

Gantt Chart

Refer to Appendix C

3.2 Financial Plan

Budget Request

Item	Priority	Cost/per unit	Free (y/n)	Quantity (or # hours)	Total Cost
Costs					
UDOO prototype board	1	\$135.00	y (wish to keep)	1	\$0.00
Stereo speakers	1	\$30.00	n	1	\$30.00
Smartphone	3	\$0.00	y (wish to keep)	1	\$0.00
MATLAB	2	\$0.00	y (wish to keep)	3	\$0.00
Microphone	1	\$30.00	n	1	\$30.00
Python DSP library	2	\$0.00	y	1	\$0.00
Linux Operating System	2	\$0.00	y	1	\$0.00
Digital Logic Analyzer	3	\$100.00	n	1	\$100.00
Contribution					
Total Contribution from Students	N/A	-\$20.00	N/A	3	-\$60.00

Contribution from Supervisor	N/A	-\$100.00	N/A	1	-\$100.00
Net requested amount from Design Center	N/A	N/A	N/A	N/A	\$0.00

Contingency Explanation

Priority 1

- UDOO prototype board - main piece of hardware that contains the processor needed to run our operating system and DSP program. We propose to use a less accessible and more expensive hardware (ex. laptop) to run our program in case the funds are not obtainable. This will require changes to the software side of the system.
- Stereo speakers - main piece of hardware for sound output. We propose to use speakers provided by the supervisor.
- Microphone input - main piece of hardware for sound input. We propose to simulate audio input via layer sine waves with ambient noise if a microphone is not available.

3.3 Feasibility Assessment

Skills and Resources

The skills and resources required for this project include, but are not limited to the following:

Skills

- An understanding of the physics behind constructive and destructive wave interference.
- An understanding of programming in the linux environment.
 - The team have all had prior experience with the linux environment and further knowledge can be acquired from online resources.
- An understanding of programming using the python scripting language.
 - The team has members who have had prior experience with python and will help other members learn through online resources.

Resources

Acquired:

- The team has already acquired the UDOO board, an embedded system running linux on ARMv7 cores [5]. This board will run the digital signal filtering and signal output program.
- The team is using the PYO sound processing and filtering library, an open source library written in python that is used to generate and filter signals [6].

Expected:

- An audio input and output source has not been finalized. At present we are considering buying our own microphone and using speakers provided by our supervisor.
- The team expects to test the system in a multitude of locations, including the anechoic chamber (SF 4th floor) and classrooms in Bahen, Sanford Fleming and Galbraith buildings.

Risk Assessment

The only immediate risk to the design team is the implementation of the audio input through the microphone of a smartphone. In order to locate the users while receiving audio input, an application specific to this design would have to be developed. Developing this application means adding another subsystem to our design which will potentially increase the risk [7]. The team will be sure to have a fallback plan which simply uses simple peripherals to input the audio before a mobile application is implemented.

4. Conclusion

To demonstrate the concept of constructive and destructive audio wave interference to high school student, our team have decided to develop a UDOO board based audio input/output system. Controlled by a physical user interface, our system will process the input audio signals and produce the corresponding output signals. Compared to the alternatives proposal, our design, which consists of 4 major modules, will have a higher level approach. With proper time and financial management, our Interactive Noise Cancellation Demonstration System can educate high school students about the role of Electrical and Computer Engineering in our noise cancellation system and assist the students in understand wave interference.

5. References

- [1] S. McCahan, (2009). *Report of the Outreach Task Force* [Website]. Available: <http://www.engineering.utoronto.ca/Assets/AppSci+Digital+Assets/pdf/Staff+and+Faculty/outreachtaskforce.pdf>
- [2] Ontario Ministry of Education. *The Ontario Curriculum Grades 11 and 12* [Website]. Available: <http://www.edu.gov.on.ca/eng/curriculum/secondary/science1112curr.pdf>
- [3] Alan Slavin, (2007). *Has Ontario Taught its High-School Students not to Think* [Website]. Available: <http://www.universityaffairs.ca/has-ontario-taught-its-high-school-students-not-to-think.aspx>
- [4] Katie Lepi, (2014). *The Beginner's Guide to the Learning Pyramid* [Website]. Available: <http://www.edudemic.com/the-learning-pyramid/>
- [5] UDOO, (2014). *Android device, Powerful computer, Arduino-compatible, ADK 2012* [Website]. Available: <http://www.udoo.org/features/>
- [6] Ajax sound studio, (2012). *Pyo - Python digital signal processing module* [Website]. Available: <https://code.google.com/p/pyo/>
- [7] John Downer, (2009). *When Failure is an Option: Redundancy, reliability and regulation in complex technical systems* [Website]. Available: <http://eprints.lse.ac.uk/36537/1/Disspaper53.pdf>

6. Appendices

Appendix A: Gantt Chart

