# DEVELOPMENT OF AN AUTOMOTIVE STEERING-WHEEL MOUNTED AUDIO USER INTERFACE

A Thesis
Presented to
The Academic Faculty

by

Matt Edwards

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in the
School of Mechanical Engineering

Georgia Institute of Technology

May 2015

# DEVELOPMENT OF AN AUTOMOTIVE STEERING-WHEEL MOUNTED AUDIO USER INTERFACE

Approved by:

Dr. Ken Cunefare, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Wayne Book
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Erica Ryherd
School of Architectural Engineering and Construction
University of Nebraska – Lincoln

Date Approved: 04/22/2015

# ACKNOWLEDGEMENTS

I wish to thank the sponsor company for their support on this project, and my lab mates for providing assistance whenever I needed it. Thank you to Dr. Erica Ryherd and Dr. Wayne Book for being a part of my thesis committee. Thank you to Dr. Kenneth A. Cunefare for playing such a prominent role in growing my knowledge of acoustics, my experience in research, and my principles as an engineer.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# NOMENCLATURE

**Psychoacoustic Metrics –** *Italics*

*Long*
*Dark*
*Sharp*
*High Frequency*
*Intense*
*Expressive*
*Tonal*
*Short*
*Loud*
*Strong*
*Smooth*
*Low Frequency*
*Solid*
*Reliable*
*Powerful*

**Command Functions –** *Italics*

*Set +*
*Set –*
*On*
*Off*
*Answer Call*
*End Call*
*Medio Mode*
*Next Track*
*Previous Track*
*Play/Pause*
*Mute On*
*Mute Off*
*Volume Up*
*Volume Down*
*Activate*

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ANOVA | Analysis of Variance |
| BLOB | Binary Large OBject |
| DAQ | Data Acquisition System |
| DAT | Data File |
| dB | Decibel |
| dBHL | Decibel Hearing Level |
| FFT | Fast Fourier Transform |
| GUI | Graphical User Interface |
| IAL | Integrated Acoustics Laboratory |
| IRB | Institutional Review Board |
| NI | National Instruments |
| PDV | Portable Digital Vibrometer |
| SD | Semantic Differential |
| SNR | Signal-to-Noise Ratio |
| WAV | Waveform Audio File |
| WT | Wavelet Transform |

# SUMMARY

A tier 1 automotive components supplier has developed a haptic user interface system for the purpose of replacing currently existing physical buttons mounted on automotive steering wheels. The system is intended to generate acoustic and vibratory feedback to the user for the purpose of mimicking the sound and feel of mechanical buttons. The work performed in this thesis served to investigate what the input waveform should be to the system in order to generate a desired audio output signal. Additionally, subjective testing in the form of a sound jury was conducted in order to identify which types of sounds should be associated with which command functions in order to indicate successful initiation of the intended function. A model for grouping command functions into banks, where each bank is assigned a single sound, was then developed for the purpose of reducing the total number of button sounds used in an automotive environment.

# CHAPTER 1. INTRODUCTION

Steering wheel-mounted button controls are a ubiquitous feature in automotive vehicles. Functions commonly found on the dashboard or center console are migrating to new positions on the steering wheel itself. Cruise control, sound system adjustment, and Bluetooth phone answering are some of the more commonly used functions which can now be found on the steering wheels of many makes and models of automotive vehicles. However, some go even further to add controls for auxiliary functions, such as air conditioning and steering wheel heaters. These steering wheel button controls serve to increase overall driving safety by allowing drivers to keep their eyes on the road and hands on the wheel at all times, rather than requiring them to move their hands and focus to the center console for a moment to adjust the controls located there.

The quantity of steering wheel button controls may be increasing, but the available real estate remains fixed, as steering wheels themselves are not getting any larger. The ability to replace these controls with a single steering wheel mounted user interface capable of providing access to multiple vehicle functions would allow an automotive manufacturer to increase the number of readily accessible controls without taking up more space on the steering wheel or reducing the size of the buttons. It would serve to increase driver safety even further by placing all the buttons in a single area on the steering wheel. This could reduce the hand movements of drivers to only the movement of a single thumb, as they would not even have to move their hands to a different location on the steering wheel to press the desired button.

The sponsor company for the work presented in this thesis has developed such a steering wheel mounted user interface, which is intended to generate audio and tactile signals that mimic the sound and feel of physical buttons. However, special care must be taken in choosing which sounds are to be matched with which vehicle functions. It has been shown that when the sound of a product does not match the expectation consumers have about how they believe that product should sound, the result is a reduced perception of quality about the product of behalf of the consumers [1]. Factors such as the sound of a closing car door can influence a customer's opinion of the value of that vehicle [2]. The sounds used in alarms are intentionally chosen because they effectively communicate the message that something important is in need of immediate attention [3]. However, no work could be found which provides a link between command functions and the relevance or appropriateness of the sounds associated with those functions.

Although it may seem ideal to investigate each individual command function and identify the best possible sound for it to be matched with, this approach may not always be the most feasible or the most desirable. Restrictions such as the capabilities of the available hardware may limit the number of sounds able to be stored. Alternatively, if there are too many different sounds being played, users may become overwhelmed with the difficulty of remembering which sound signifies which action has just taken place. Identifying a small subset of sounds that, when matched with a larger set of groups of multiple command functions, indicate successful initiation of the intended functions, would allow one to avoid these possible restraints.

Prior work has been conducted to investigate the connection between the various psychoacoustic metrics of button sounds. However, no work could be found which investigates those metrics' connections to vehicle command functions. Therefore, the objective of the work in this thesis is twofold. First, to develop a test method that evaluates the psychoacoustic characteristics of button sounds that, when mapped to certain groups of vehicle command functions, indicate successful initiation of the intended function. Second, to evaluate the capabilities of a haptic transducer system in order to identify its suitability for synthetically reproducing those button sounds in a steering wheel-mounted automotive user interface. The following sections of this chapter introduce the background on the human perception of sound, a note on the nomenclature used in this thesis, and an overview of prior relevant work.

## 1.1. Background

Determining the appropriateness of button sounds as they relate to vehicle command functions requires an investigation into the human perception of sound, which falls under the field of psychoacoustics. Human hearing is not a purely mechanical process: factors other than the waveform of a sound dictate how a person actually perceives it. The human ear is nonlinear with respect to both sound intensity and frequency, meaning that a tone of constant intensity played at varying frequencies will not necessarily be perceived by the human ear as being the same "volume" at each of those frequencies. The field of psychoacoustics seeks to quantify these perceptive nonlinearities in order to gain a deeper understanding of how to intentionally manipulate sounds to successfully indicate indented function initiation.

The appropriateness of the sounds made by a product is related not only to the human perception of the sounds themselves, but also to a person's prior opinions of that product or expectations of how that product should perform. The ability to meet a person's expectations of that product's sound determines the person's perception of the quality of that product. Utilization of a sound jury is one way of identifying which psychoacoustic metrics of a sound determine that product's quality [4].

### 1.2. Nomenclature

In this thesis, a number of terms are used to describe characteristics that the reader may be unfamiliar with. A formatting convention was used to allow the user to identify those instances in which these terms are being used. The full list of terms is available in the forward of this thesis. All psychoacoustic metrics and command functions that were evaluated by jurors are in *italics*, such as *tonal* and *long (*psychoacoustic metrics) or *set +* and *volume up* (command functions).

### 1.3. Prior Work

The purpose of this section is to present a review of prior work relevant to the creation of an automotive button sound test method. Studies conducted in regards to transient push button sounds, automated evaluation of button sound quality, and the link between look, feel, and sound of virtual buttons are discussed.

#### 1.3.1. Transient Push Button Sounds

Work was conducted by Ishimitsu et al. at Hiroshima City University to develop a method of evaluating the impression of transient sounds that are non-periodic and non-stationary, such as those used for push buttons. While methods already exist to analyze steady state sounds and periodic, stationary transient sounds, they cannot be accurately applied to push button sounds. The objective was to use the semantic differential (SD) method to extract the impression of eleven different button sounds, which all came from a car audio unit, then discover the relationship between each sound's representation of the wavelet transform (WT) and sound impression. That is, the goal was to find a quantitative way of describing the qualitative characteristics of push button sounds [5].

Figure 1 shows the jury results from the test conducted. Eleven button sounds from six car audio unit models were utilized. In an anechoic chamber, a microphone, placed 30 cm from the audio unit, recorded three button pushes per sound. Both the sound of the button being pressed (push sound) and the sound of the button being released (back sound) were recorded. A jury was then asked to rate the button sounds on a “like-dislike” scale. These ratings were compared to the loudness of the button sounds, identifying a positive linear regression (i.e. the louder a sound becomes, the worse the impression is on the listener) [5].

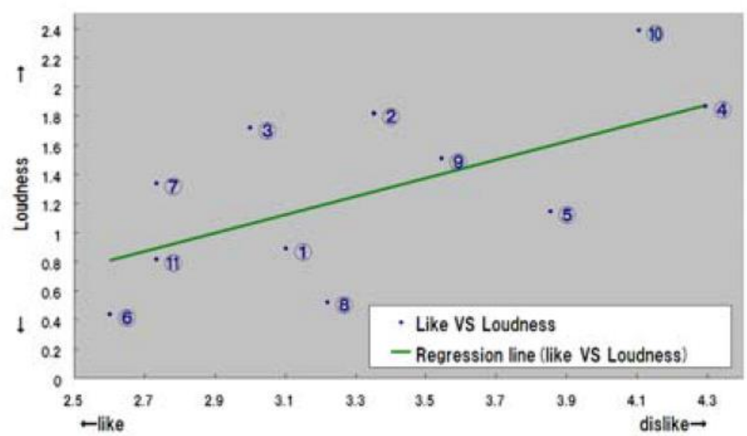



Figure 1. The Relation Between the Score of the “like-dislike” of the Jury Test and Loudness [5]

The psychoacoustic dimensions used in the study to describe sounds were “loudness”, “pitch”, and “sound quality and tone”. The SD method was used to quantify the number of dimensions present for each button sound. Sixty-seven healthy people were used in the SD method jury test. Each button sound was matched with several pairs of adjectives, and the jury members were asked to rate which adjective from each pair better described the sound on a 1 – 5 scale. Factor analysis and WT matching analysis was then conducted to transform these quantitative responses to qualitative values. It was found that “like” button sounds had low metallic, low force factor, and high esthetic factor scores [5].

#### 1.3.2. Automated Evaluation of Button Sound Quality

Work was conducted by Fujinoki et al. at the Oshima National College of Maritime Technology to develop an automated process of evaluating the sound quality of push button sounds based on the their psychoacoustic characteristics. Triangular biorthogonal wavelets that were two-dimensional and non-separable were used to extract features for local and global sound characteristics. The discrete wavelet transform was used to decompose the signal while maintaining the isotropy of the energy. Therefore, the features extracted from the isotropic component of the sound could be applied to the whole sound, including the successive sharp transients typical of push button sounds [6].

Using these methods, a plot was developed for each button sound graphing the wavelet-based cumulative sound pressure as a function of time. Local variations in the wavelet-based cumulative sound pressure corresponded to the localized transient sound properties of the buttons, while the global variations in the wavelet-based cumulative sound pressure corresponded to the button sound reverberation. To automatically assess the sound quality, the sum of squared difference method and dynamic programing matching were used. When using low scale features, the method was able to automatically match the “preference” and “annoyance” of each push button sound [6].

### 1.3.3. The Look, Feel, and Sound of Virtual Buttons

Work was conducted by Hoggan et al. at the University of Glasgow to discover a way to design mobile touchscreen user interfaces that are more pleasing to consumers. The objective was to develop a method for assigning the visual, audio and tactile characteristics of a touchscreen button such that the three parameters complement each other and result in a higher perceived quality. Touchscreen buttons, while having the benefit of almost limitless visual design possibilities, traditionally do not possess audio or tactile characteristics that match the physical buttons they are trying to mimic (e.g. a touchscreen keyboard vs a physical keyboard) [7]. This has been shown to result in a lower perceived quality by the user [8].

For the congruence experiment, test subjects were given a series of visual or audio/tactile buttons and a series of events that those buttons may trigger. They were then asked to match each button with an event, choosing the ones they believed to be the most appropriate. The device used was a Nokia 770 Internet Tablet. Ten different physical buttons were studied in an effort to gain insight into ideal touchscreen button design. Several features of these physical buttons were recreated virtually and implemented into the touchscreen device [7].

The tactile feedback was created by a piezo-electric feedback device and a vibrotactile feedback device. The former is capable of creating localized vibration of the touchscreen itself, and single transients that represent physical button feedback. The latter is capable of creating longer feedback which vibrates the mass of the whole tablet. The audio feedback was created by the natural sound of the tactile sensors. Four audio signals were produced: a “soft click” for both the piezo-electric and vibrotactile device, a “clicky click” for the piezo-electric device, and a “long rough click” for the vibrotactile device. [7].

Figure 2 shows the results of the perception of quality of various button look, feel, and sound combinations. Four sounds were given names of *Vibra 1*, *Piezo 1*, *Vibra 2*, and *Piezo 2*, which were chosen to qualitatively describe the characteristics of the sounds. For circular touchscreen buttons, it was found that small flat buttons with soft piezo clicks and small raised buttons with soft vibra clicks were the most congruent. Additionally, large buttons were found to be most congruent with soft vibra clicks, both when flat and raised. For rectangular touchscreen buttons, it was found that small flat buttons with short sharp “clicky” piezo clicks, raised buttons with soft piezo clicks, large flat buttons with short sharp “clicky” piezo clicks, and large raised buttons with soft vibra clicks were the most congruent [7].

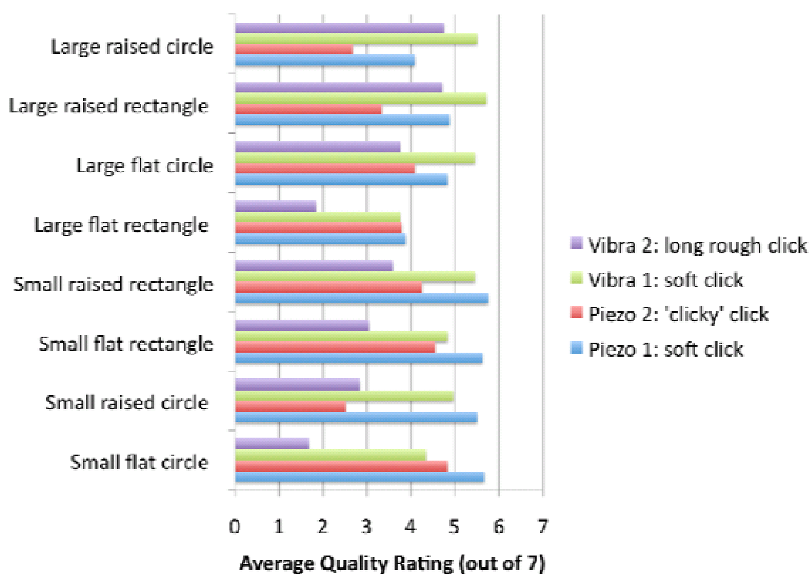



Figure 2. Average Rating of Quality (or Congruence) for each Crossmodal Touchscreen Button [7]

# CHAPTER 2. EQUIPMENT OVERVIEW AND LABORATORY ENVIRONMENTS

This chapter presents an overview of the unique hardware and software utilized, as well as the two environments used for conducting all experiments performed in this thesis. Which environment was used depended on the type of data being acquired (audio or voltage).

## 2.1. Equipment Overview

The sponsor company provided several components that were utilized in this thesis, the first of which is a haptic transducer system. The system is made up of two individual components: the exciter and the haptic board. Together these components are intended to generate a single signal containing both audio and tactile waveforms. A multitude of switch packs were also provided, of which the audio responses were measured experimentally. Unique software, also provided by the sponsoring company, was used to communicate with these components and perform actions such as generating/loading waveforms and modifying volume settings. The specifics of these components are detailed below.

### 2.1.1. Exciter

The exciter is a small transducer capable of emitting both tactile and audio signals. It can be embedded inside another material and/or device (such as a steering-wheel), causing that device to vibrate in response to the movement of the exciter. The work done in this thesis focuses on the audio capabilities of the exciter.

### 2.1.2. Haptic Board

Figure 3 shows a diagram of the haptic user interface. The haptic board is the relay between the computer and exciter that allows specific waveforms to be generated. It consists of a microprocessor and a small amount of memory storage. The haptic board can store signals in the form of custom binary large object (BLOB) files and send them to the exciter at the command of the user. It also stores signal characteristics such as volume and gain. The board can connect to a computer via a COM port, and is powered via USB.

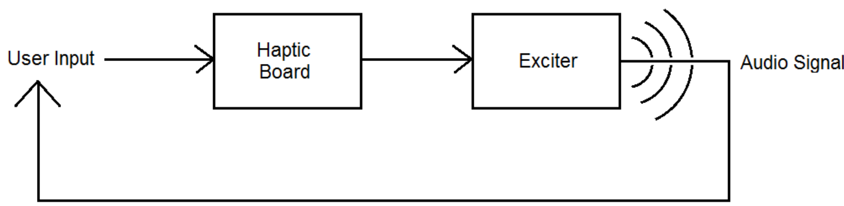



Figure 3. Haptic User Interface Diagram

### 2.1.3. Switch Packs

Figure 4 shows an example photo of a steering wheel switch pack. The term “switch pack” refers to a cluster of buttons that are a part of the construction of the steering wheel itself. Each switch operates a single vehicle command function, such as *volume up* or *cruise on.* Eleven switch packs were supplied, containing a total of sixty-six switches. They are used in steering wheels for current-model automotive vehicles from a multitude of global automotive manufacturers.


Figure 4. Example of a Steering Wheel Switch Pack

#### 2.1.4. Haptic Board Communication Software

The haptic board communication software allows the user to communicate with the haptic board. It contains a user interface where the user can load signals to be stored on the board, modify various signal parameters, and request the board to send a signal to the exciter.

#### 2.1.5. Signals BLOB Generator Software

The Signals BLOB Generator is a piece of computer software that converts multiple audio and tactile signals in waveform audio file (WAV) format to a single data (DAT) file. The Signals BLOB Generator requires at least one tactile and one audio WAV file for each BLOB file it creates, and is programmed to prevent the user from generating a BLOB file that is too large for the memory capacity of the haptic board.

### 2.2. Laboratory Environments

#### 2.2.1. Hemi-anechoic Chamber

The recording environment for all audio measurements was the hemi-anechoic chamber at the Georgia Institute of Technology, which is part of the Integrated Acoustics

Laboratory (IAL). This includes the completion of the switch pack audio response measurement and sound jury test. The hemi-anechoic chamber has a low frequency cutoff of 80 Hz [9]. The background noise of the chamber has an equivalent sound level of 21 dB [10], making it adequate for conducting reliable acoustic measurements.

### 2.2.2. Active Control Laboratory

The recording environment for all tactile measurements performed was the Active Control Laboratory at the Georgia Institute of Technology, which is part of the IAL. This includes the haptic board and exciter output measurements. As noise was not a factor, it was not deemed necessary to perform these tests in the hemi-anechoic chamber. The Active Control Lab has multiple optical tables with integrated damping, making them ideal for conducting vibration experiments.

# CHAPTER 3. EXPERIMENTAL METHOD

The purpose of this chapter is to present the methods used in the hardware measurement and sound jury tests. Details from both hardware measurement tests are presented in this chapter, as well as both parts of the sound jury test. For the hardware measurement tests, the first test is referred to as the switch pack audio response measurement, the second is referred to as the haptic board output measurement, and the third is referred to as the exciter output measurement. For the sound jury test, part 1 is referred to as the psychoacoustic metric section, and part 2 is referred to as the command function section.

## 3.1. Switch Pack Audio Response Measurement

In developing the sound jury experiment, one of the necessary tasks was to identify the sample sounds to be used. The sample sounds presented to the test subjects should not contain peripheral noise which is not part of the intended signal to be evaluated, and sound duration should be at least 3 – 5 seconds [13]. If such a duration is not feasible, allowing for multiple playbacks can be used [13]. As part of this task, the audio responses of all supplied switch packs were measured, and several of the recorded waveforms were chosen to be used as sample sounds in the sound jury test. This decision process is detailed in section 3.4.1.

The experimental setup of the switch pack audio response test conducted in the hemi anechoic chamber, including microphone placement, is shown in Figure 5 and Figure 6. Each switch pack was clamped to a stand, positioned in the center of the room.

One ½” random incidence microphone was used to capture the sound emitted, positioned normal to the switch.

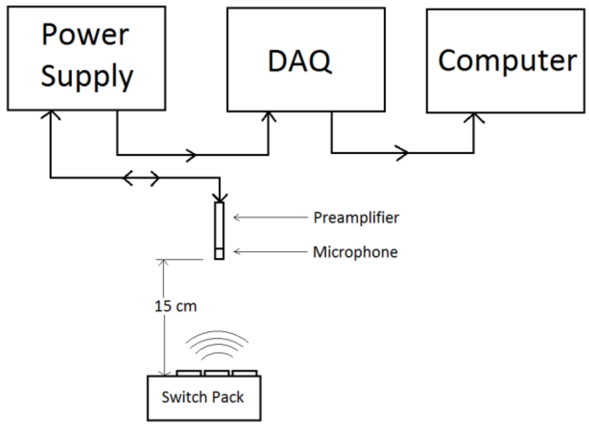



Figure 5. Switch Pack Audio Experimental Setup

Figure 6. Photograph of Switch Pack Audio Experimental Setup

The microphone was a Larson-Davis ½” random incidence model 2560, serial number 2467, with a model PRM900C pre-amp. The microphones were calibrated with a Larson-Davis CAL250. The acquisition equipment consisted of a Larson-Davis Model 2200C power supply, serial number 0784, National Instruments USB-4431 data acquisition chassis (NI USB-4431 DAQ), serial number 1915CC7, and NI Signal Express software. The data acquired for the exciter was taken 44,100 samples/second to fully capture the 20-20,000 Hz spectrum of human hearing.

A background noise sample was first acquired for the chamber. Next, the recording sequence was initiated, and a switch was pressed. The process was then repeated for the switch release. The experimenter’s index finger was simply used to press and release each switch.

### 3.2. Haptic Board Output Measurement

The haptic board voltage output test was conducted in the Active Control Lab. Three impedance conditions were chosen to be measured: exciter loaded, resistor loaded, and no load. The purpose of this was to identify whether the impedance loading on the haptic board played any role in the shaping of the output signal. For the exciter loaded condition, the exciter was attached to the output port on the haptic board, as it typically would be when in its standard operating condition. For the resistor loaded condition, a decade resistance box was attached to the output port on the haptic board, and the resistance was set to be equivalent to that of the exciter (8.5 Ω). A multimeter was used to verify both the exciter and resistor had equal resistance. For the no-load condition, nothing was attached to the output port on the haptic board.

The experimental setup is shown in Figure 7. To distinguish the three impedance conditions, the no-load setup is shown in blue, the exciter loaded setup in green, and the resistor loaded setup in red.

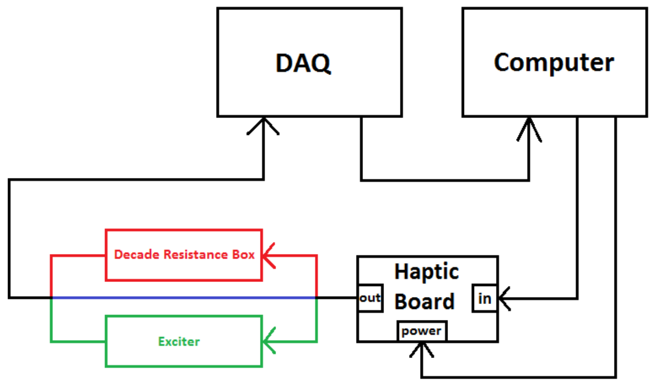


Figure 7: Haptic Board Voltage Output Experimental Setup. No-Load Setup in Blue, Exciter Loaded Setup in Green, Resistor Loaded Setup in Red.

The decade resistance box used was a model RS-201W, serial number 09291225. The acquisition equipment consisted of a National Instruments USB-4431 data acquisition chassis (NI USB-4431 DAQ), serial number 1915CC7, and NI Signal Express software. The data acquired for the exciter was taken at 44,100 samples/second to fully capture the 20-20,000 Hz spectrum of human hearing.

Table 1 summarizes the different testing configurations used. A sine sweep was chosen as the input signal, which swept from 20 Hz to 5512.5 Hz, where the high frequency limit was chosen based on the Nyquist frequency associated with an 11,025 Hz sampling rate [11]. The MATLAB code used to generate this file can be found in Appendix O. The no-load condition was measured at all four gain settings available on the haptic board (6, 12, 18, 24 dB), which revealed a limit on how high the volume could practically be set. The haptic board allows the user to specify an integer volume setting between 0 and 255. It was found that a volume of 35 was the highest setting possible before distortion occurred, regardless of gain setting. Therefore, each signal was measured at a gain of 6 dB and at two volume settings: 35 and 36.

Table 1. Haptic Board Voltage Measurement Configurations

| Impedance Load | Gain (dB) | Volume |
|---|---|---|
| No load | 6 | 35 |
| | | 36 |
| | 12 | 35 |
| | | 36 |
| | 18 | 35 |
| | | 36 |
| | 24 | 35 |
| | | 36 |
| Exciter loaded | 6 | 35 |
| | | 36 |
| | | 35 |
| | | 36 |
| | | 35 |
| | | 36 |
| | | 35 |
| | | 36 |
| Resistor loaded | 6 | 35 |
| | | 36 |
| | | 35 |
| | | 36 |
| | | 35 |
| | | 36 |
| | | 35 |
| | | 36 |

The signals were converted from WAV to DAT format using the Signals BLOB Generator software. The DAT files were written to the haptic board using the haptic board communication software. The communication software was also used to communicate with the exciter to initiate the audio signal. The recording sequence was initiated, and then the communication software was used to prompt an output signal from the exciter.

### 3.3. Exciter Output Measurement

The experimental setup for the exciter output test is shown in Figure 8. The test was conducted in the Active Control Lab. The exciter was positioned in a clamp on a rigid table and one portable digital vibrometer (PDV) was used to capture the motion, which was focused on the vibrating head of the exciter. The PDV was set to a velocity setting of 500, a 22 kHz low pass filter, and no high pass filter.

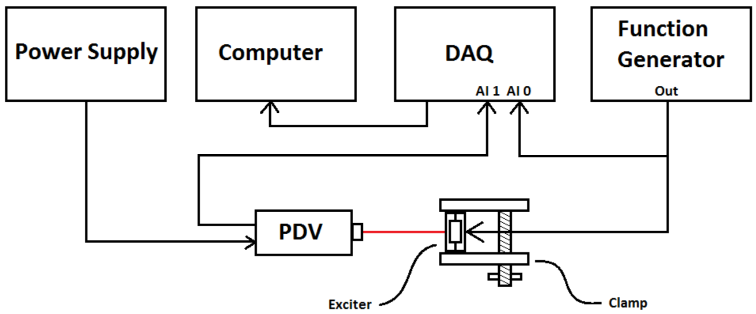



Figure 8: Exciter Output Experimental Setup

The PDV used was a Polytec Portable Digital Vibrometer, model PDV 100, serial number 113266. The function generator used was a Hewlett-Packard Function Generator, model 33120A, serial number US34016798. The acquisition equipment consisted of a National Instruments USB-4431 data acquisition chassis (NI USB-4431 DAQ), serial number 1915CC7, and NI Signal Express software. The data acquired for the exciter was taken at 44,100 samples/second to fully capture the 20-20,000 Hz spectrum of human hearing. A sine sweep was again chosen as the input signal from the function generator, which swept from 20 Hz to 20,000 Hz. The peak-to-peak voltage on the function generator was set to 4 V to stay within the maximum power handling of the exciter.

### 3.4. Sound Jury Procedures

The following section details the procedure used in the sound jury experiment to obtain the psychoacoustic characteristics of the sample button sounds, as well as their mapping to a set of vehicle command functions. In order to acquire reliable results from the sound jury, certain aspects were selected to ensure the results were not negatively skewed by subpar testing conditions [12]. All of the procedures used were approved by the Georgia Institute of Technology Institutional Review Board (IRB), which oversees all tests conducted at Georgia Tech involving human subjects. All IRB approved documents are provided in Appendix N.

Figure 9 shows a block diagram of the sound jury methodology. The experiment was conducted to identify the appropriateness of various command function button sound mappings as they relate to indicating successful initiation of the intended functions. This was done by having listeners rate their perception of button sounds based on several psychoacoustic metrics (part 1), and then rank what they believed to be the ideal button sound for a given set of command functions (part 2). The psychoacoustic metrics and command functions were then correlated to one another by comparing the results from parts 1 and 2.

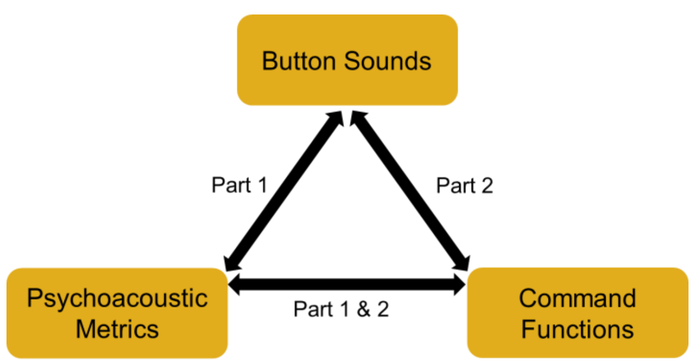



Figure 9. Sound Jury Methodology

### 3.4.1. Sound Samples Selection

Table 2 lists a brief description of the sample sounds used in the test. The subjects were not given the following descriptions; they were simply given the numbers, so as to not introduce a bias into the results (e.g. Sample 1, Sample 2, etc.). Plots of the time and frequency spectra of the sample sounds can be found in Appendix A.

Table 2. Sample Sound Descriptions

| Sound | Description | Type |
|---|---|---|
| Sample 1 | 2 kHz cue | Synthetic |
| Sample 2 | 150 Hz thump | |
| Sample 3 | Low falling tonal cue | |
| Sample 4 | Low rising tonal cue | |
| Sample 5 | High falling tonal cue | |
| Sample 6 | High rising tonal cue | |
| Sample 7 | Computer Keyboard click | Recorded |
| Sample 8 | Pack 11 switch 9 thump | |
| Sample 9 | Pack 5 switch 1 click | |
| Sample 10 | Pack 4 switch 2 click | |
| Sample 11 | Pack 3 switch 3 click | |
| Sample 12 | Pack 2 switch 1 click | |
| Sample 13 | Mouse click | |

The sample sounds used in the sound jury experiment were gathered from several different sources. Five of the thirteen sounds were taken from the results of the switch pack experiments, which are detailed in section 4.1. Another six of the thirteen sounds were created synthetically via MATLAB. The final two sounds were captured using an M-Audio Microtrack II professional 2-channel mobile digital recorder. The Microtrack II records high quality audio in stereo with 16 bits per sample, at 44,100 samples/second [14].

### 3.4.2. Sound Jury Room

Sound jury tests may utilize either live or pre-recorded sounds. Live sounds offer a more realistic approach, but are also more difficult to control due to the high number of variables and low degree of repeatability. Pre-recorded sounds offer the same experience for all jurors, therefore ensuring a highly repeatable test.

Sounds may be played via either loudspeakers or headphones. Loudspeaker playback introduces extra variables into the test which depend on the characteristics of the listening room and the quality of the signal. Headphone playback allows many issues regarding room acoustics to be ignored, though some issues must still be addressed. Noise isolation must be sufficient to ensure the subject does not have difficulty hearing the sounds. Sound playback must be done at a comfortable volume so as to neither jeopardize the hearing of the jurors nor force them to strain to hear sounds that are too quiet. The overall comfort of the jurors should be maximized by making the test room as natural an environment as possible. Lighting and air temperature should be adequate to allow subjects to perform all necessary tasks [13].

The original plan called for conducting the sound jury in a quiet office, due to its familiar environment, natural lighting, and comfortably adjustable air temperature. However, no suitable location could be found, so the hemi-anechoic chamber was chosen instead. The chamber has an added drawback of not being a natural environment as compared to an office, though this was deemed to not be of significant concern. The lighting in the chamber was satisfactory and the temperature was comfortable. The experiment was conducted on a laptop with external mouse, keyboard, and monitor on a table positioned in the center of the chamber. Each subject wore a pair of high quality over-ear headphones, which provided a flat frequency response and a high level of ambient noise reduction, making them optimal for the study.

### 3.4.3. Session Length

A sound jury test consists of several parts: reading and signing consent forms, administration of an audiogram hearing test, test training, and administration of the jury test. The time required to complete each section of the test should be estimated to identify how much time each subject will have to complete the full test, and thus the workload that should be presented to each subject for that test. The length of the sound jury test should be kept within 30 – 45 minutes to reduce the onset of subject fatigue. The number of sound samples and number of questions asked about those samples should be determined based on that time limit [13].

Each test was estimated to take about one hour to complete. This included reading and signing of consent forms, taking an audiogram, training, and the actual experiment. In actuality, all subjects finished in less than 45 minutes.

### 3.4.4. Subject Training

Each subject was administered an audiogram prior to the start of the test. The audiogram quantifies a person's hearing loss in terms of dB hearing level (dBHL). Subjects who fell outside the normal range of hearing (-10 to 15 dBHL) did not qualify for the study.

Before the start of the jury test, the subjects should be administered a training session to familiarize them with the questions being asked. Subjects should be presented with examples of how to complete each task, as well as explanations of any terminology, which the subject may be unfamiliar with. This should be done to ensure each subject is comfortable with the test [13].

For part 1, subjects were given a lay-term example of how to perform the experiment, which is shown below. The example used was for rating the flavor of a cup of coffee [10]. The purpose of the training was to allow the subjects to become familiar with the exact format of part 1 of the experiment and show how to provide their answers.

> As an example, a study created to determine the perceived strength for a cup of coffee might consist of a descriptive response scale similar to the one below:

| **Bold Flavor?** | Not at all | Slightly | Moderately | Very | Extremely | Don't Know |
|---|---|---|---|---|---|---|
| | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |

> You, the subject, will test the cup of coffee and then rate your impression of the strength, based on the graduated scale (not at all, slightly, moderately, …). If you feel that the cup of coffee was served to you far too bold, then the most appropriate response might be "Extremely" bold. If the cup that was served to you was not bold enough, then you might respond with "Not at all". If the cup of coffee is a little bold, then the descriptor "Moderately" may be the appropriate response for you to give in the survey. If you are not confident you understand what is meant by "Bold Flavor", selecting "Don't Know" may be the appropriate response.

For part 2, subjects were given a second lay-term example of how to perform the experiment, which is shown below. The example used was for matching the flavor of a cup of coffee to the season it would be most appropriate to drink it in. The purpose of the training was to allow the subjects to become familiar with the exact format of part 2 of the experiment and show how to provide their answers.

> As an example, a study created to determine the seasons in which customers prefer different flavors of coffee:
>
> | | |
> |---|---|
> | **Sample 1** | ___ Spring |
> | **Sample 2** | ___ Summer |
> | **Sample 3** | ___ Autumn |
> | **Sample 4** | ___ Winter |
>
> You, the subject, will test each cup of coffee and then match it with the season you believe it would be most desirable to drink it in. Each season must be assigned a sample, and you may assign a single sample to more than one season if you so choose. If you feel the second cup of coffee has flavors of pumpkin, then you may choose to pair it with the autumn season, and thus would write the number 2 next to "Autumn" in the right column. If you feel the third cup of coffee would be ideal to drink in autumn and summer, then you may write the number 3 next to both "Autumn" and "Summer" in the right column.

The subjects were also given definitions of each command function to ensure they were familiar with the terms they were being asked about. The command functions to be used in the study were provided by the sponsoring company. These definitions are shown below.

**Cruise**

1. <u>Set +</u>
   When cruise control is on and active, increases the set speed by 1 mile per hour
2. <u>Set -</u>
   When cruise control is on and active, decreases the set speed by 1 mile per hour
3. <u>On</u>
   Turns on cruise control.
4. <u>Off</u>
   Turns off cruise control.

**Phone**

5. <u>Answer</u>
   Answer incoming phone call
6. <u>End</u>
   End current phone call

**Media**

7. Media Mode

   Change the media source (CD, FM/AM, Aux, etc.)
8. Next Track

   Advance to the next media track
9. Previous Track

   Return to the previous media track
10. Play / Pause

    Play / pause the current media track
11. Mute - On

    Mute the current media track
12. Mute - Off

    Unmute the current media track
13. Volume Up

    Increase the media volume by one unit
14. Volume Down

    Decrease the media volume by one unit

**Voice**

15. Activate

    Begin listening for voice-issued commands

### 3.4.5. Experiment

The jury test itself consisted of two parts. Part 1 called for the subjects to listen to each of the sample sounds outlined in section 3.4.1 and answer a series of questions regarding the psychoacoustic metrics of those sounds. In part 2, the subjects listened to those same sounds again, and asked to match each one to a set of command functions. Details of each part are shown in the sections that follow.

For the test, a Google Survey was used as the data capture means to ask subjects questions and allow them to provide answers. A Google Survey is an internet-based form which can be customized to create unique questionnaires for a multitude of purposes. It automatically aggregates the data and stores it in spreadsheet form, eliminating the need to manually convert answers provided on written questionnaires to electronic format. This was done to facilitate quicker data analysis. Screenshots of the Google Survey can be found in Appendix L. A MATLAB graphical user interface (GUI) was used to present the sample sounds. The GUI provided the subjects with buttons which, when clicked, played a sample sound. The subjects were allowed to listen to each sound as many times as they wanted throughout the entire duration of the test. The MATLAB GUI code can be found in Appendix O. Screenshots of the GUI itself can be found in Appendix M.

*3.4.5.1. Part 1: Psychoacoustic Metrics*

A total of sixteen psychoacoustic metrics were used in part 1 of the sound jury test, which are show in Table 3. The metrics were provided by the sponsor company and were chosen to encompass a broad spectrum of sound characteristics. They include several antonym pairs such as *Long*/*Short* and *High Frequency*/*Low Frequency*.

Table 3. Psychoacoustic Metrics Used in Part 1 of the Sound Jury Test

| **Psychoacoustic Metrics** |
|---|
| *Long* |
| *Dark* |
| *Dynamic* |
| *Sharp* |
| *High Frequency* |
| *Intense* |
| *Expressive* |
| *Tonal* |
| *Short* |
| *Loud* |
| *Strong* |
| *Smooth* |
| *Low Frequency* |
| *Solid* |
| *Reliable* |
| *Powerful* |

Figure 10 shows a sample evaluation form for a single sound sample in part 1 of the test. The subjects were asked to rate how accurately they felt each metric described a given sample sound across a five point scale, ranging from *Not at all* to *Extremely*. A sixth point, labeled *Don't Know*, was included to allow the subjects to denote when they felt they did not adequately understand the question being asked of them. This was included to ensure the subjects were not simply guessing, and all *Don't Know* answers were excluded from statistical analysis.

## Sample 1

| | | | | | | |
|---|---|---|---|---|---|---|
| **Long?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Dark?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Dynamic?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Sharp?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **High Frequency?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Intense?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Expressive?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Tonal?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Short?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Loud?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Strong?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Smooth?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Low Frequency?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Solid?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Reliable?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |
| **Powerful?** | Not at all ☐ | Slightly ☐ | Moderately ☐ | Very ☐ | Extremely ☐ | Don’t Know ☐ |

Figure 10. Psychoacoustic Metric Sample Evaluation Form

*3.4.5.2. Part 2: Command Functions*

A total of fifteen command functions were used in part 2 of the sound jury test, which are shown in Table 4. The command functions were provided by the sponsor company and were chosen to encompass a broad spectrum of functions one might encounter in an automotive environment.

Table 4. Command Functions Used in Part 2 of the Sound Jury Test

| **Command Functions** |
|---|
| *Set +* |
| *Set -* |
| *On* |
| *Off* |
| *Answer Call* |
| *End Call* |
| *Media Mode* |
| *Next Track* |
| *Previous Track* |
| *Play/Pause* |
| *Mute On* |
| *Mute Off* |
| *Volume Up* |
| *Volume Down* |
| *Activate* |

Figure 11 shows a sample evaluation form for part 2 of the test. The subjects were asked to match each provided command function to what they considered to be the “ideal” sample sound for that particular function. They were allowed to assign a single sound to more than one command function if they believed it best represented more than one function, but were not allowed to assign multiple sounds to a single command function.

| Sounds | Cruise |
|---|---|
| Sample 1 | ___ Set + |
| Sample 2 | ___ Set - |
| Sample 3 | ___ On |
| Sample 4 | ___ Off |
| Sample 5 | **Media** |
| Sample 6 | ___ Media Mode |
| Sample 7 | ___ Next Track |
| Sample 8 | ___ Previous Track |
| Sample 9 | ___ Play / Pause |
| Sample 10 | ___ Mute On |
| Sample 11 | ___ Mute Off |
| Sample 12 | ___ Volume Up |
| Sample 13 | ___ Volume Down |
| | **Voice** |
| | ___ Activate |
| | **Phone** |
| | ___ Answer Call |
| | ___ End Call |

Figure 11. Command Function Sample Evaluation Form

### 3.4.6. Equipment Used

A VX Pocket 440 soundcard, serial number 2270.00020283 was installed in the laptop that was used to administer both the audiogram and jury test. The audiogram was administered with the Digital Recordings Digital Audiometer Screening software, version 6.8a. A pair of Sennheiser model HD 380 Pro headphones, serial number 502717 was used for both the audiogram and the jury test to allow the subjects to listen to the sounds. The headphones had a frequency respose of 8 – 27,000 Hz. The equipment used to calibrate the audiogram consisted of a Larson Davis model 824 sound level meter, serial number 1769, a Larson Davis model PRM902 pre amplifier, serial number 2268, and a Larson Davis ½" random incidence microphone, serial number 3344.

### 3.5. Sound Jury Statistical Analysis

#### 3.5.1. Normalization

For part 1 each qualitative response was given a numerical value in accordance with its magnitude (*Not at all* = 1, *Slightly* = 2, etc.). The only exception was the *Don't Know* option, which was omitted from all analyses. For each sample sound and each psychoacoustic metric, the average value for all responses was taken. For part 2, the total number of responses from all subjects was summed for each command function.

#### 3.5.2. Levene Test

A Levene test was conducted on the normalized data for all psychoacoustic metrics and command functions in Microsoft Excel using the Real Statistics resource pack [15]. The test was used to determine the homogeneity of variance of the data sets, to discern whether further parametric or non-parametric statistical tests were most appropriate. Due to the nature of the data, separate tests were conducted for each psychoacoustic metric, but only once for all command functions. There are three ways to conduct a Levene test: using the mean, using the median, or using the 10% trimmed mean. All three methods were used in the following analyses. A P-value above $\alpha = 0.05$ indicates equal variance, and therefore parametricity for that particular metric [16].

#### 3.5.3. Analysis of Variance (ANOVA)

An ANOVA test was conducted in Excel using the Real Statistics resource pack on the normalized data for the parametric data sets (i.e. those that passed the Levene test)

[15]. It is a parametric test that may be used when the assumption that the data sets have equal variances is valid. The test was used to determine whether or not the subjects' responses were random, as well as whether or not the sample sounds used were noticeably different from each other. Due to the nature of the data, separate tests were conducted for each psychoacoustic metric, but only once for all command functions. The values of interest in an ANOVA test are the F statistic, $F_{crit}$, and P-value. For a given metric, An F statistic above $F_{crit}$ demonstrates that the subjects were able to discern that all the sound samples were not simply the same sound played over and over. A P-value below $\alpha = 0.05$ implies with 95% certainty that the distributions for each metric did not happen by chance [16].

### 3.5.4. Kruskal-Wallis Test

A Kruskal-Wallis test was conducted in Microsoft Excel using the Real Statistics resource pack on the normalized data for the non-parametric data sets (i.e. those that did not pass the Levene test) [15]. It is a non-parametric test that may be used in place of an ANOVA test when the assumption that the data sets have equal variances is not valid. Due to the nature of the data, separate tests were conducted for each psychoacoustic metric, but only once for all command functions. A P-value below $\alpha = 0.05$ implies with 95% certainty that the distributions for each metric did not happen by chance [16].

### 3.5.5. Correlation Coefficient

Two correlation coefficients were calculated in order to find the correlation between psychoacoustic metrics and command functions. Pearson product moment correlation coefficients were calculated for parametric metrics (i.e. those which were found by the Levene test to have equal variances). Spearman's rank order correlation coefficients were calculated for non-parametric metrics (i.e. those which were found by the Levene test to have unequal variances). Both analyses were done in Microsoft Excel using the Real Statistics resource pack [15]. The critical correlation coefficient for each metric was determined by the alpha value of $\alpha = 0.05$ and the number of responses for that metric. For all command functions, which all had $n = 31$ responses, a critical correlation coefficient of ±0.355 was chosen. However, due to the fact that the *Don't Know* responsese were discarded from the part 1 of the soud jury, the critical correlation coefficient for each psychoacoustic metric varied between ±0.355 and ±0.468. All correlation values smaller in magnitude than the critical value were discarded as they are considered to have statistically happened by chance. All correlation values higher in magnitude than the critical value were considered significant [16].

### 3.5.6. Post Hoc T-test

A T-test was performed in Microsoft Excel using the TTEST function to compare the results from each combination of sample sounds across all parametric data sets (i.e. those that passed the Levene test). It was used to determine whether the difference between a given pair of sample sounds was significant, based on each sample's mean and variance. All values above $\alpha = 0.05$ are considered significant. The number of tails was set to 2, and the matched pairs method was used, as the sound jury has a within-subjects design [16].

### 3.5.7. Post Hoc Wilcoxon Signed Ranks Test

A Wilcoxon signed-ranks test was performed in Microsoft Excel using the Real Statistics resource pack to compare the results from each combination of sample sounds across all non-parametric data sets (i.e. those that did not pass the Levene test) [15]. It was used to determine whether the difference between a given pair of sample sounds was significant, based on each sample's mean and variance. All values above $\alpha = 0.05$ are considered significant. The number of tails was set to 2, and the matched pairs method was used, as the sound jury has a within-subjects design [16].

#### 3.5.8. Sample Ranking

The thirteen sample sounds were ranked in terms of popularity from 1 to 13 (1 being most popular, 13 being least popular). To do so, the sample sounds were first ranked for each individual command function. Each ranking was provided a weighting in accordance with its popularity, and these scores were summed for each sample sound across all command functions. The higher a sound sample's total score, the greater its popularity, and therefore the more ideal that sound is overall for being matched to all command functions.

#### 3.5.9. Command Function / Sample Sound Grouping

Because there may exist scenarios where it is either not desirable or not feasible to assign a unique sound to each individual command function, it was deemed beneficial to identity groups of command functions that could each be assigned to a single sample sound. For example, if it is desired to use $n$ number of sounds, then the fifteen command functions need to be assigned to one of the $n$ groups. This would be beneficial in a scenario where hardware constraints limit the number of sounds that are able to be used.

Which command functions are placed in which groups depends on their relative correlation coefficients. Each group was assigned an ideal sample sound from the thirteen used in the study which had a high popularity ranking for each command function within that group. Next, average correlation coefficients were taken for each group. Ideally, each group should have as high an average correlation coefficient as possible.

# CHAPTER 4. RESULTS AND DISCUSSION

## 4.1. Switch Pack Audio Response Measurement

The following section presents the results from the measurement of the audio signatures of the switch packs provided by the sponsor company. Discussion of which signals ended up being used in the sound jury test is provided in section 4.4.

### 4.1.1. Data Analysis

The audio response of each switch was obtained for both the time and frequency domains. A Fast Fourier Transform (FFT) was performed on each of the signals to convert the signals from the time to frequency domain. All analyses used a sampling rate of $f_s$ = 44,100 Hz to fully capture the spectrum of human hearing. A high pass filter was applied at 80 Hz to remove frequencies below the lower limit of the hemi-anechoic chamber. The MATLAB code used to achieve this can be found in Appendix O. The normalized amplitude vs. time and normalized amplitude vs. frequency plots of each switch can be found in Appendix K.

### 4.1.2. Signal-to-Noise Ratio

Table 5 shows the signal-to-noise ratio (SNR) for each switch. The sound pressure of the switch audio response and the background noise were compared, and the total SNR was calculated over the whole frequency spectrum. It can be seen that all values are above 25 dB. Thus, no correction factor is needed.

Table 5. Switch Pack Audio Response SNR

| Pack | Switch | $SNR_{total}$ (dB) | Pack | Switch | $SNR_{total}$ (dB) |
|---|---|---|---|---|---|
| 1 | 1 | 52.6 | 7 | 1 | 40.6 |
| | 2 | 60.0 | | 2 | 43.7 |
| | 3 | 40.3 | | 3 | 44.2 |
| | 4 | 37.6 | | 4 | 44.9 |
| | 5 | 46.0 | | 5 | 41.7 |
| | 6 | 51.1 | | 6 | 46.4 |
| | 7 | 50.3 | 8 | 1 | 36.7 |
| 2 | 1 | 47.6 | | 2 | 47.9 |
| | 2 | 37.9 | | 3 | 43.4 |
| | 3 | 47.4 | | 4 | 48.0 |
| | 4 | 60.4 | | 5 | 36.8 |
| | 5 | 66.4 | | 6 | 38.0 |
| 3 | 1 | 44.2 | 9 | 1 | 51.1 |
| | 2 | 39.2 | | 2 | 33.6 |
| | 3 | 47.4 | | 3 | 25.6 |
| | 4 | 42.1 | | 4 | 40.0 |
| | 5 | 37.8 | | 5 | 52.6 |
| 4 | 1 | 44.9 | 10 | 1 | 51.6 |
| | 2 | 46.3 | | 2 | 47.7 |
| | 3 | 41.8 | | 3 | 54.4 |
| | 4 | 40.3 | | 4 | 52.5 |
| 5 | 1 | 43.5 | | 5 | 60.4 |
| | 2 | 36.6 | | 6 | 52.6 |
| | 3 | 47.2 | | 7 | 53.6 |
| | 4 | 46.7 | | 8 | 49.5 |
| | 5 | 45.5 | 11 | 1 | 47.2 |
| | 6 | 39.5 | | 2 | 43.0 |
| | 7 | 42.4 | | 3 | 43.8 |
| 6 | 1 | 40.9 | | 4 | 41.5 |
| | 2 | 34.6 | | 5 | 42.5 |
| | 3 | 47.1 | | 6 | 40.6 |
| | 4 | 45.9 | | 7 | 43.6 |
| | | | | 8 | 53.1 |
| | | | | 9 | 44.4 |

## 4.2. Haptic Board Output Measurement

The voltage output for all the listed configurations of the haptic board was obtained for both the time and frequency domains. An FFT was performed on each of the time domain signals to achieve the frequency spectrum. All analyses again used a sampling rate of $f_s$ = 44,100 Hz, and a band pass filter of 10 – 5512.5 Hz, which was chosen so as to mimic the operating frequency range of the haptic board. The MATLAB code used to achieve this can be found in Appendix O. The normalized amplitude vs. time and normalized amplitude vs. frequency plots of each signal can be found in Appendix I.

Figure 12 – Figure 19 show the calculated transfer functions of the sine-sweep signal measured at each gain setting under the no-load impedance condition. Comparing Figures 11 and 12, there is a significant increase in the noise of the signal, such that the link between two spectra becomes virtually unrecognizable. The only difference between these two transfer functions is the volume setting, which is 35 in Figure 12 and 36 in Figure 13. The same comparison can be said for Figures 13 – 14, 15 – 16, and 17 – 18. Thus, regardless of gain setting, 35 is the highest possible volume setting before distortion occurs. Furthermore, the transfer function measurement revealed no significant resonance frequencies, as none of the figures show any significantly large peaks.

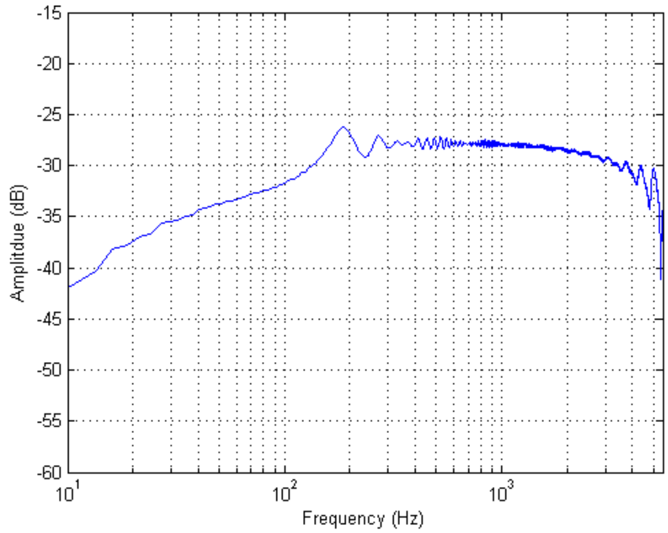



Figure 12. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 35 Volume, No-load Impedance Condition

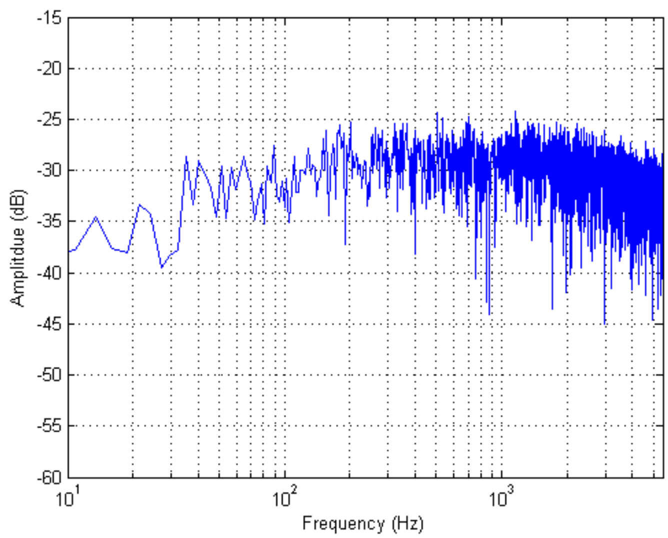



Figure 13. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 36 Volume, No-load Impedance Condition

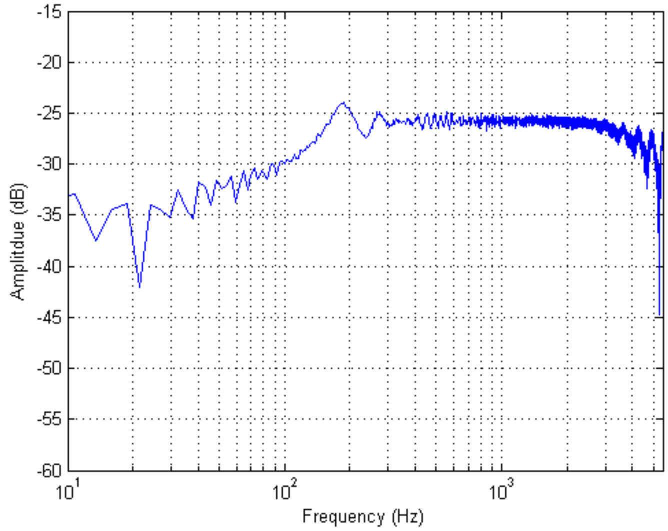



Figure 14. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 35 Volume, No-load Impedance Condition

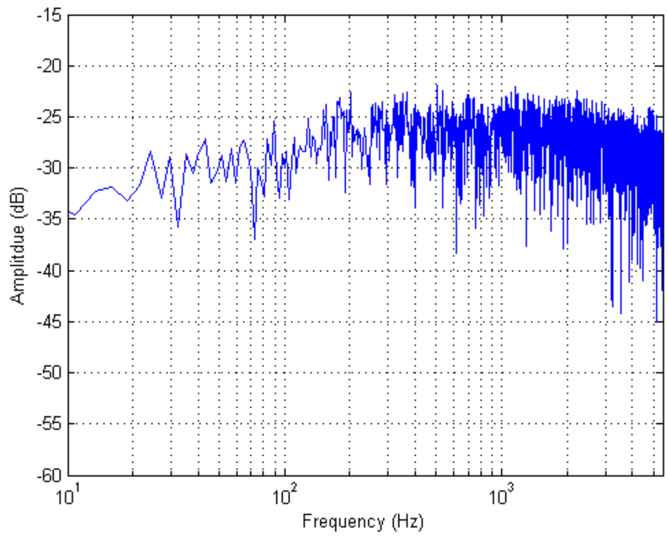



Figure 15. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 36 Volume, No-load Impedance Condition

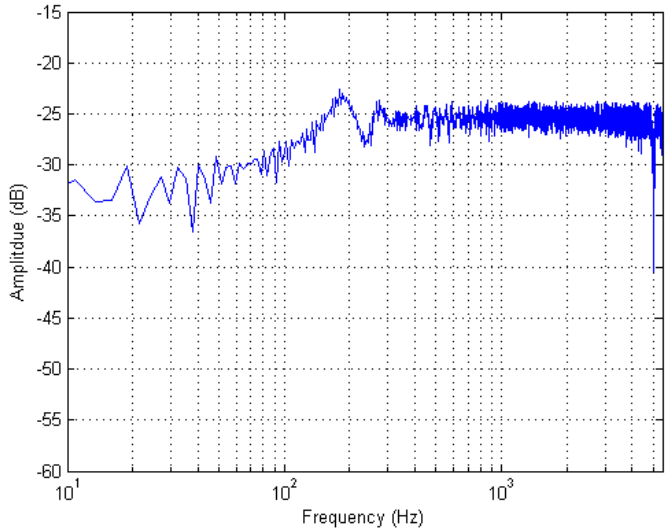



Figure 16. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 35 Volume, No-load Impedance Condition

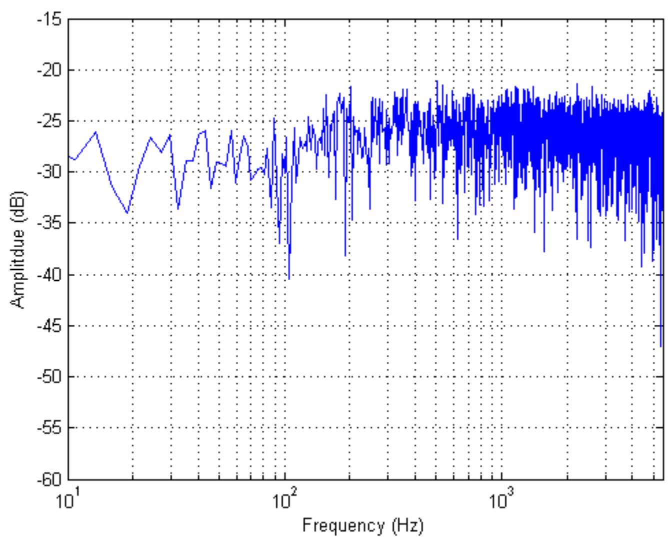



Figure 17. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 36 Volume, No-load Impedance Condition

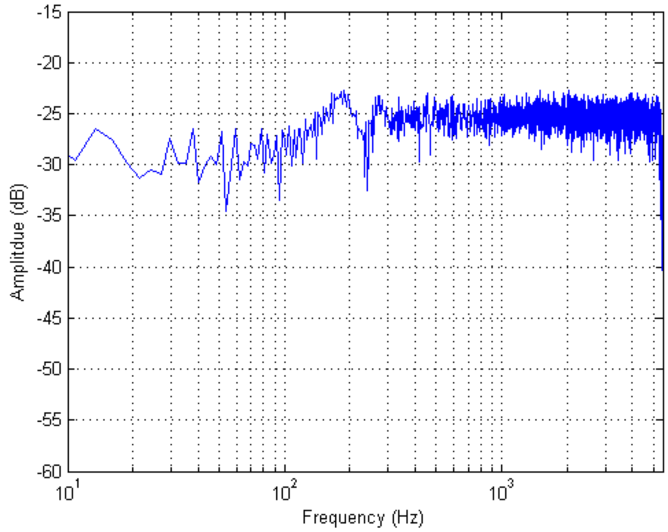



Figure 18. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 24 dB gain, 35 Volume, No-load Impedance Condition

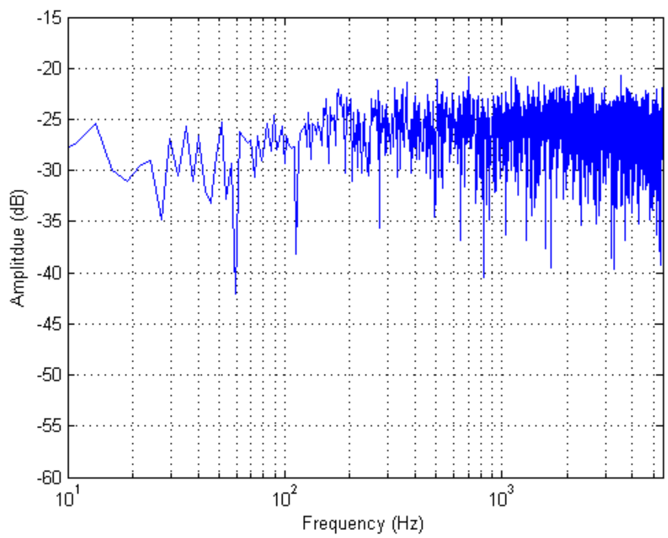



Figure 19. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 24 dB gain, 36 Volume, No-load Impedance Condition

Figure 20 – Figure 25 show the calculated transfer functions of the sine-sweep signal measured at 6 dB gain under each of the three impedance conditions. Once again, comparing Figures 19 – 20, 21 – 22, and 23 – 24 confirms that distortion occurs above a volume setting of 35. Additionally, comparing Figures 19, 21, and 23 reveals virtually no

change in the shape of the transfer function as the impedance condition is changed. Therefore, the impedance condition plays no significant role in the behavior of the transfer function.

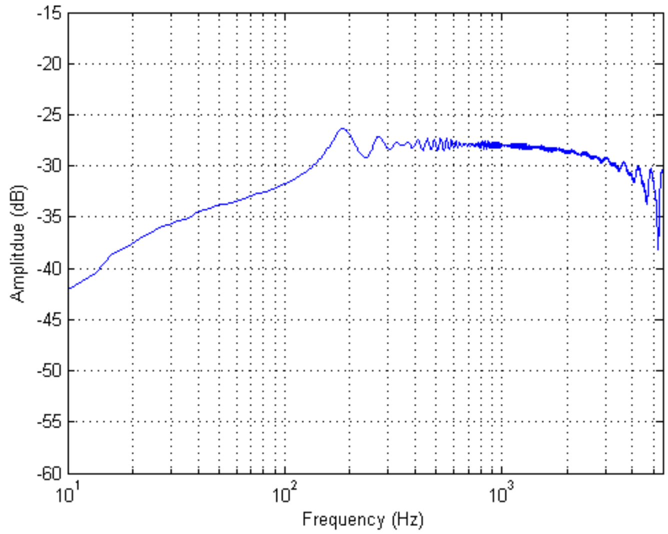



Figure 20. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 35 Volume, Exciter loaded Impendence Condition

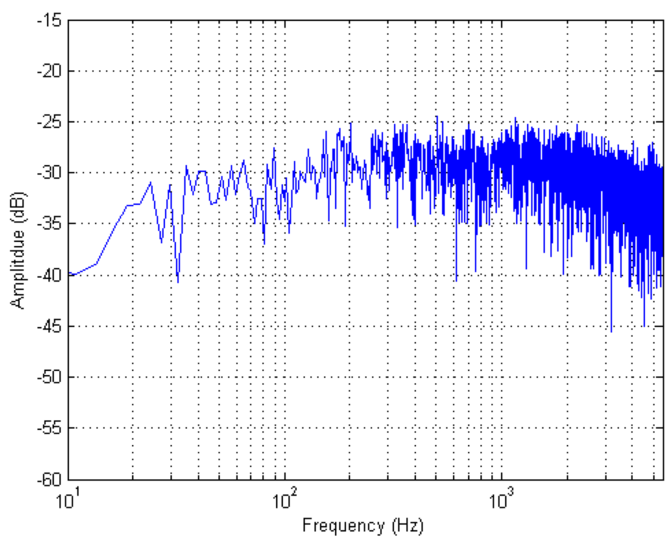



Figure 21. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 36 Volume, Exciter loaded Impendence Condition

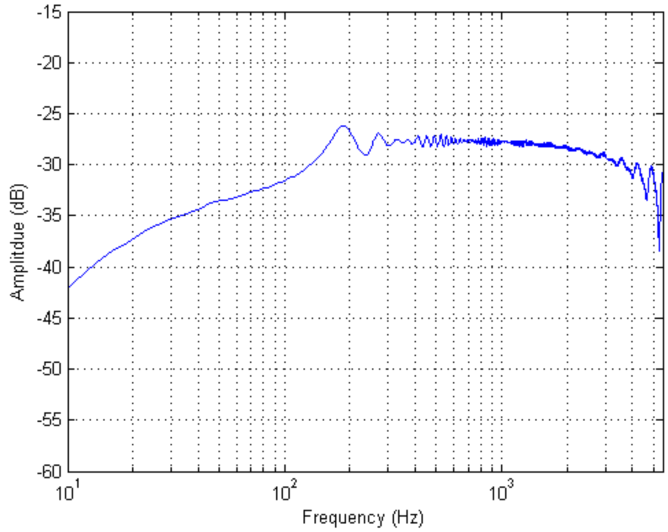



Figure 22. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 35 Volume, Resistor loaded Impendence Condition

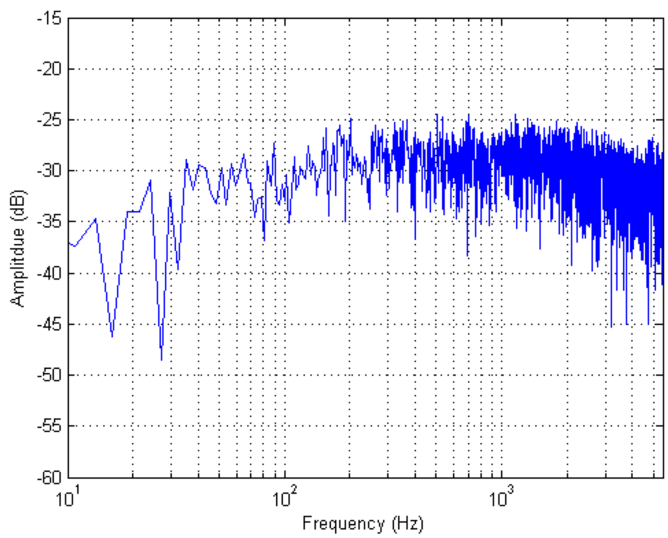



Figure 23. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 36 Volume, Resistor loaded Impendence Condition

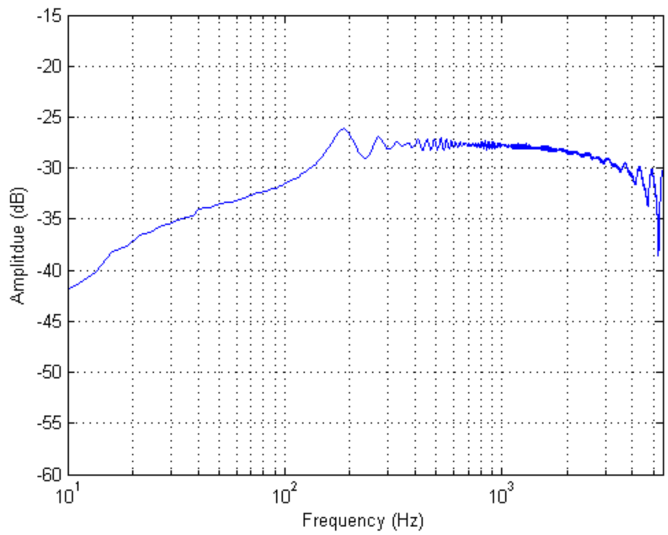



Figure 24. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 35 Volume, No-load Impedance Condition

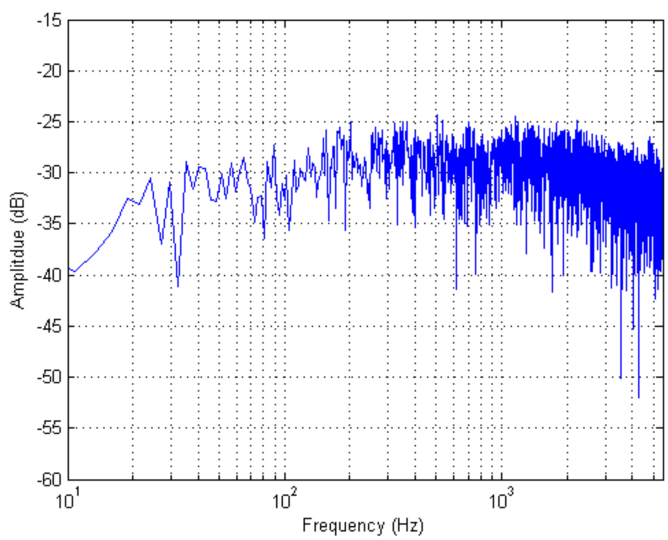



Figure 25. Haptic Board Voltage Output Transfer Functions – Sine Sweep, 6 dB gain, 36 Volume, No-load Impedance Condition

At all gain settings, transfer functions were clean and linear for all volume settings at or below 35 (Figures 19, 21, and 23). However, volumes of 36 and above resulted in highly nonlinear transfer functions (Figures 20, 22, and 24), and frequency spectra that appeared as noise, regardless of input signal. A clean sine sweep or pure tone input would result in a clean sine sweep or pure tone output at 1 – 35 volume, but would manifest as broadband noise at 36 – 255 volume. This change at 35/36 volume points to a problem in the internal signal processing of the haptic board itself, rather than harmonic distortion due to the mechanical properties of the system.

### 4.3. Exciter Output Measurement

The voltage output of the exciter for all the listed configurations was obtained for both the time and frequency domains. An FFT was performed on each of the time domain signals to achieve the frequency spectrum. All analyses again used a sampling rate of $f_s$ = 44,100 Hz, and a band pass filter of 10 – 10,000 Hz, which was chosen so as to mimic the operating frequency range of the exciter. The MATLAB code used to achieve this can be found in Appendix O. The normalized amplitude vs. time and normalized amplitude vs. frequency plots of each signal can be found in Appendix J.

Figure 26 shows the calculated transfer function of the exciter. A resonance frequency was identified at about 190 Hz, as well as a second, smaller resonance frequency around 1725 Hz, which are recognized by the significantly large peak and smaller peak at each of those frequencies, respectively. The second resonance frequency may be considered negligible since it has a significantly smaller magnitude that the first.

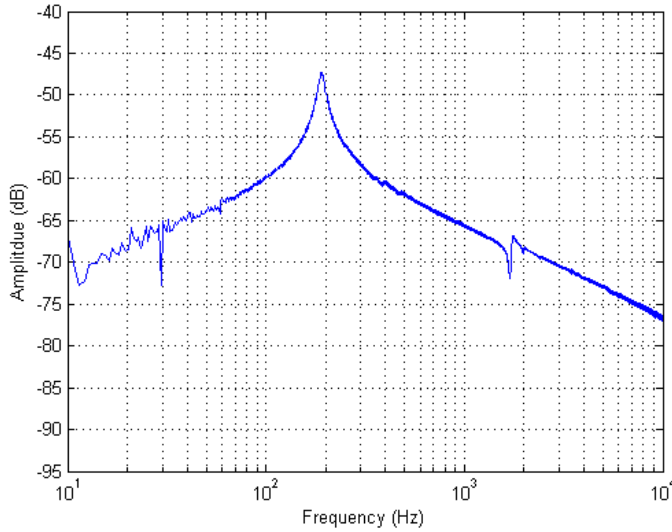



Figure 26. Exciter Transfer Function

## 4.4. Sound Jury

The following section presents the results from the conducted sound jury test. Results from both the psychoacoustic metric mapping and command function matching segments of the test are shown below.

### 4.4.1. Subject Demographics

The sound jury consisted of a total of thirty-one test subjects: twenty of which were male and eleven of which were female. As the jury included only Georgia Tech students, all subjects were in the age range of 18 – 24.

### 4.4.2. Normalization

#### *4.4.2.1. Part 1: Psychoacoustic Metrics*

Figure 27 – Figure 39 show the responses from part 1 of the test for each of the thirteen sample sounds. For a given plot, each bar is the average value of all the responses given for that metric, for that particular sound sample. The added standard deviation bars show the standard deviation for that particular psychoacoustic metric across all thirteen

sample sounds, not for that particular sample sound across all sixteen psychoacoustic metrics. Several of the sounds exhibited similar psychoacoustic characteristics. For example, Figure 27 and Figure 32 show that sample 1 & 6 were both considered to be very *sharp* and *high frequency*. Figure 29 and Figure 30 show that sample 3 & 4 were considered to be very *dynamic*. Conversely, multiple sounds also had several metrics in common for which there was highly negative association. The majority of the sounds were found to be neither *long* nor *dark*. Figure 27 and Figure 32 show that sample 1 & 6 were found to not be very *smooth*. Several of these findings follow logical reasoning, such as how the same sounds that were identified as very *short* were also identified as not being very *long*, as one would expect. The implications of these similarities among the psychoacoustic metrics of multiple sounds are discussed further in later sections.

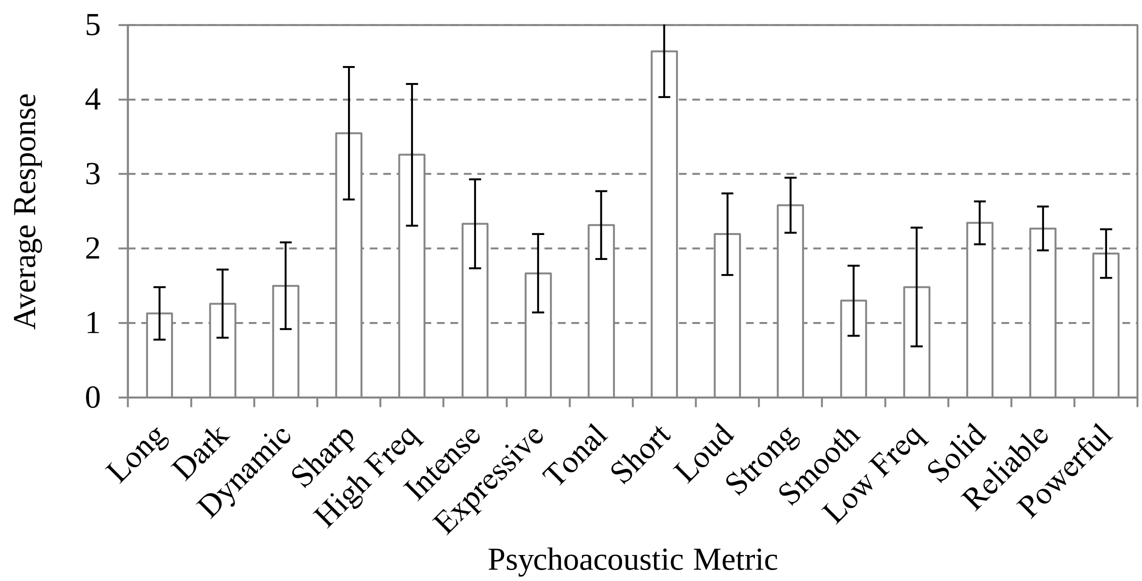



Figure 27. Response vs. Psychoacoustic Metric: Sample 1. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

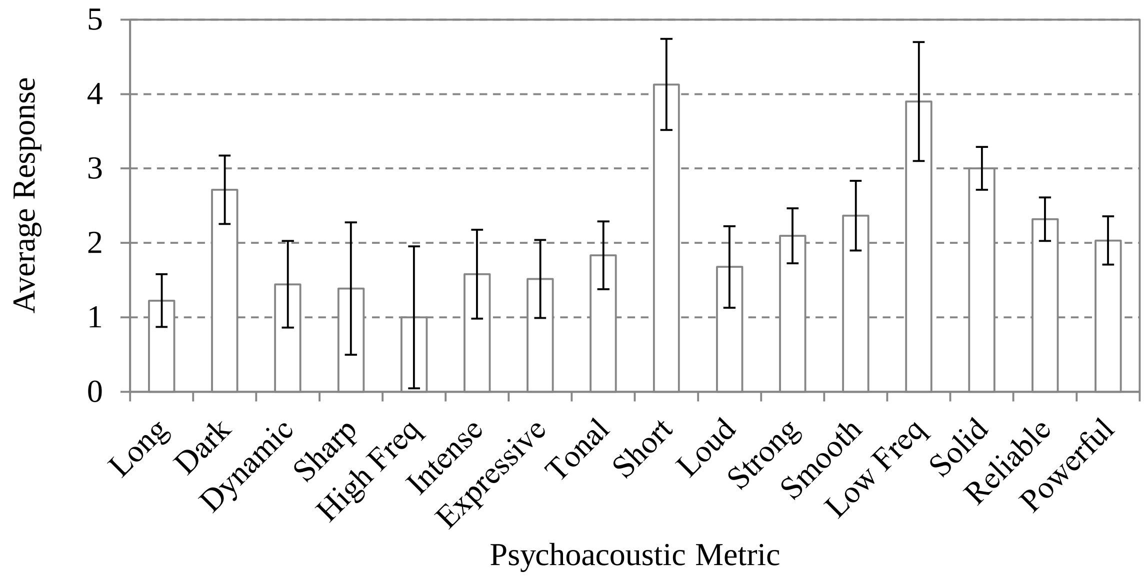



Figure 28. Response vs. Psychoacoustic Metric: Sample 2. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

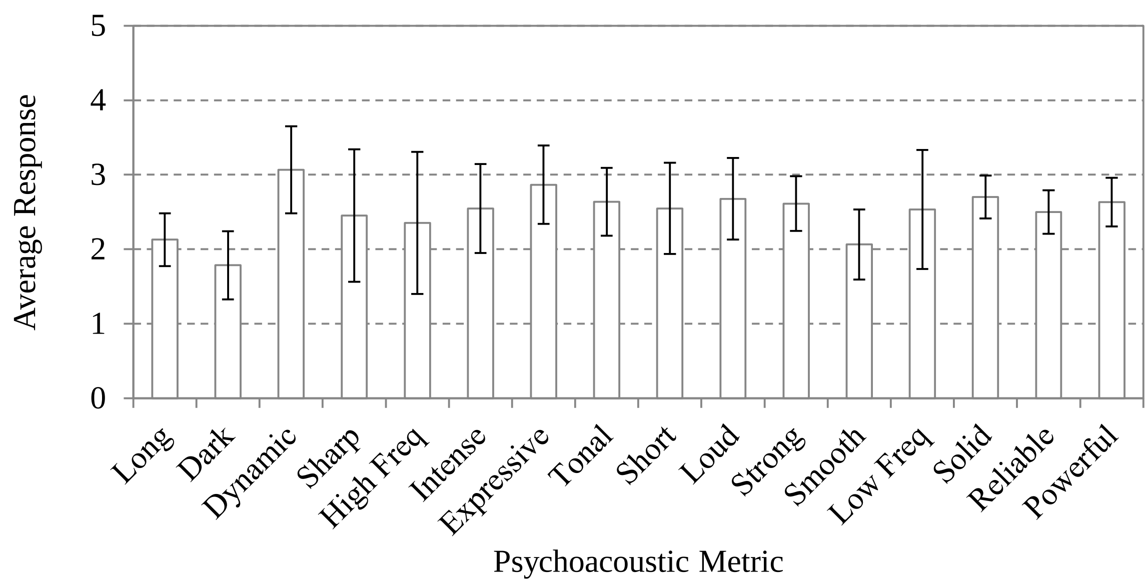



Figure 29. Response vs. Psychoacoustic Metric: Sample 3. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

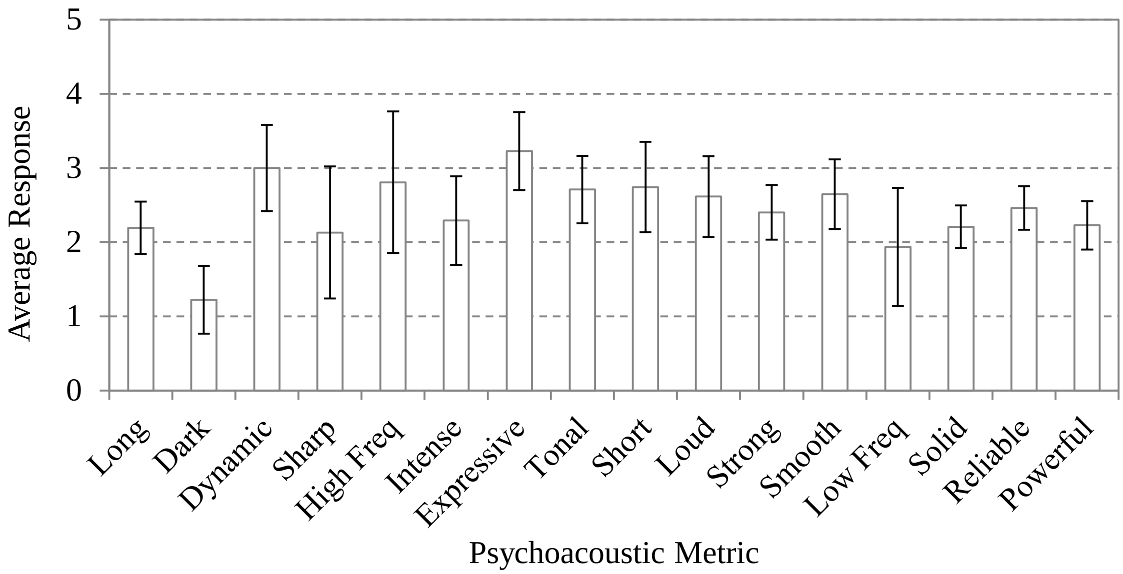



Figure 30. Response vs. Psychoacoustic Metric: Sample 4 . The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

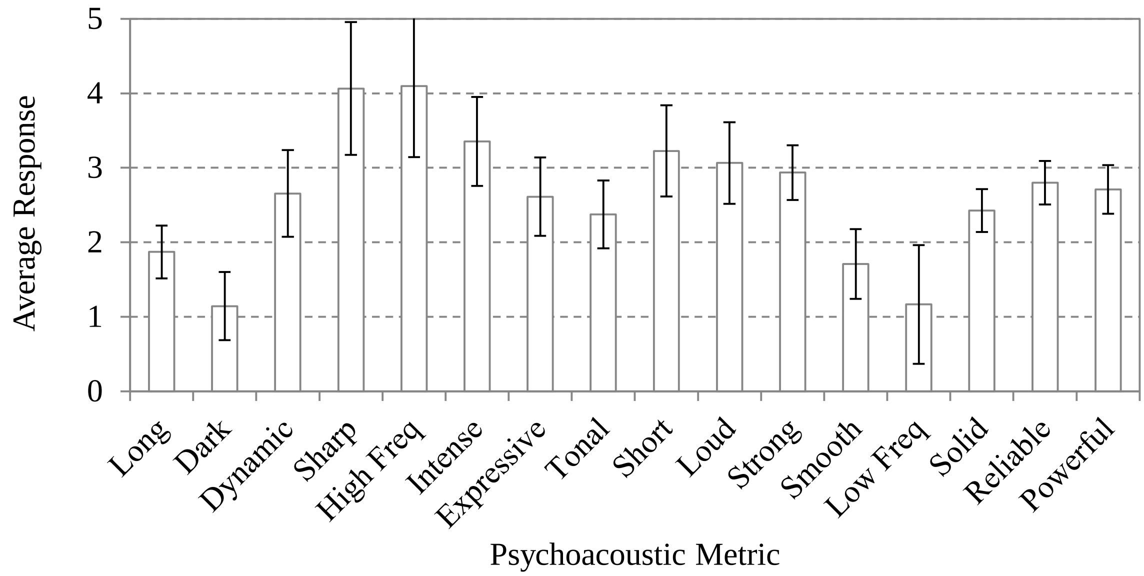



Figure 31. Response vs. Psychoacoustic Metric: Sample 5. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

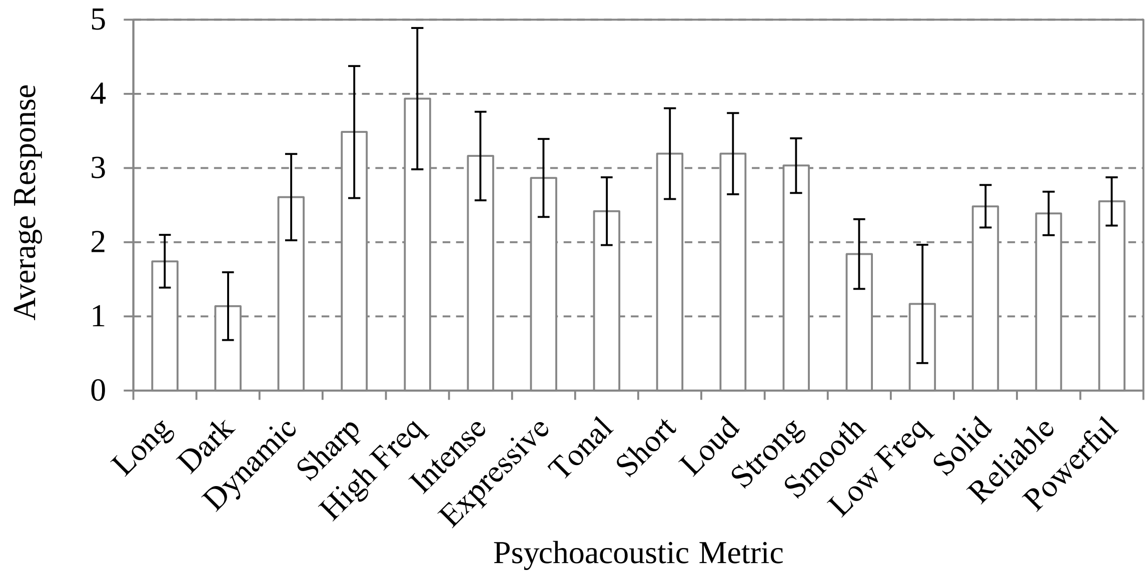



Figure 32. Response vs. Psychoacoustic Metric: Sample 6. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

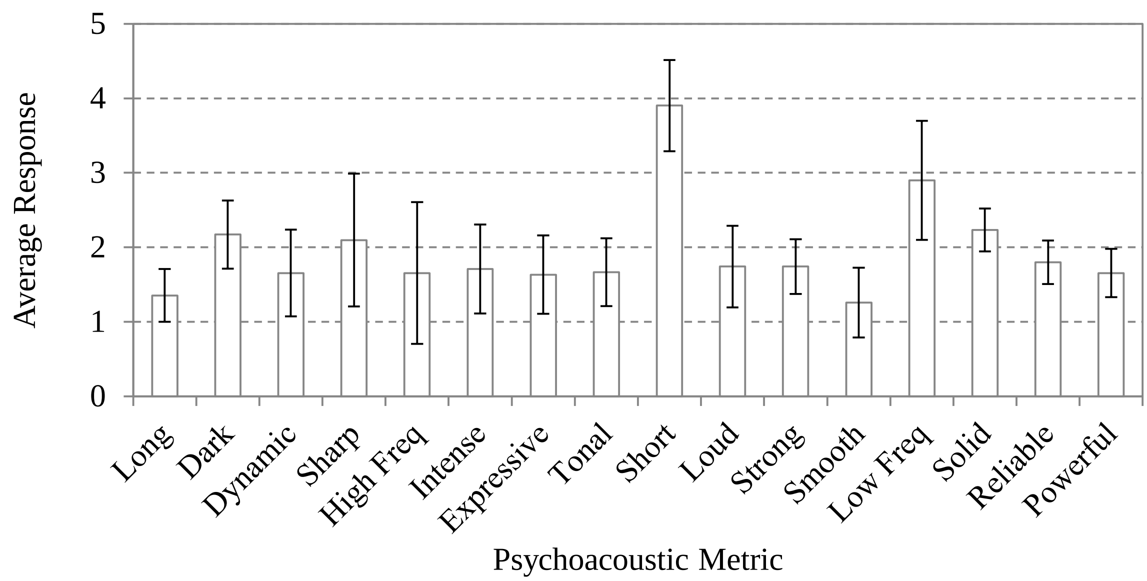



Figure 33. Response vs. Psychoacoustic Metric: Sample 7. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

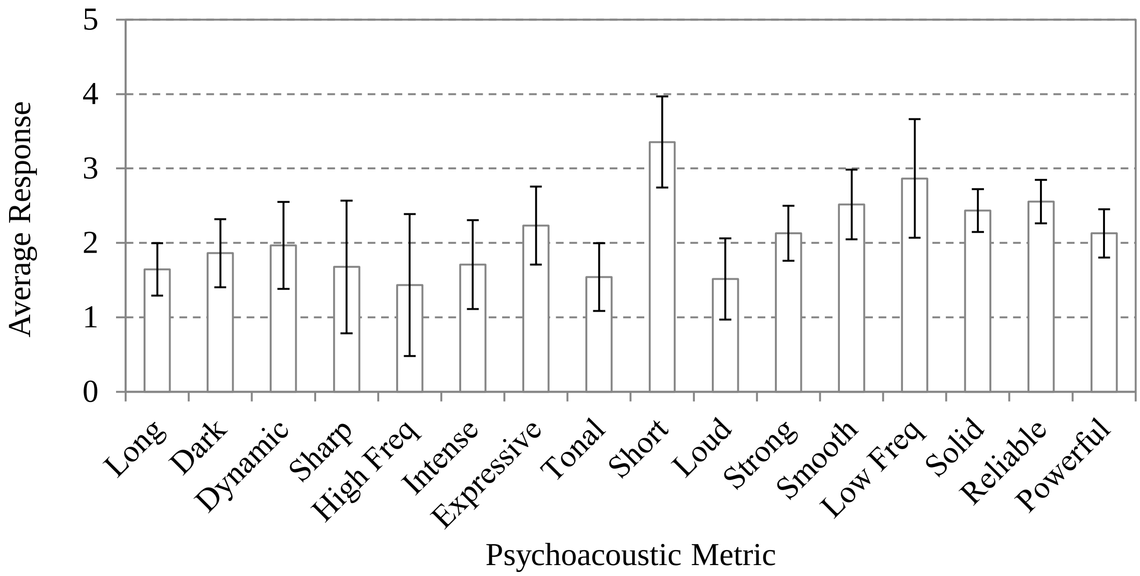



Figure 34. Response vs. Psychoacoustic Metric: Sample 8 . The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

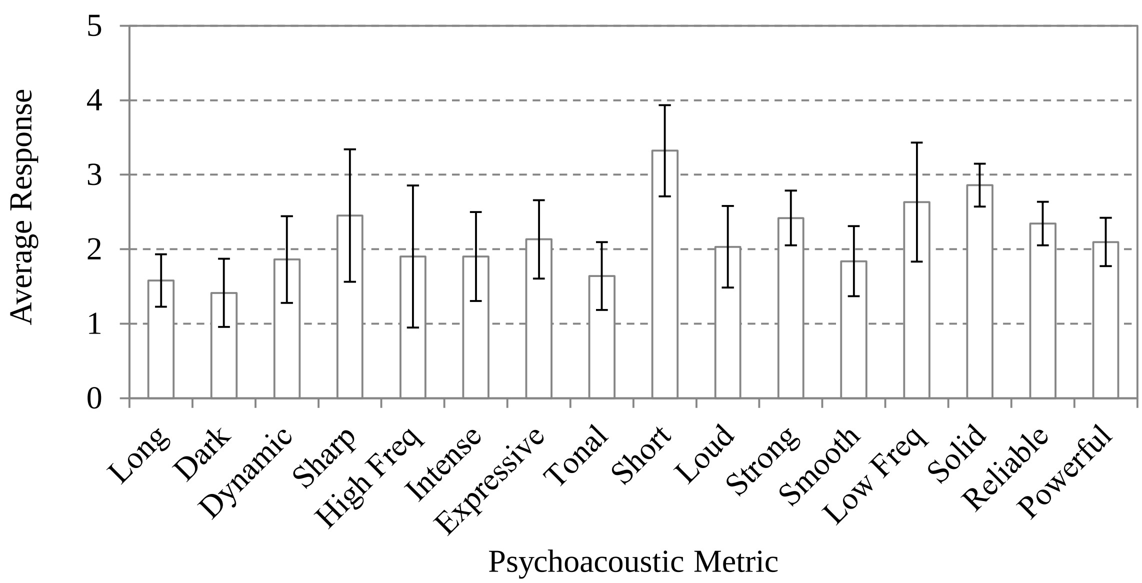



Figure 35. Response vs. Psychoacoustic Metric: Sample 9. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

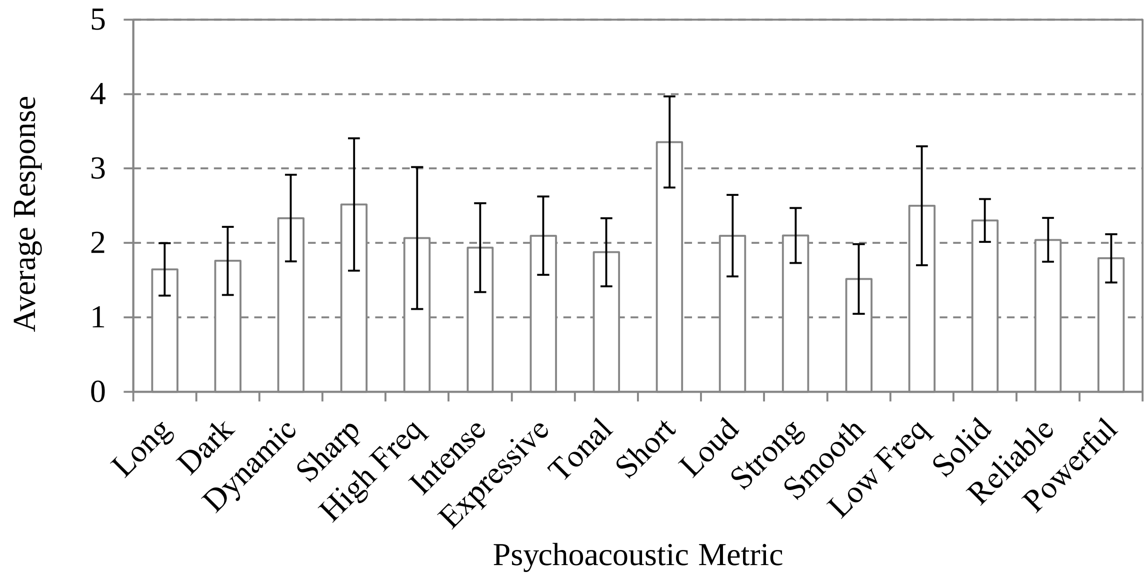



Figure 36. Response vs. Psychoacoustic Metric: Sample 10. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

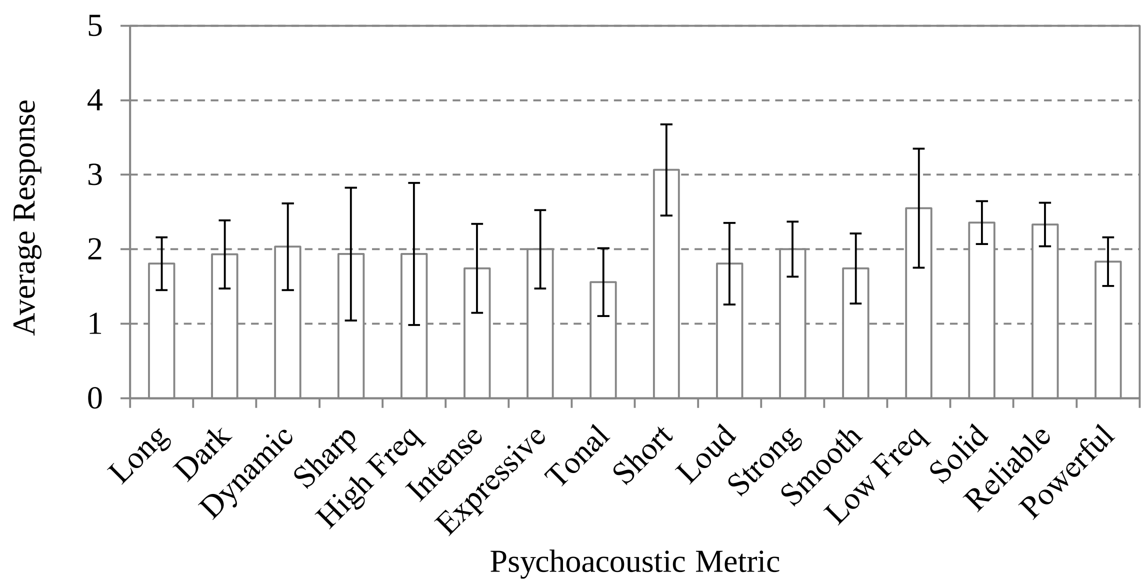



Figure 37. Response vs. Psychoacoustic Metric: Sample 11. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

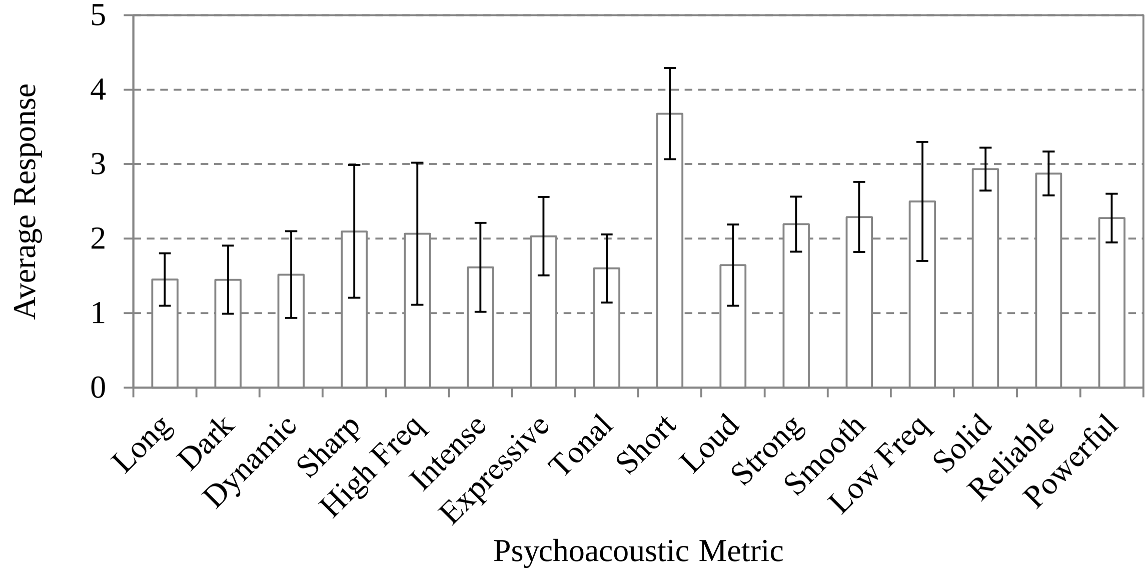



Figure 38. Response vs. Psychoacoustic Metric: Sample 12. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

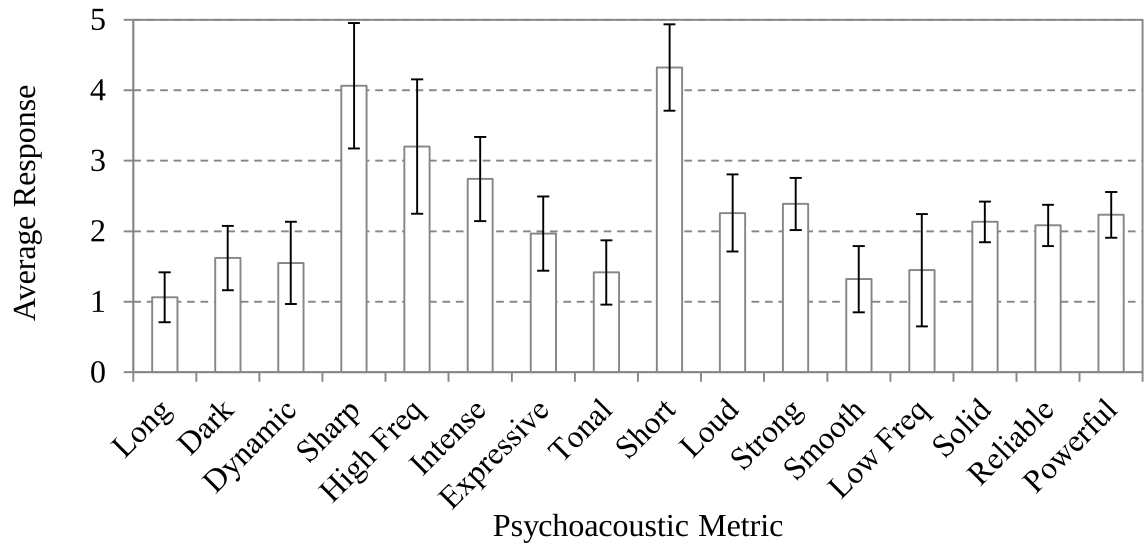



Figure 39. Response vs. Psychoacoustic Metric: Sample 13. The Response Scale Ranges From 1 = *Not At All* to 5 = *Extremely*.

*4.4.2.2. Part 2 – Command Functions*

Figure 40 – Figure 52 show the responses from part 2 of the test for each of the thirteen sample sounds. For a given plot, each column is the total number of responses

given for that function, for that particular sound sample. The added standard deviation bars show the standard deviation for that particular command function across all thirteen sample sounds, not for that particular sample sound across all fifteen command functions. Several of the sounds were found to match well with similar command functions. For example, Figure 40 and Figure 45 show that sample 1 & 6 both match well with the *set +* function. Figure 41 and Figure 51 show that sample 2 & 12 both match well with the *volume down* function. The implications of these similarities among the command functions matched with multiple sounds is discussed further in later sections.

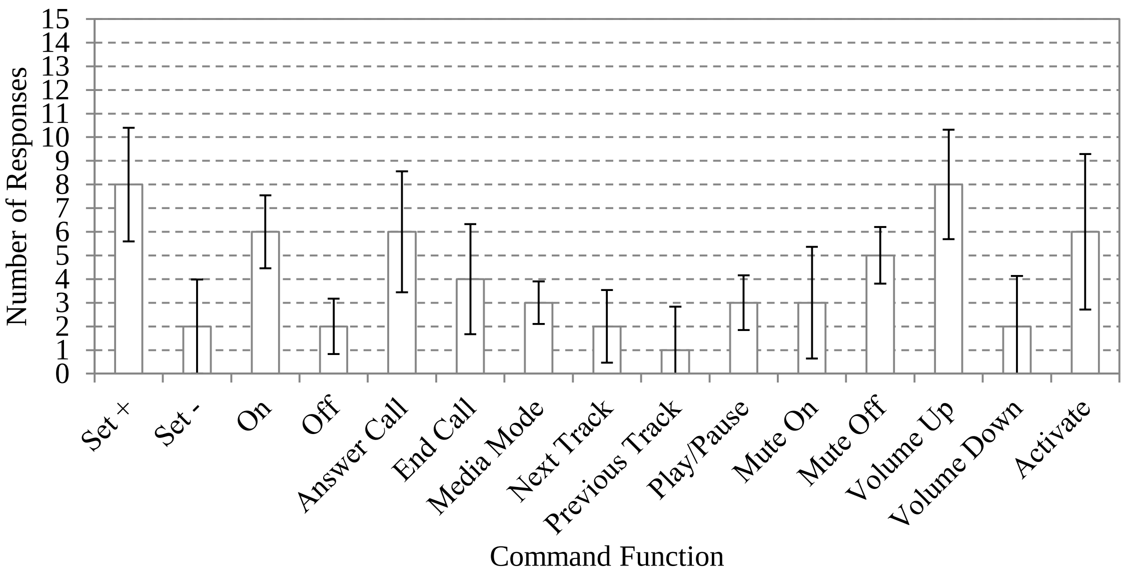



Figure 40. Response vs. Command Function: Sample 1

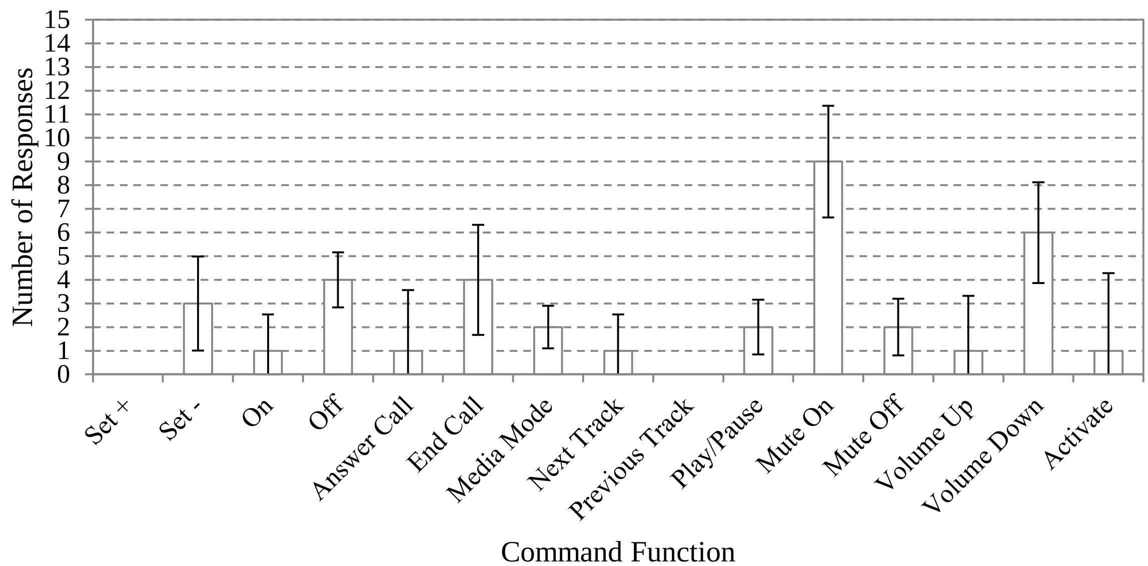



Figure 41. Response vs. Command Function: Sample 2

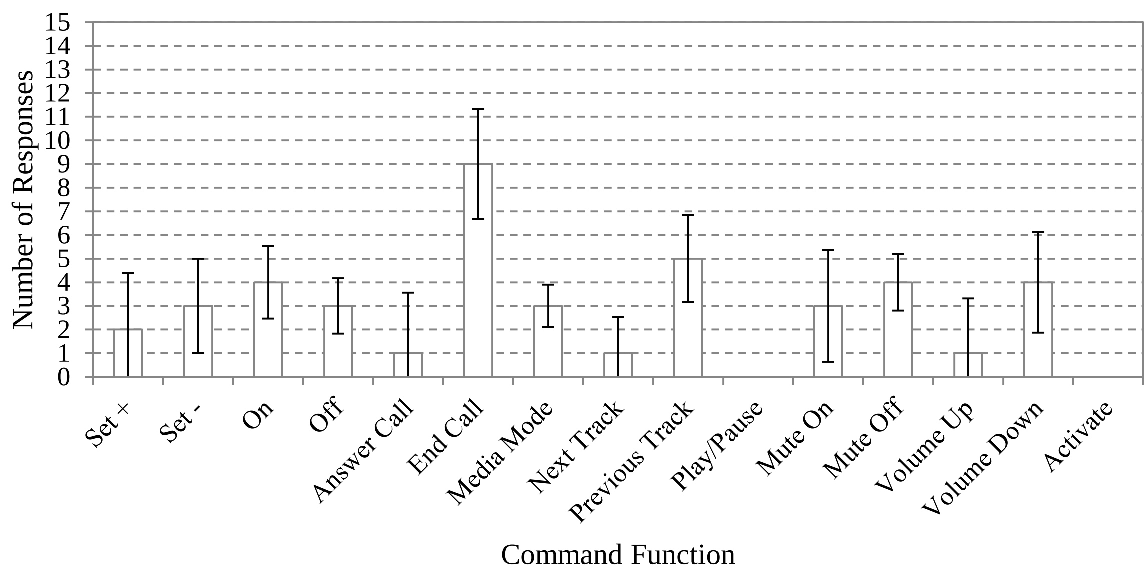



Figure 42. Response vs. Command Function: Sample 3

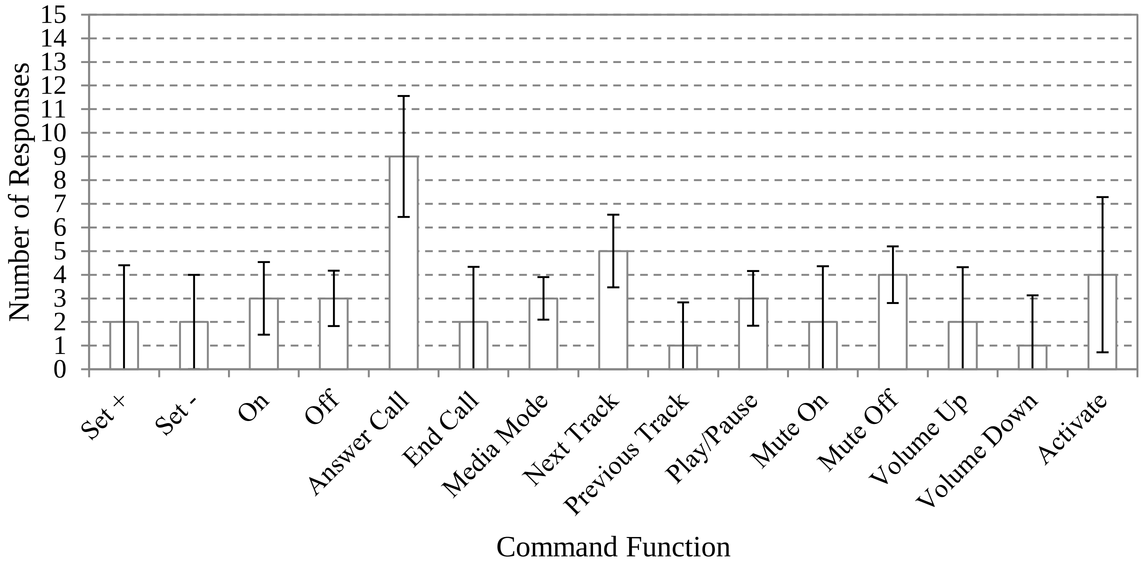



Figure 43. Response vs. Command Function: Sample 4

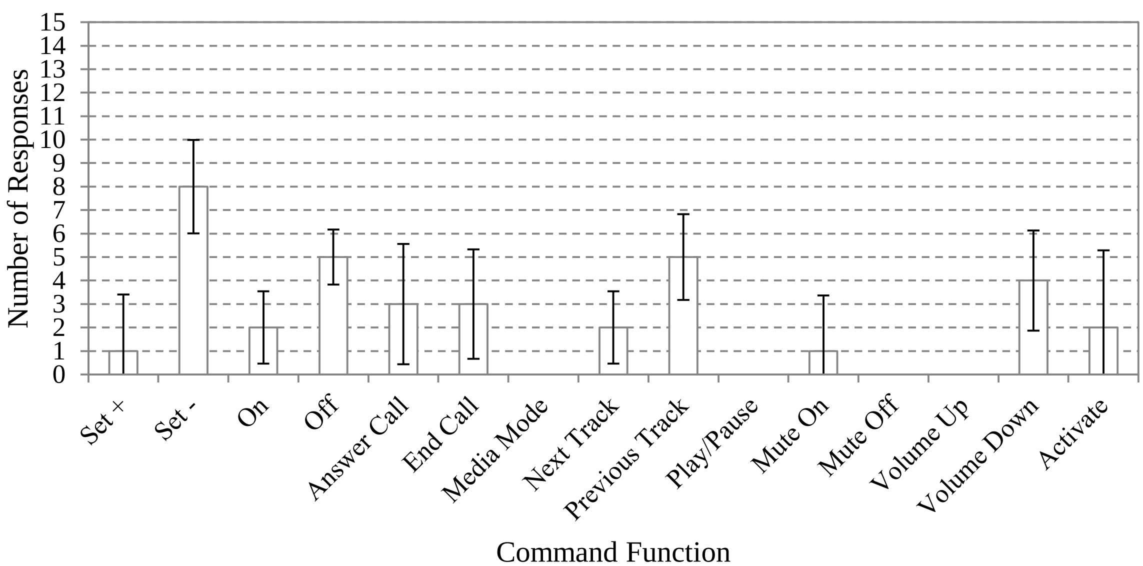



Figure 44. Response vs. Command Function: Sample 5

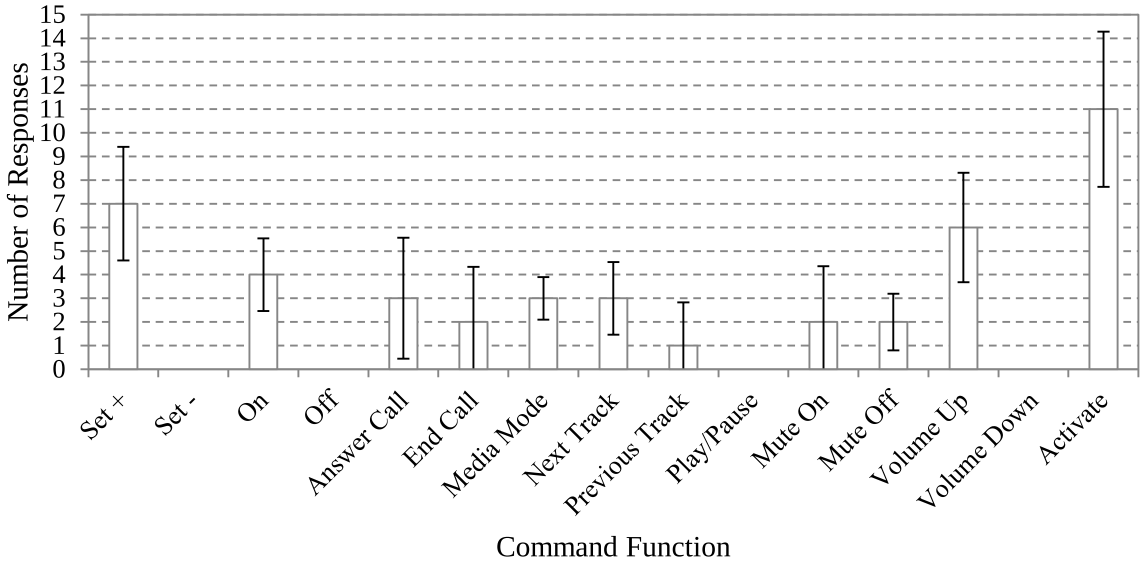



Figure 45. Response vs. Command Function: Sample 6

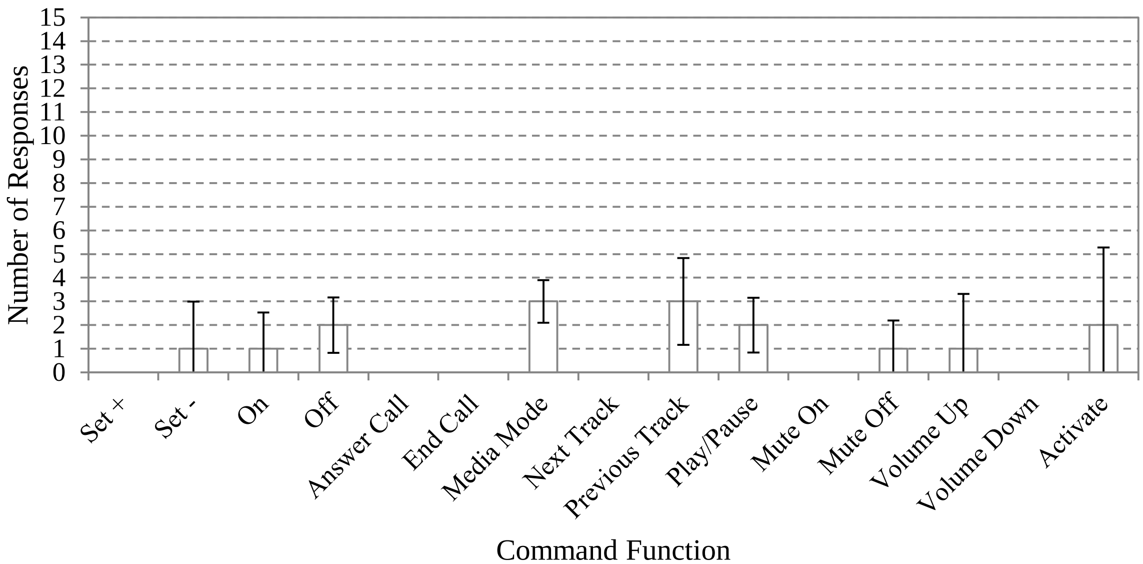



Figure 46. Response vs. Command Function: Sample 7

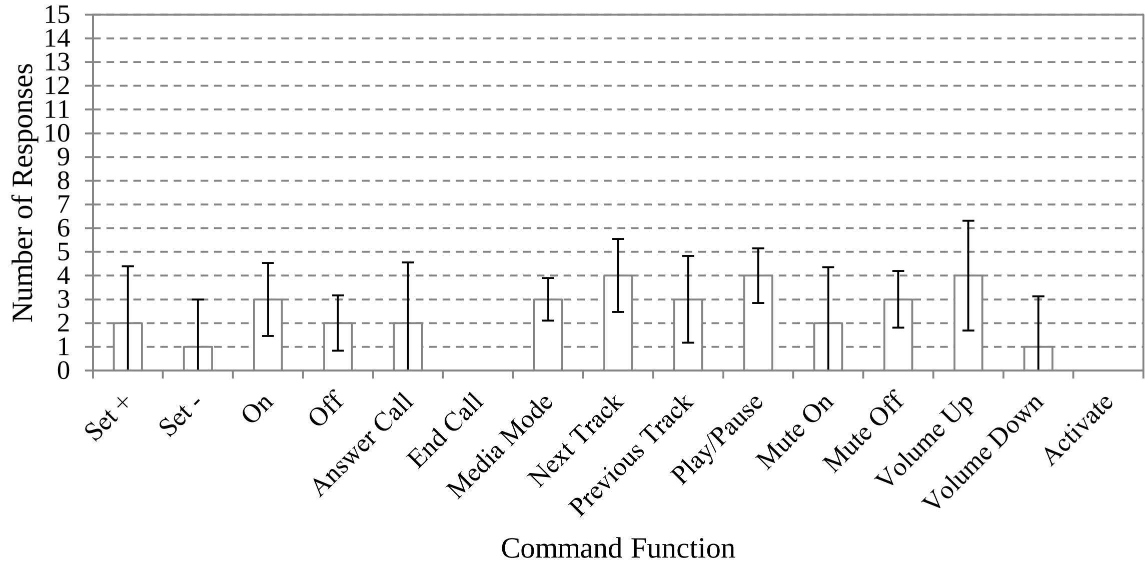



Figure 47. Response vs. Command Function: Sample 8

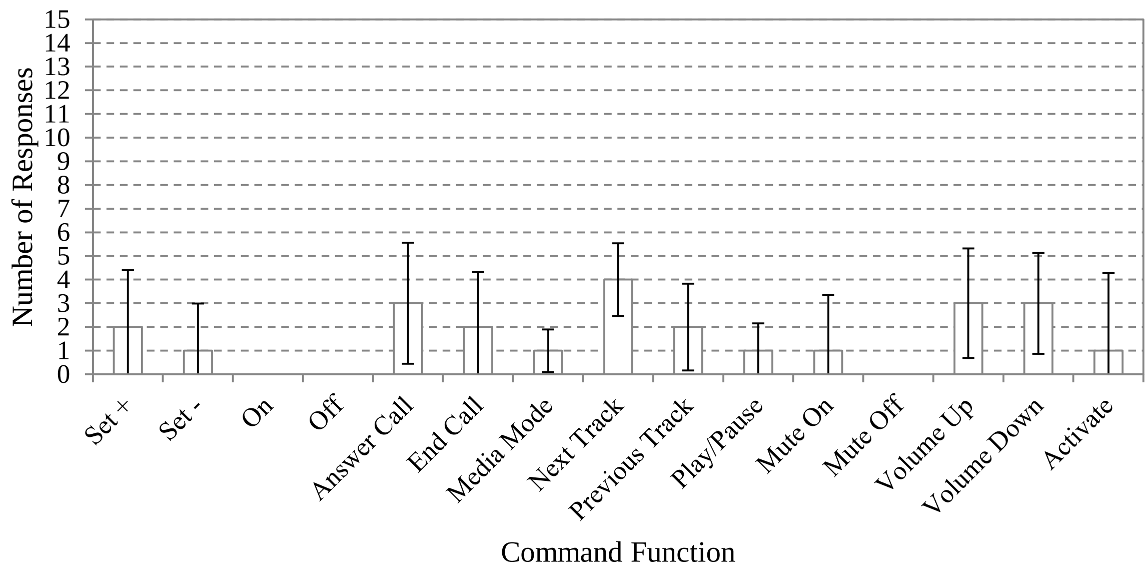



Figure 48.  Response vs. Command Function: Sample 9

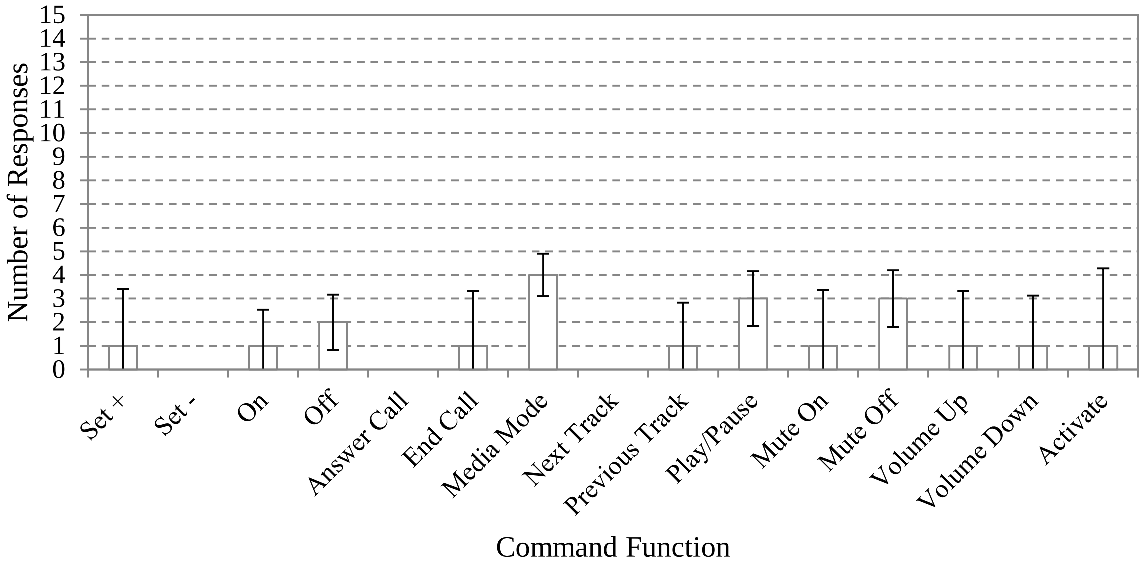



Figure 49. Response vs. Command Function: Sample 10

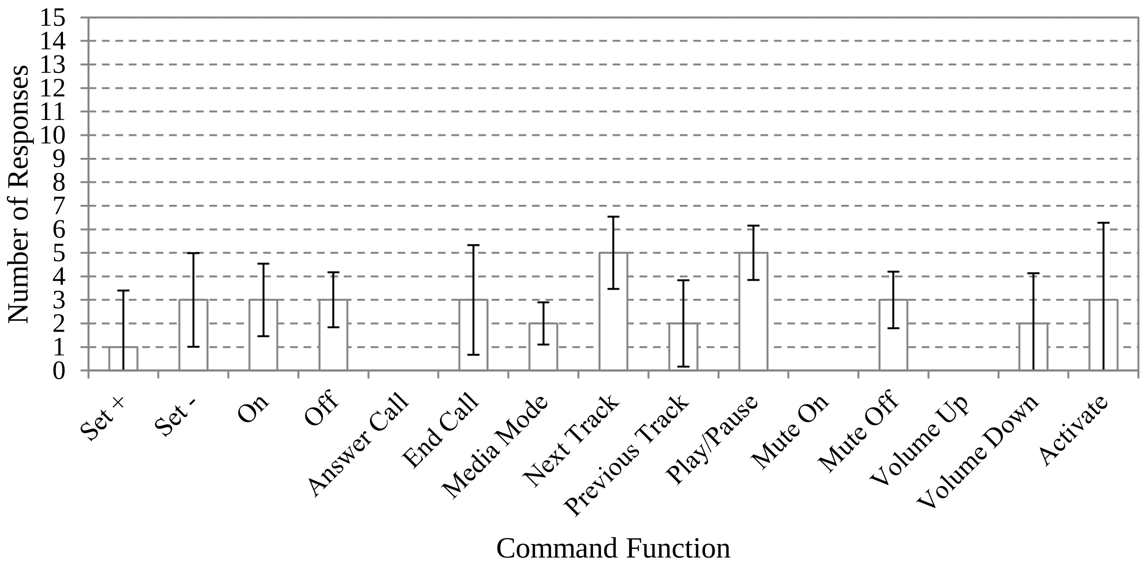



Figure 50. Response vs. Command Function: Sample 11

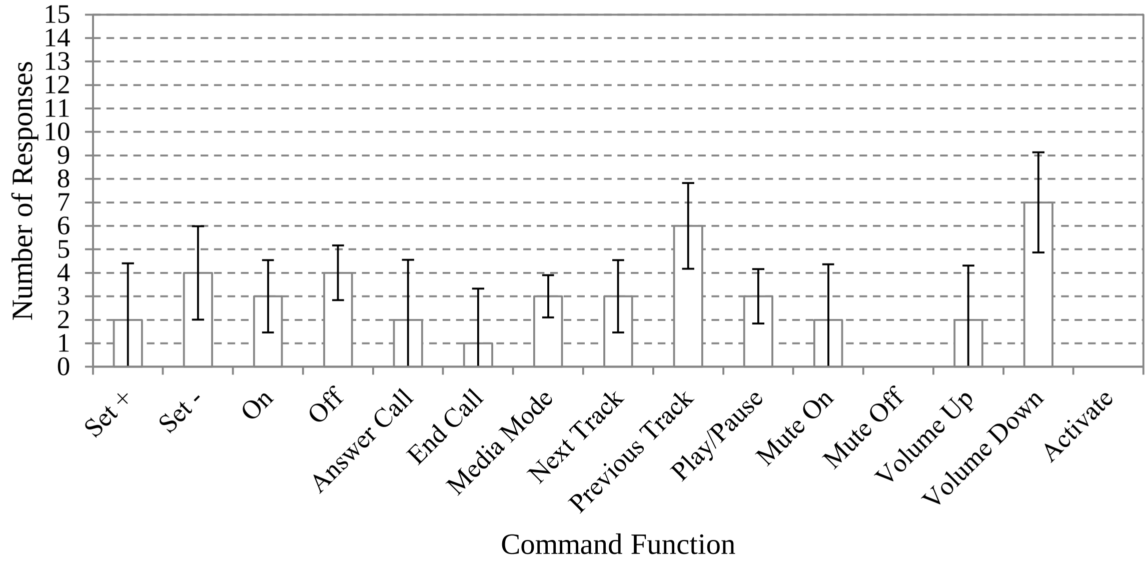



Figure 51. Response vs. Command Function: Sample 12

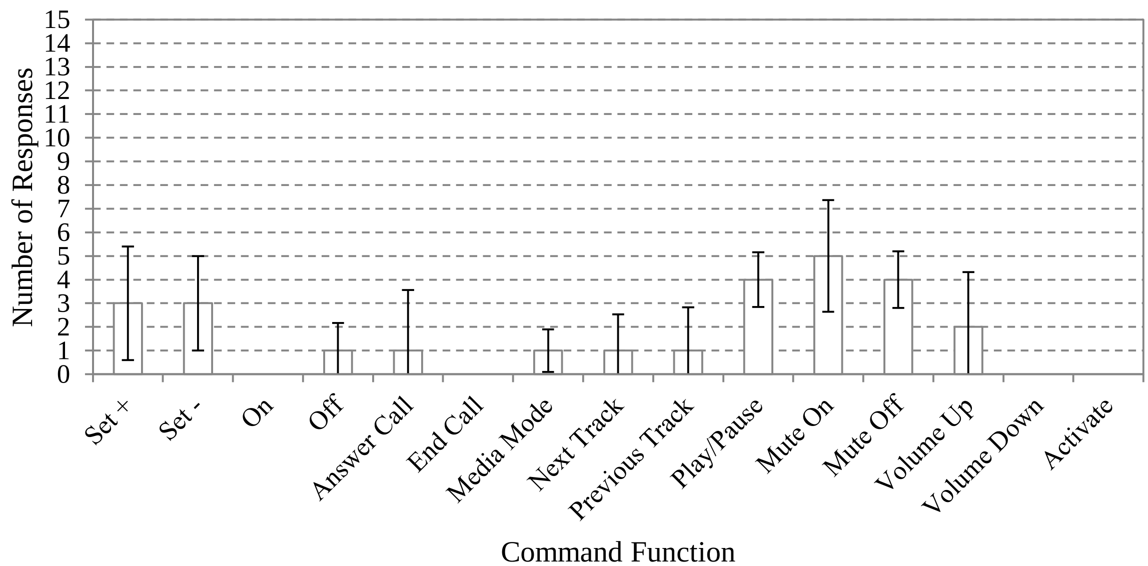



Figure 52. Response vs. Command Function: Sample 13

### 4.4.3. Levene Test

Table 6 shows the results from the Levene tests. Values below $\alpha = 0.05$ are shown in red. Five of the sixteen psychoacoustic metrics failed to produce a P-value greater than

$\alpha = 0.05$ for any of the three methods, indicating that those metrics were found to have unequal variances and were therefore non-parametric. Two had at least one method pass, while the remaining eight metrics, as well as the command functions, were successful for all three methods. This indicates that those metrics were found to have equal varainces and were therefore parametric.

Table 6: Levene Test Results

| | Mean | Median | 10% Trimmed Mean |
|---:|---:|---:|---:|
| Long | 0.01 | 0.05 | 0.02 |
| Dark | < 0.01 | < 0.01 | < 0.01 |
| Dynamic | < 0.01 | < 0.01 | < 0.01 |
| Sharp | < 0.01 | < 0.01 | < 0.01 |
| High Freq | 0.09 | 0.29 | 0.09 |
| Intense | 0.01 | 0.41 | 0.01 |
| Expressive | < 0.01 | < 0.01 | < 0.01 |
| Tonal | 0.03 | 0.15 | 0.03 |
| Short | 0.57 | 0.71 | 0.54 |
| Loud | 0.29 | 0.69 | 0.31 |
| Strong | < 0.01 | < 0.01 | < 0.01 |
| Smooth | 0.08 | 0.57 | 0.09 |
| Low Freq | 0.11 | 0.21 | 0.12 |
| Solid | 0.88 | 0.99 | 0.91 |
| Reliable | 0.14 | 0.68 | 0.17 |
| Powerful | 0.26 | 0.50 | 0.30 |
| Command Functions | 0.13 | 0.36 | 0.16 |

Table 7 shows the parametricity of each data set based on whether or not they passed the Levene test. The parametricity of a given data set determined which subsequent statistical tests would be conducted. The results from the median method were used in determining which test to choose, as it is found to have a good balance of robustness and power for many types of non-normal data [16]. Because the median method was used for all cases (including those with conflicting results), additional tests examining underlying data distribution were not necessary.

Table 7: Levene Test Conclusions

| | **Parametric** | **Non-Parametric** |
|---|---|---|
| Long | X | |
| Dark | | X |
| Dynamic | | X |
| Sharp | | X |
| High Freq | X | |
| Intense | X | |
| Expressive | | X |
| Tonal | X | |
| Short | X | |
| Loud | X | |
| Strong | | X |
| Smooth | X | |
| Low Freq | X | |
| Solid | X | |
| Reliable | X | |
| Powerful | X | |
| Command Functions | X | |

### 4.4.4. Analysis of Variance

Table 8 shows a summary of the ANOVA tests conducted on all parametric data sets. All analyzed metrics were found to have F statistics above $F_{crit}$, meaning that subjects were able to discern that all the sound samples were not simply the same sound played over and over. All analyzed metrics were also found to have P-values below $\alpha = 0.05$, which determines with 95% certainty that the distributions for each metric did not happen by chance. The complete ANOVA tables for each applicable data set can be found in Appendix B.

Table 8: Summary of ANOVA Test Results

| | **F** | **$F_{crit}$** | **P-value** |
|---|---|---|---|
| Long | 24.50 | 1.78 | < 0.01 |
| Dark | | | |
| Dynamic | | | |
| Sharp | | | |
| High Freq | 11.44 | 1.78 | < 0.01 |
| Intense | 11.54 | 1.78 | < 0.01 |
| Expressive | | | |
| Tonal | 12.44 | 1.78 | < 0.01 |
| Short | 5.36 | 1.79 | < 0.01 |
| Loud | 12.74 | 1.78 | < 0.01 |
| Strong | | | |
| Smooth | 3.93 | 1.78 | < 0.01 |
| Low Freq | 8.45 | 1.78 | < 0.01 |
| Solid | 1.93 | 1.78 | 0.03 |
| Reliable | 1.90 | 1.78 | 0.03 |
| Powerful | 3.21 | 1.78 | < 0.01 |
| Command Functions | 5.86 | 1.78 | < 0.01 |

#### 4.4.5. Kruskal-Wallis Test

Table 9 shows a summary of the Kruskal-Wallis tests conducted on all non-parametric data sets. All analyzed metrics were found to have P-values below $\alpha = 0.05$, which determines with 95% certainty that the distributions for all of the metrics did not happen by chance. The complete Kruskal-Wallis tables for each applicable metric can be found in Appendix C.

Table 9: Summary of Kruskal-Wallis Test Results

| | **P-value** |
|---:|---:|
| Long | |
| Dark | < 0.01 |
| Dynamic | < 0.01 |
| Sharp | < 0.01 |
| High Freq | |
| Intense | |
| Expressive | < 0.01 |
| Tonal | |
| Short | |
| Loud | |
| Strong | < 0.01 |
| Smooth | |
| Low Freq | |
| Solid | |
| Reliable | |
| Powerful | |
| Command Functions | |

### 4.4.6. Correlation Coefficient

Figure 53 – Figure 67 show the significant correlation coefficients across all psychoacoustic metrics for each command function. Parametric data sets were measured with the Pearson product moment correlation coefficient, and non-parametric data sets were measured with Spearman's rank-order correlation coefficient. High correlation between psychoacoustic metrics and command functions points to a linkage between a sound which contains a given characteristic, and that sound's suitability to be played for a given command function. For example, in Figure 56 *reliable* and *off* are shown to be highly correlated to one another. Therefore, a sound which one may consider to be *reliable* would most likely also be well suited to be played for the *off* command. These correlations aide to simplify the process of assigning ideal sounds to each command

function, as it becomes apparent that a single sound may be assigned to more than one function while indicating successful initiation of the desired function to the user. Tabulated correlation coefficient values can be found in Appendix D.

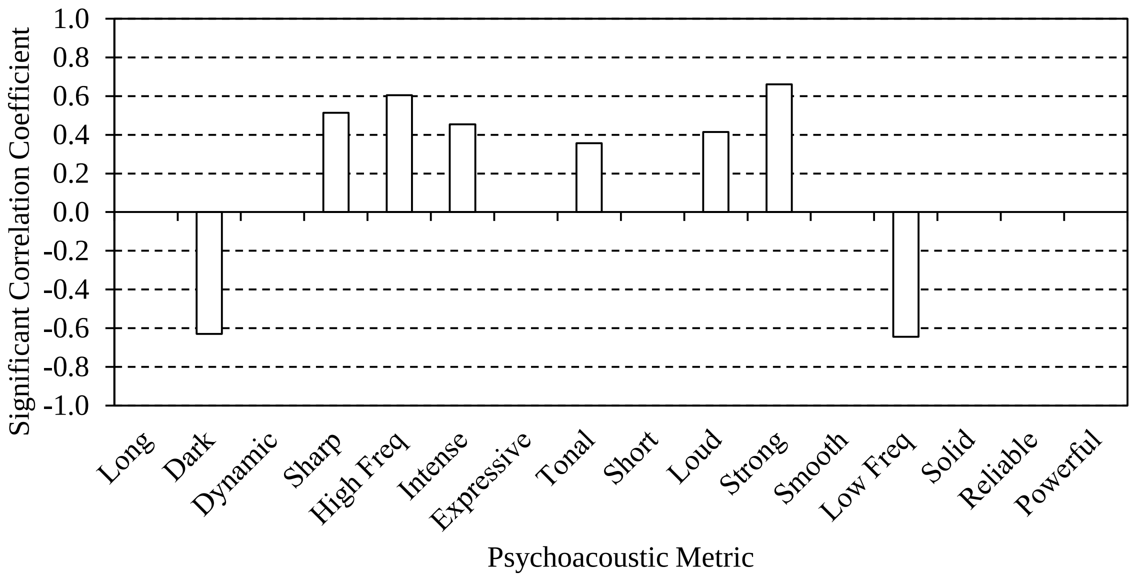



Figure 53. Significant Correlation Coefficient vs. Psychoacoustic Metric: Set +. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman’s Rank-Order Correlation Coefficient.

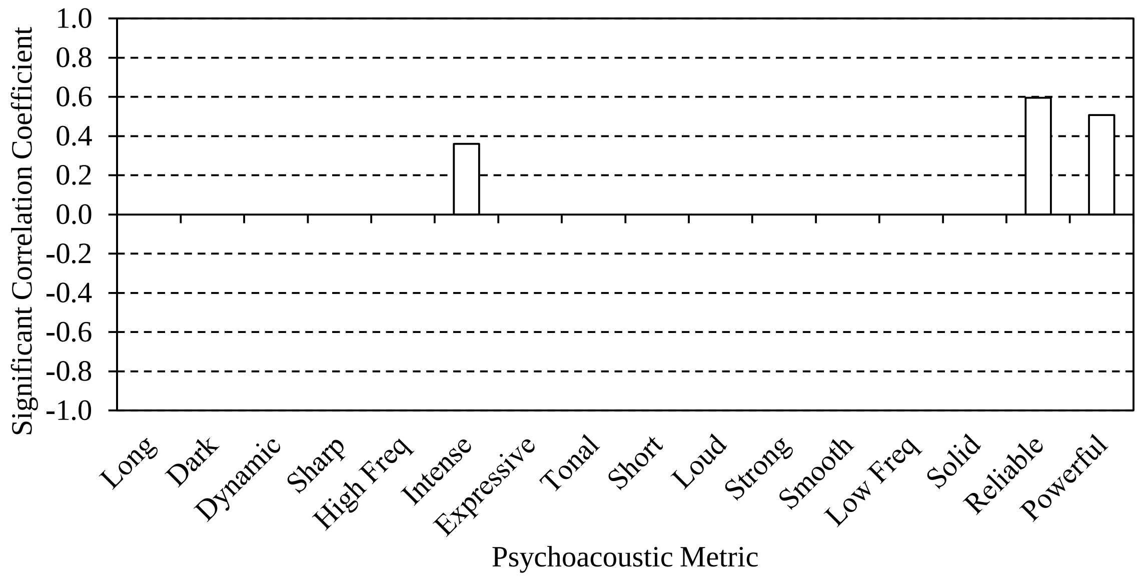



Figure 54. Significant Correlation Coefficient vs. Psychoacoustic Metric: Set –. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman’s Rank-Order Correlation Coefficient.

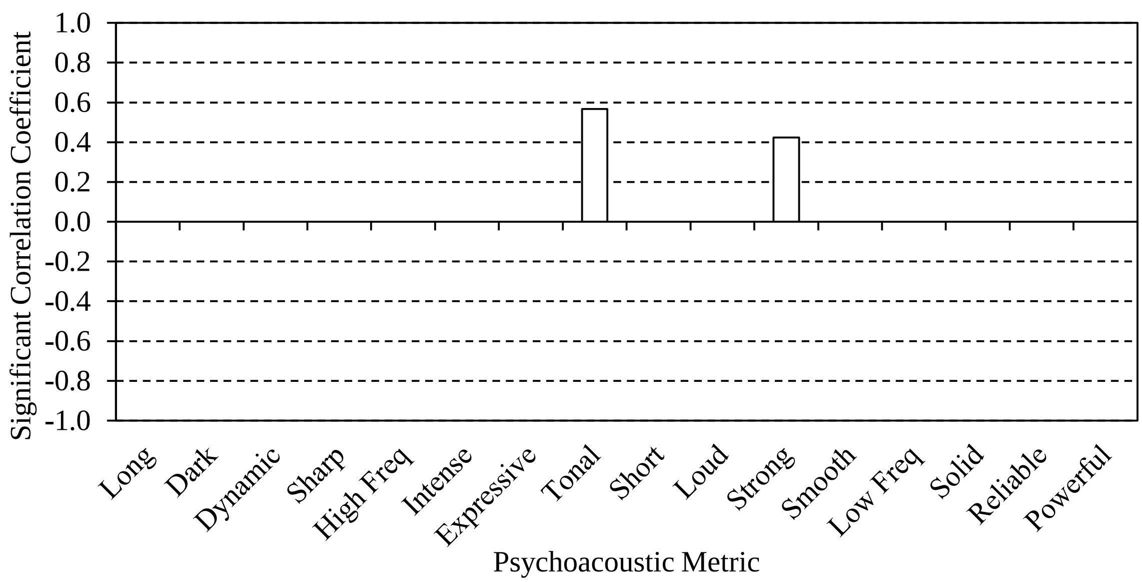



Figure 55. Significant Correlation Coefficient vs. Psychoacoustic Metric: On–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman’s Rank-Order Correlation Coefficient.

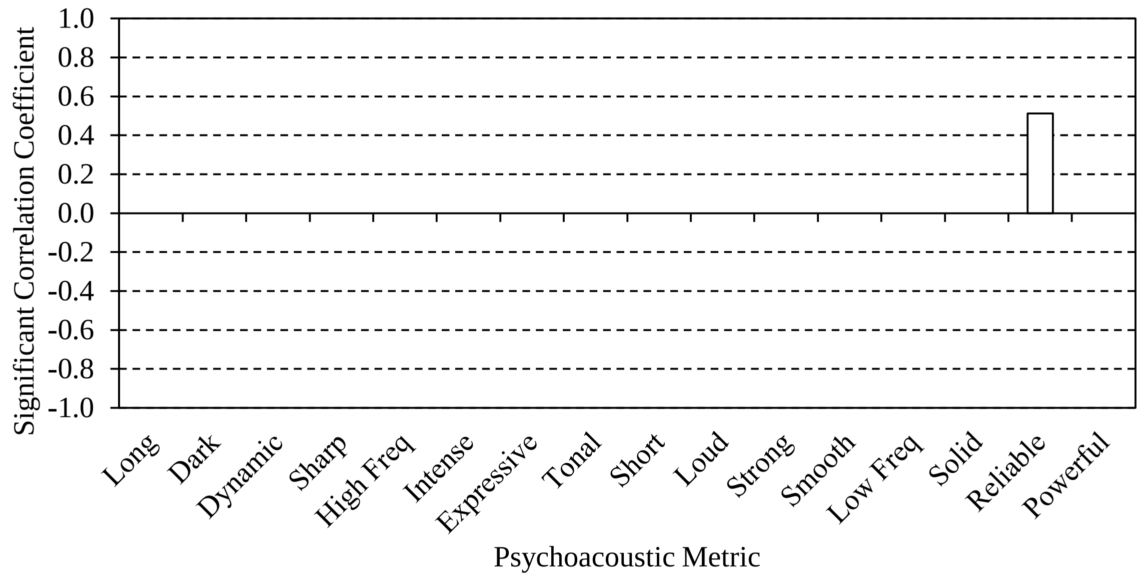



Figure 56. Significant Correlation Coefficient vs. Psychoacoustic Metric: Off–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman’s Rank-Order Correlation Coefficient.

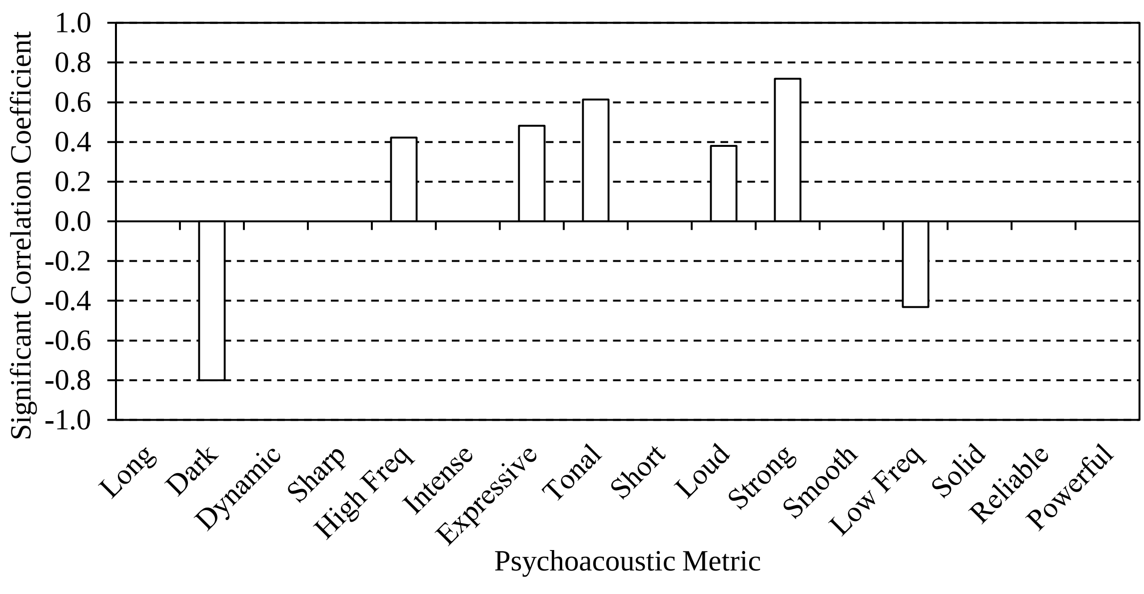



Figure 57. Significant Correlation Coefficient vs. Psychoacoustic Metric: Answer Call–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman’s Rank-Order Correlation Coefficient.

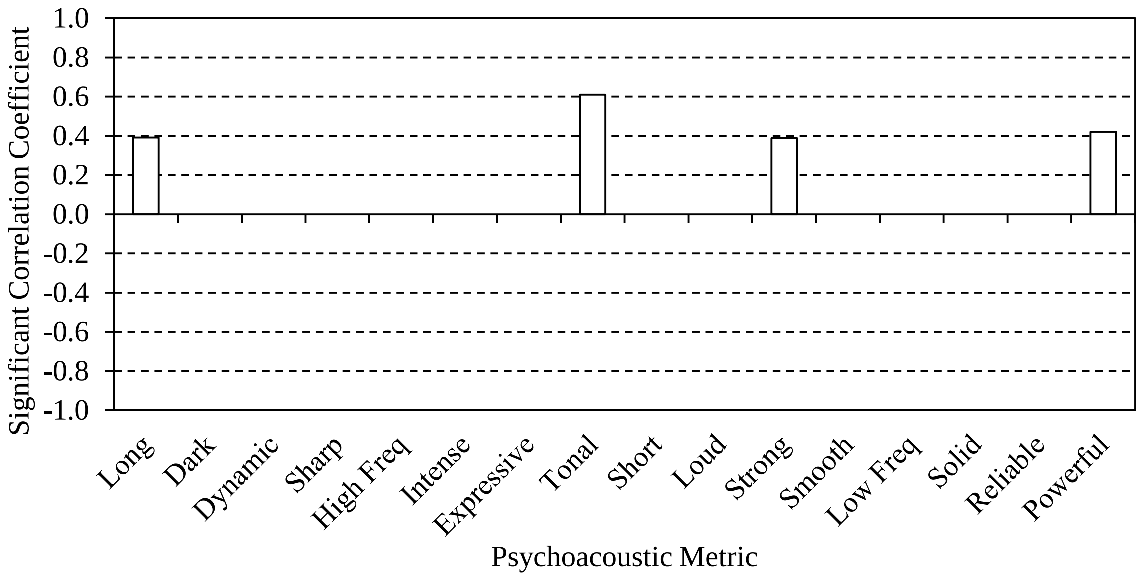



Figure 58. Significant Correlation Coefficient vs. Psychoacoustic Metric: End Call–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman's Rank-Order Correlation Coefficient.

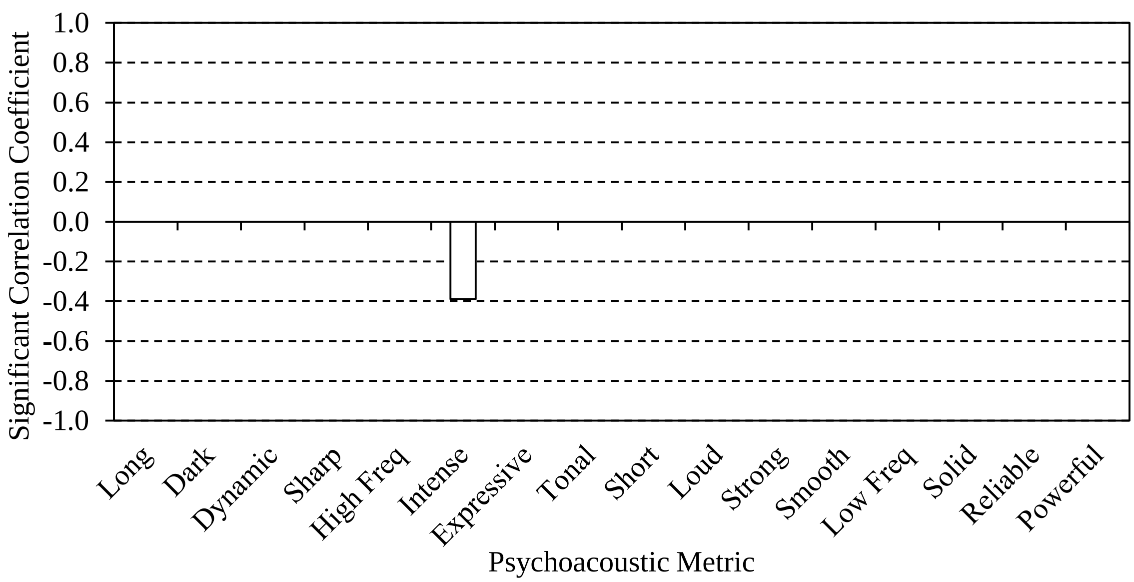



Figure 59. Significant Correlation Coefficient vs. Psychoacoustic Metric: Media Mode–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman's Rank-Order Correlation Coefficient.

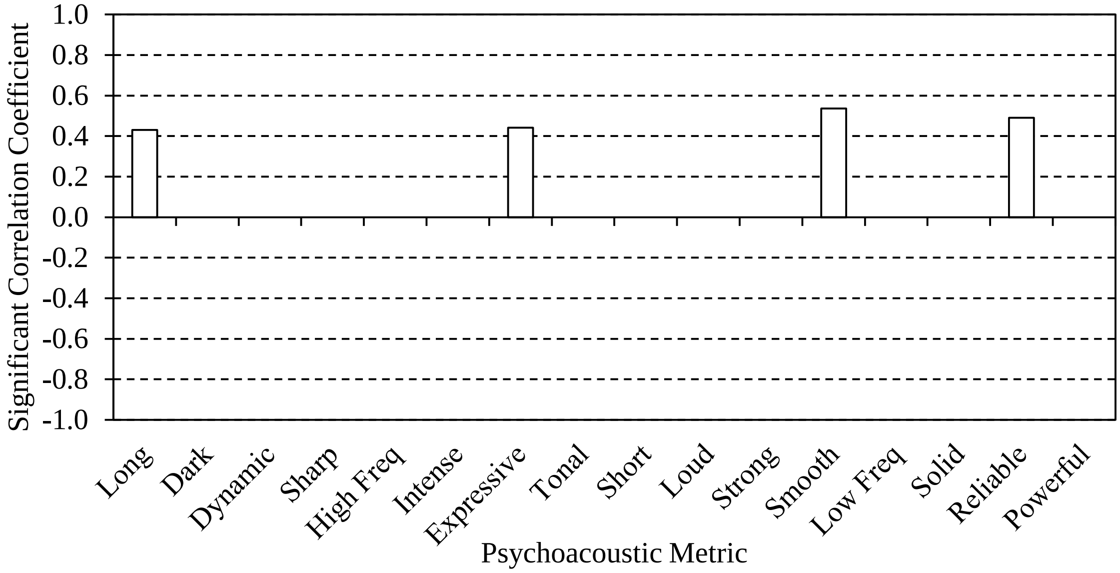



Figure 60. Significant Correlation Coefficient vs. Psychoacoustic Metric: Next Track–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman’s Rank-Order Correlation Coefficient.

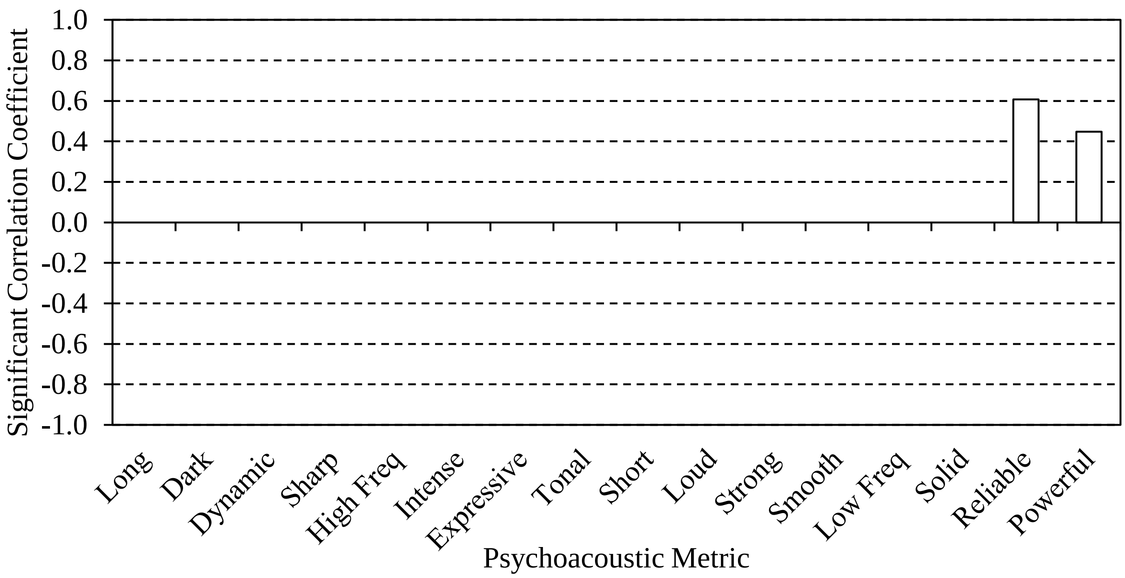



Figure 61. Significant Correlation Coefficient vs. Psychoacoustic Metric: Previous Track–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman’s Rank-Order Correlation Coefficient.

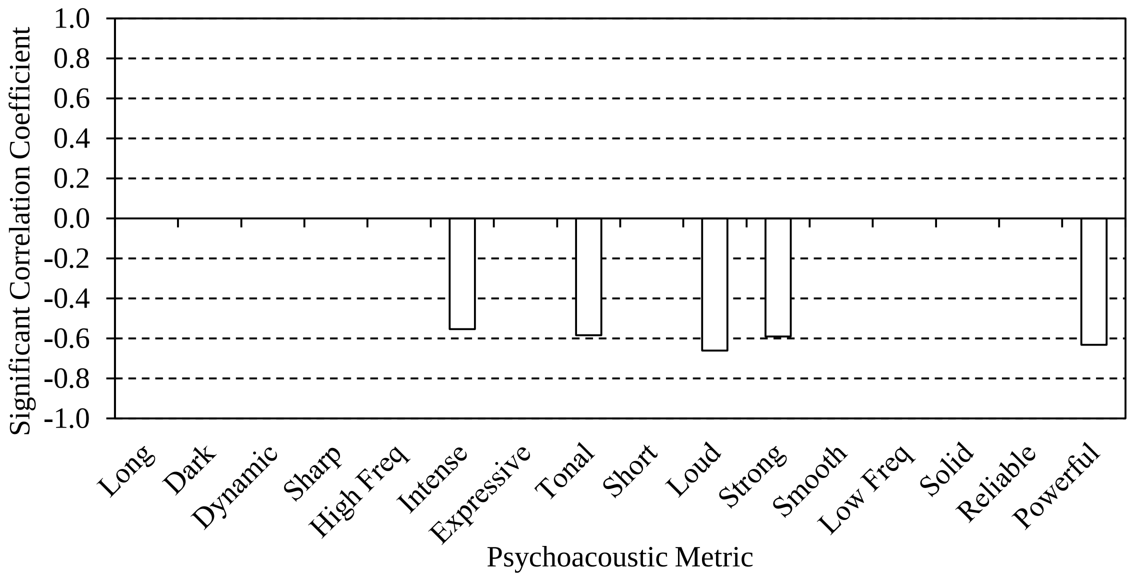



Figure 62. Significant Correlation Coefficient vs. Psychoacoustic Metric: Play/Pause–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman's Rank-Order Correlation Coefficient.

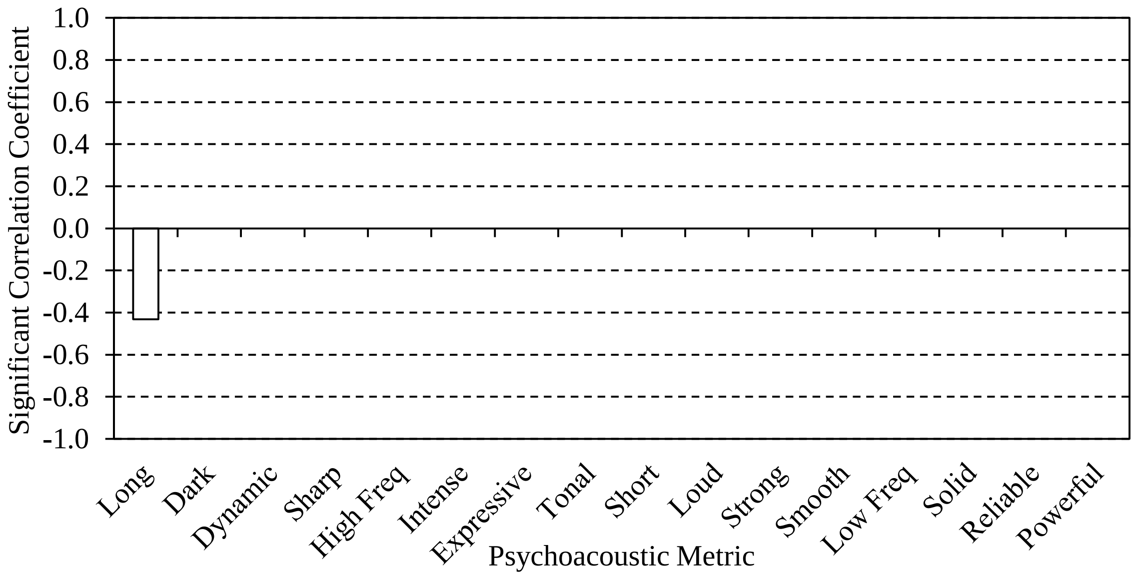



Figure 63. Significant Correlation Coefficient vs. Psychoacoustic Metric: Mute On–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman's Rank-Order Correlation Coefficient.

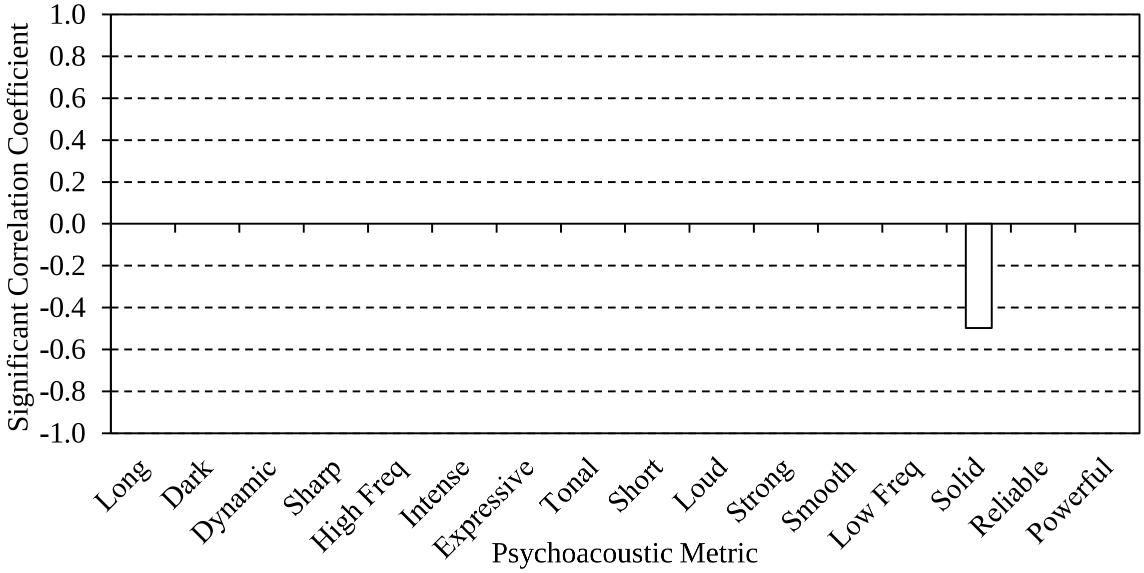



Figure 64. Significant Correlation Coefficient vs. Psychoacoustic Metric: Mute Off–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman's Rank-Order Correlation Coefficient.

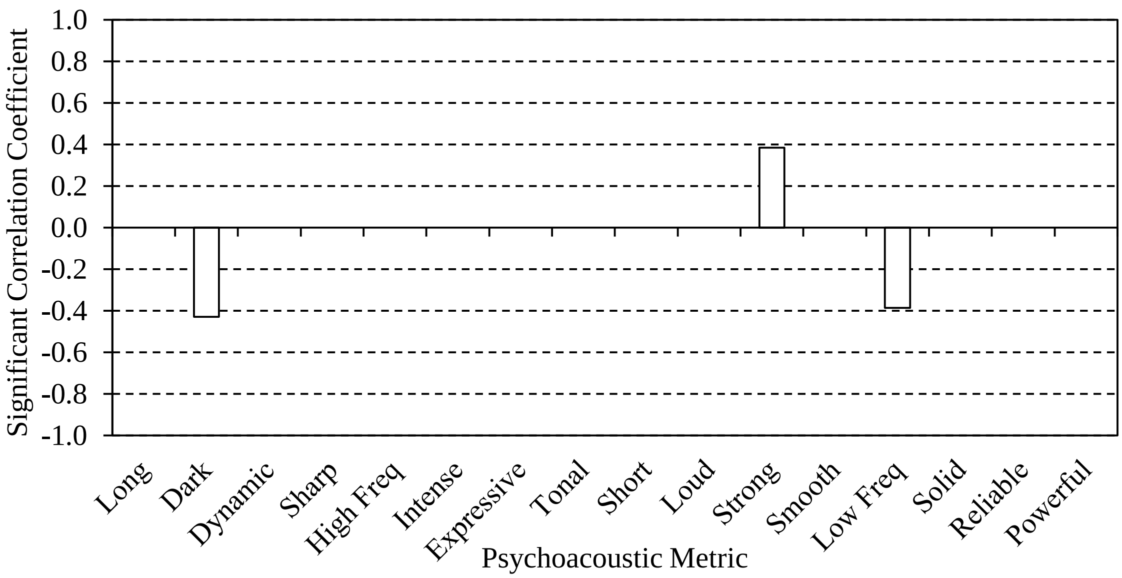



Figure 65. Significant Correlation Coefficient vs. Psychoacoustic Metric: Volume Up–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman's Rank-Order Correlation Coefficient.

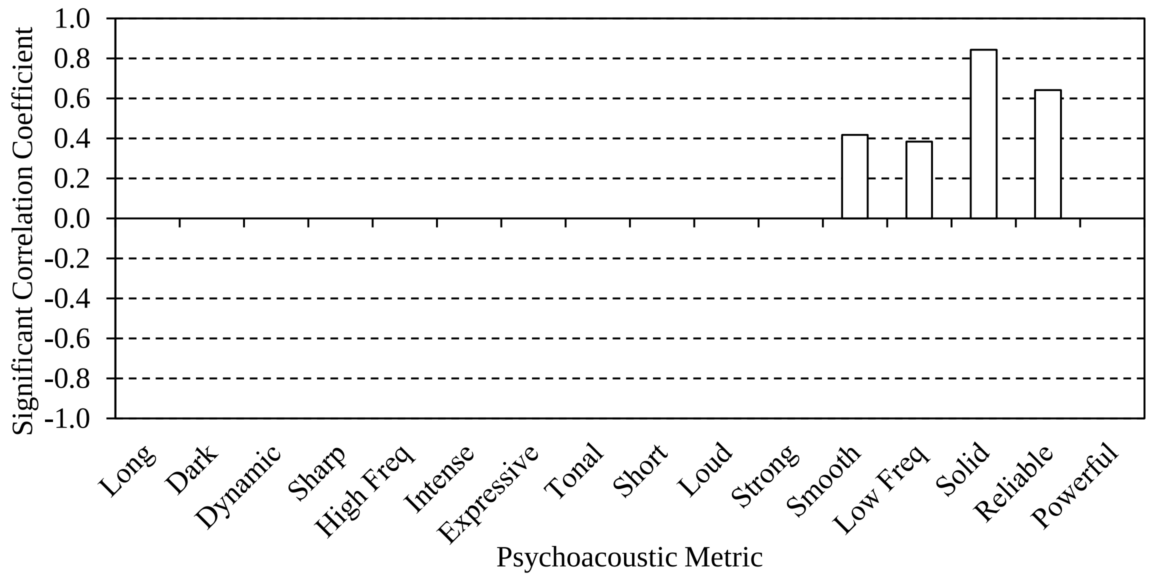



Figure 66. Significant Correlation Coefficient vs. Psychoacoustic Metric: Volume Down–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman's Rank-Order Correlation Coefficient.

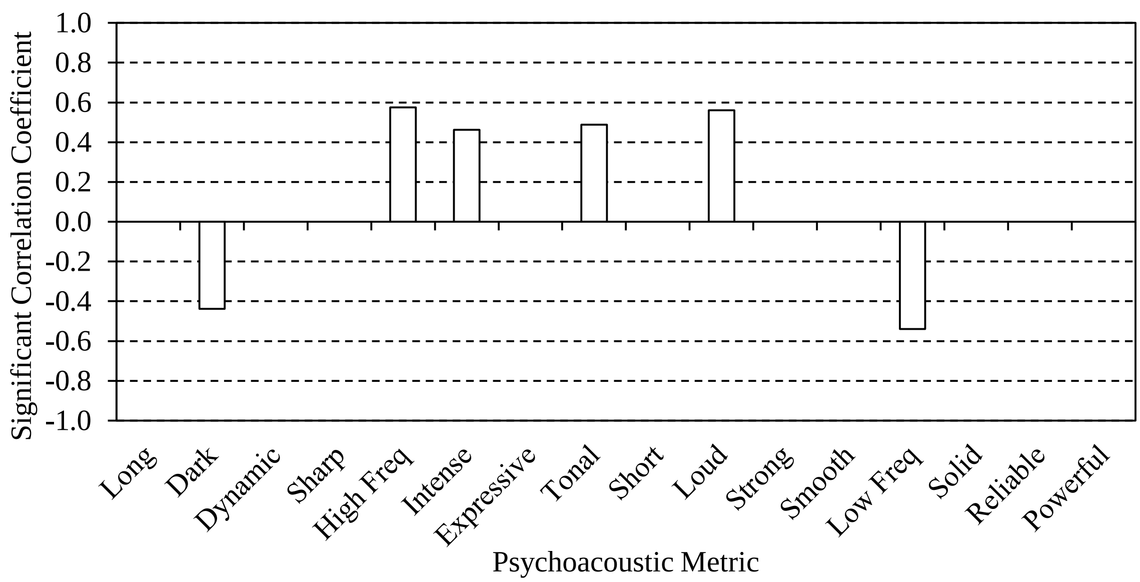



Figure 67. Significant Correlation Coefficient vs. Psychoacoustic Metric: Activate–. Parametric Metrics Used the Pearson Product Moment Correlation Coefficient. Non-Parametric Metrics Used Spearman's Rank-Order Correlation Coefficient.

Figure 68 – Figure 82 show the significant correlation coefficients across all command functions for each command function. Several pairs or groups of functions were found to have a strong correlation to one another. High correlation between multiple command functions points to a linkage between the sounds which activate those functions. For example, in Figure 76 and Figure 81 *previous track* and *volume down* are shown to be highly correlated to one another. Therefore, a sound which one may consider to be well suited to play for the *previous track* command would most likely be well suited to play for the *volume down* command as well. Thus there is no need to assign a unique sound to each individual command function. A single group of correlated command functions can all be assigned the same sound without sacrificing successful indication of function initiation.

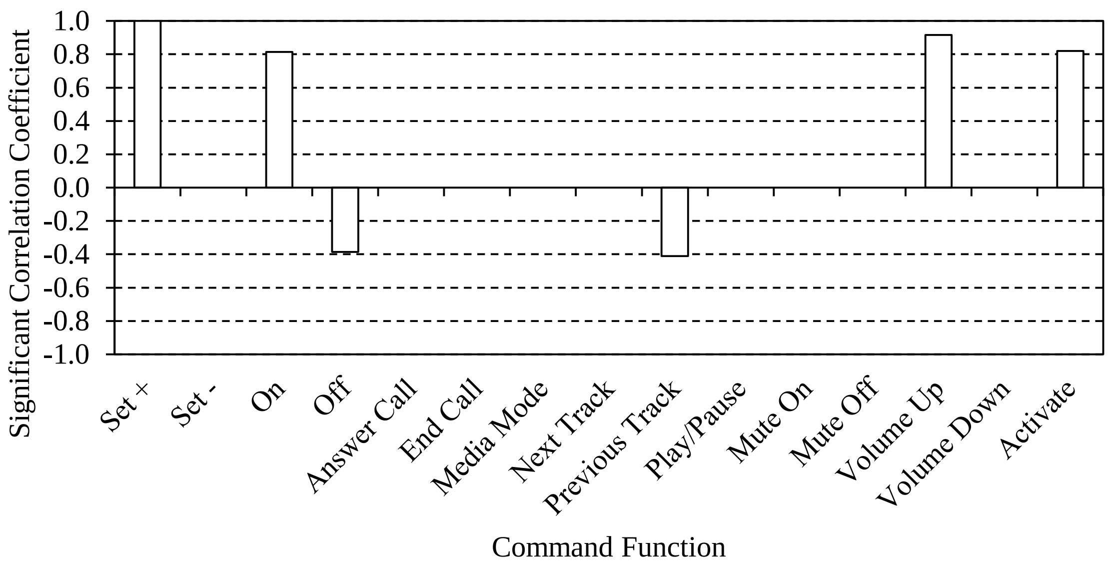



Figure 68. Significant Correlation Coefficient vs. Command Function: Set +

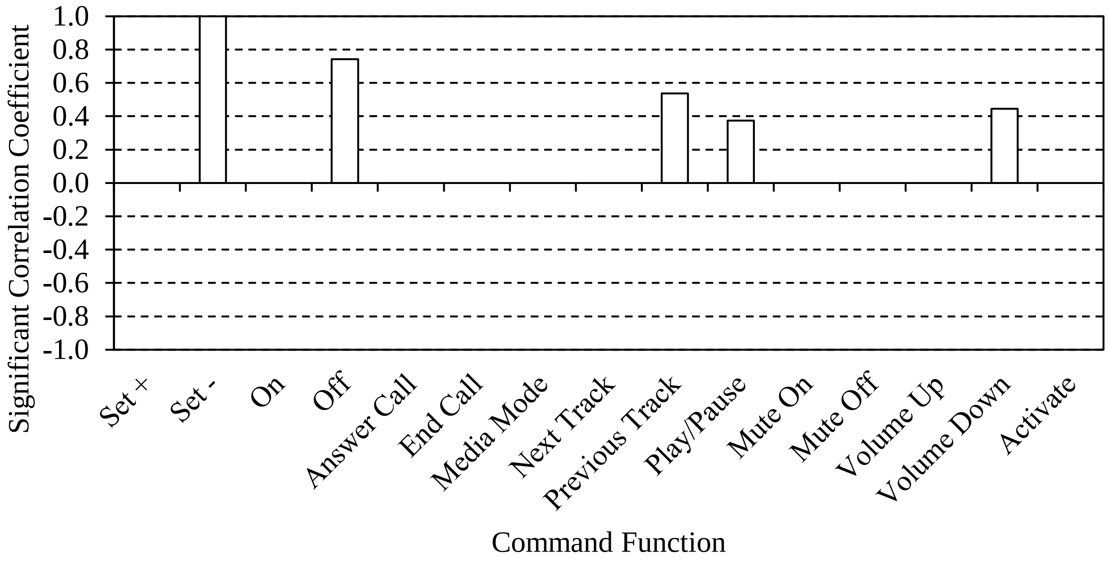



Figure 69. Significant Correlation Coefficient vs. Command Function: Set –

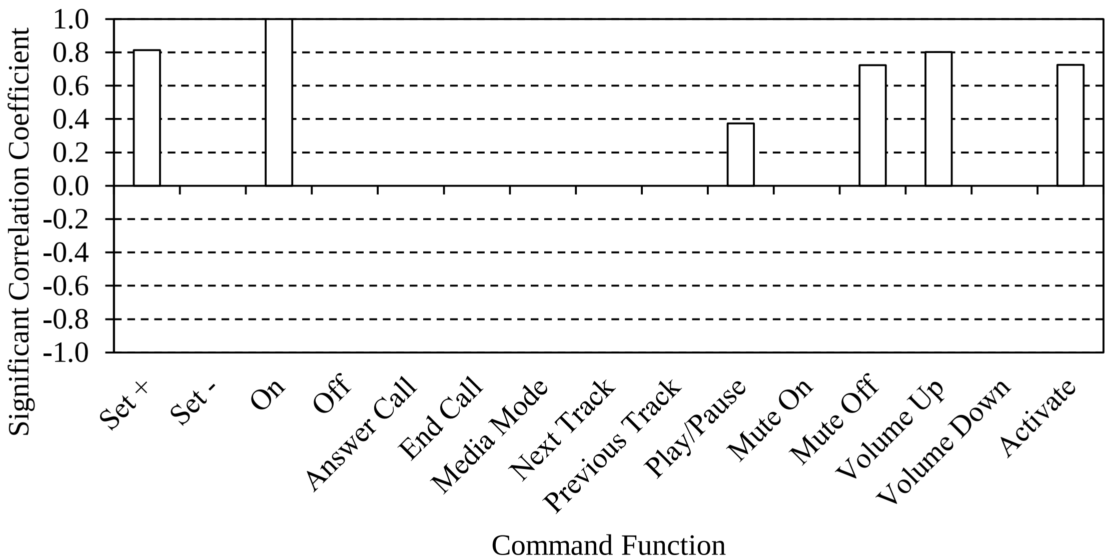



Figure 70. Significant Correlation Coefficient vs. Command Function: On

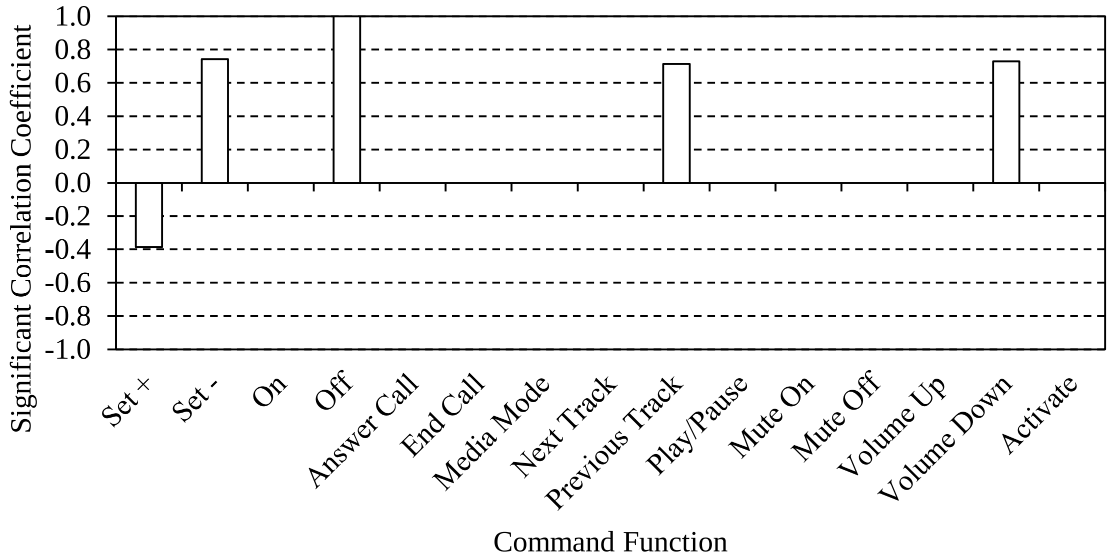



Figure 71. Significant Correlation Coefficient vs. Command Function: Off

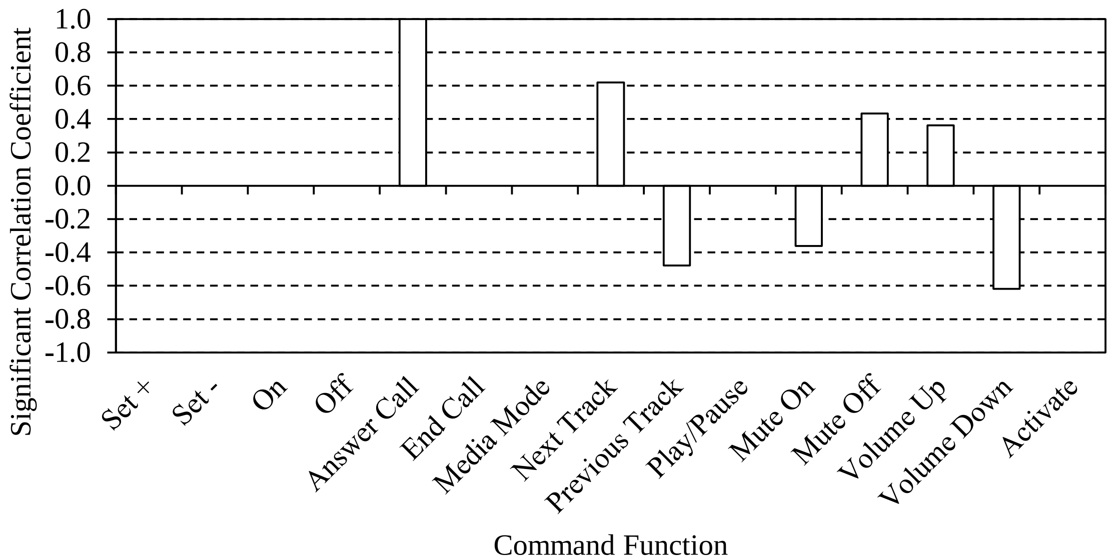



Figure 72. Significant Correlation Coefficient vs. Command Function: Answer Call

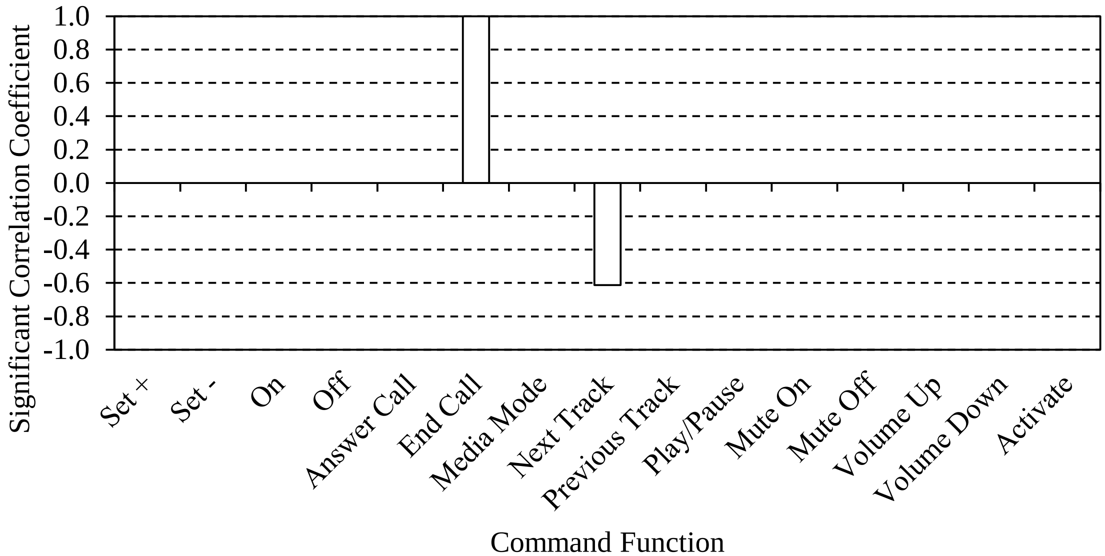



Figure 73. SignificantCorrelation Coefficient vs. Command Function: End Call

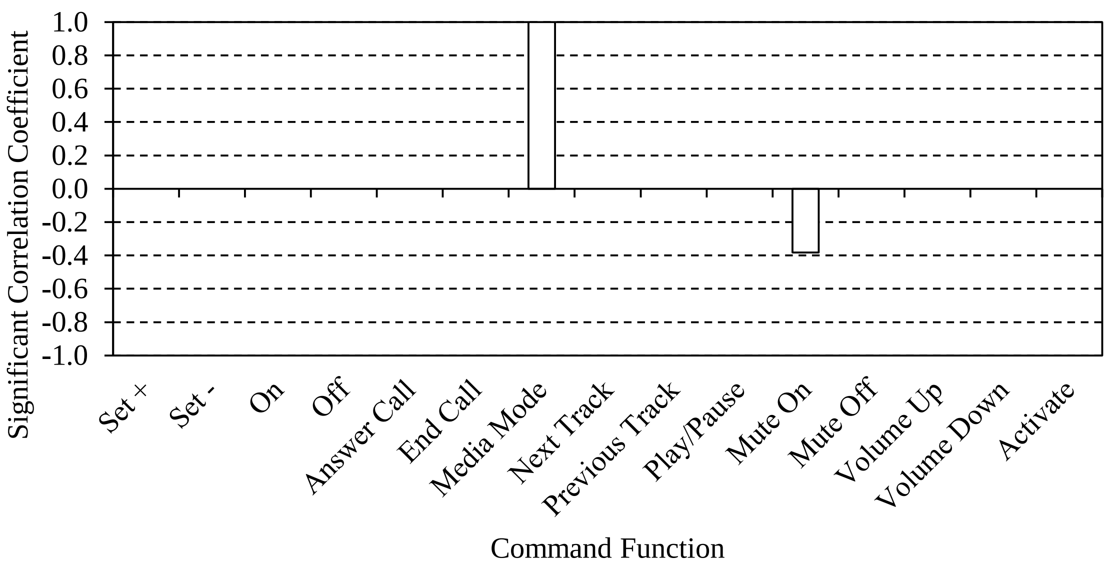



Figure 74. Significant Correlation Coefficient vs. Command Function: Media Mode

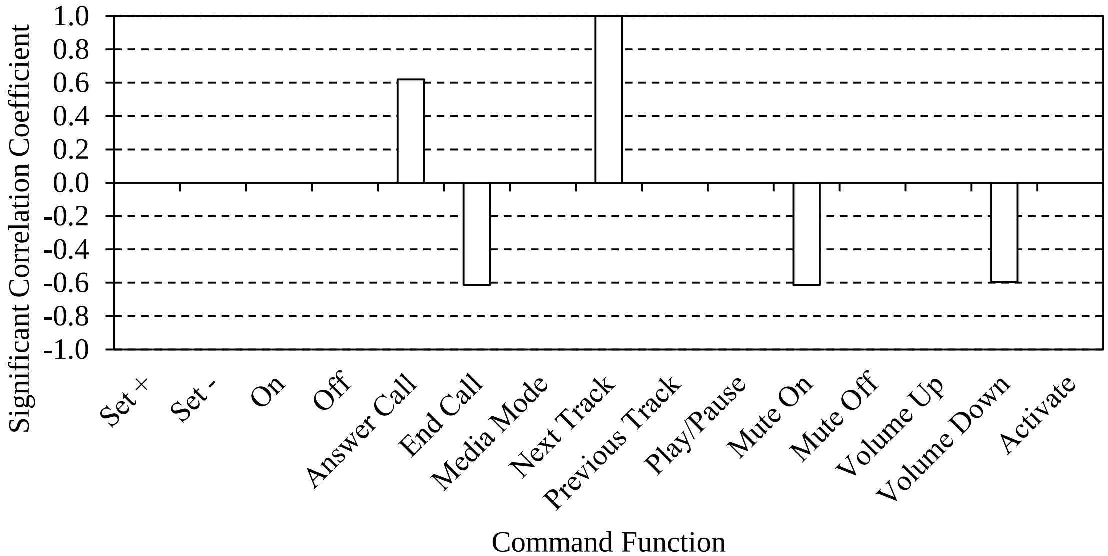



Figure 75. Significant Correlation Coefficient vs. Command Function: Next Track

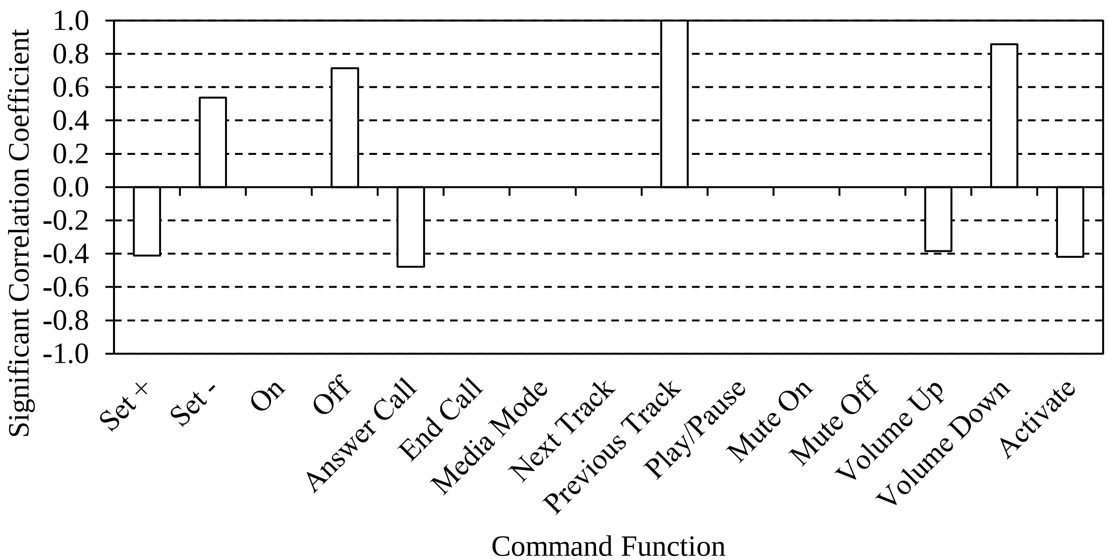



Figure 76. Significant Correlation Coefficient vs. Command Function: Previous Track

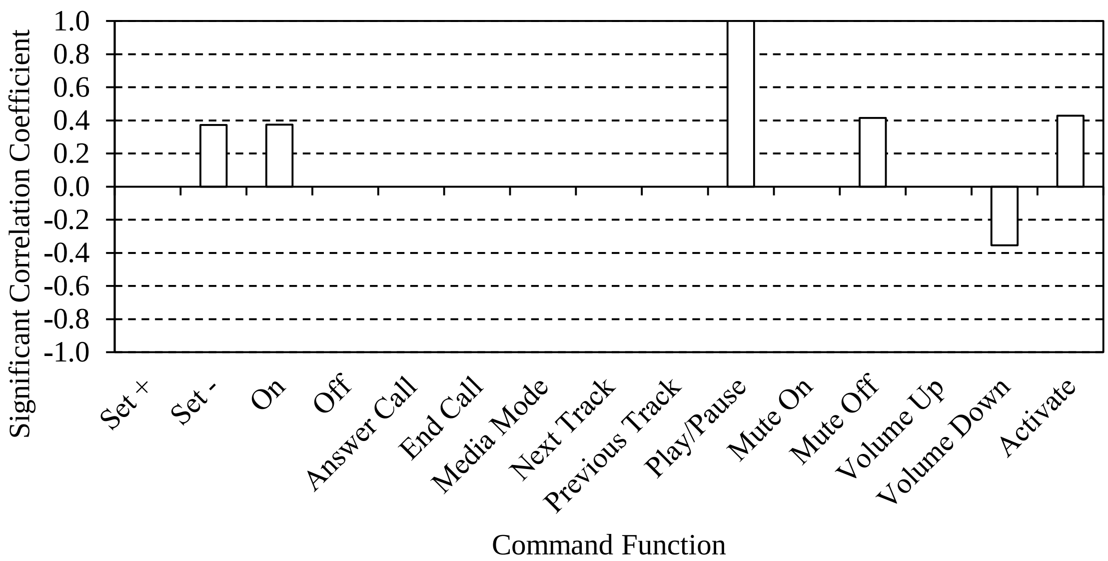



Figure 77. Significant Correlation Coefficient vs. Command Function: Play/Pause

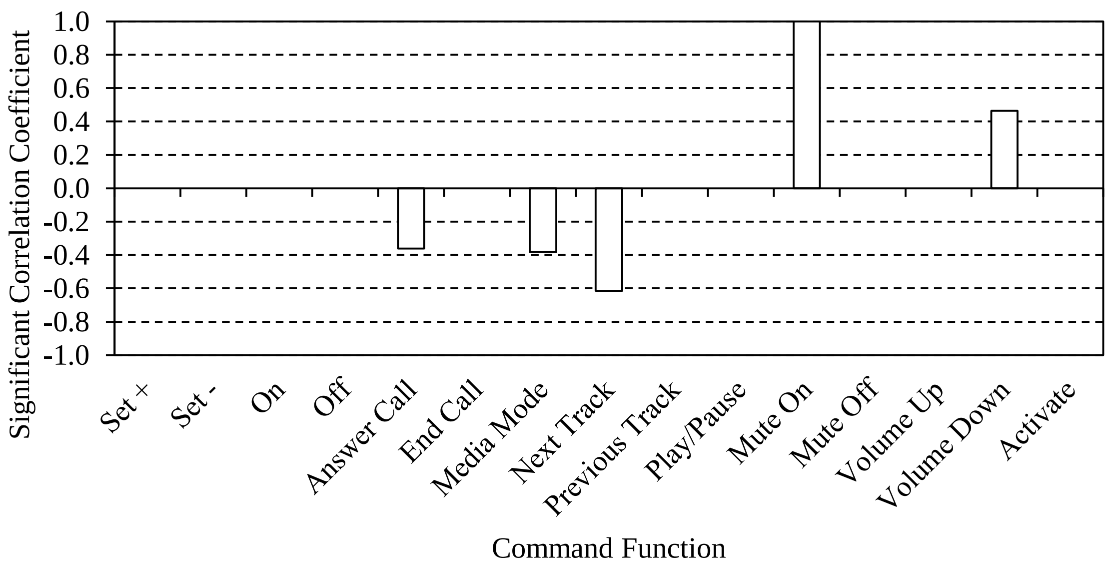



Figure 78. Significant Correlation Coefficient vs. Command Function: Mute On

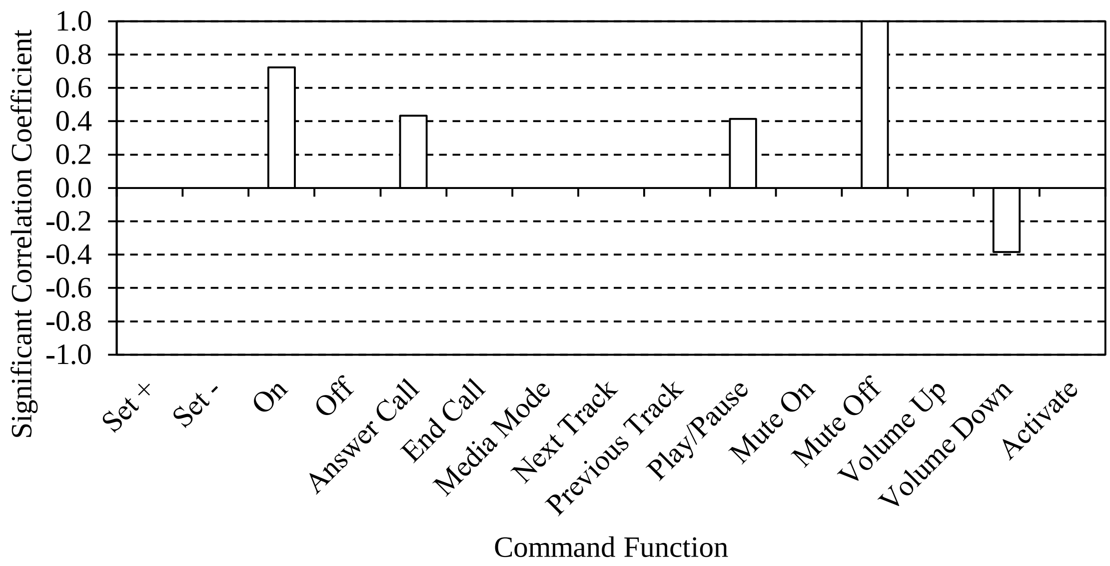



Figure 79. Significant Correlation Coefficient vs. Command Function: Mute Off

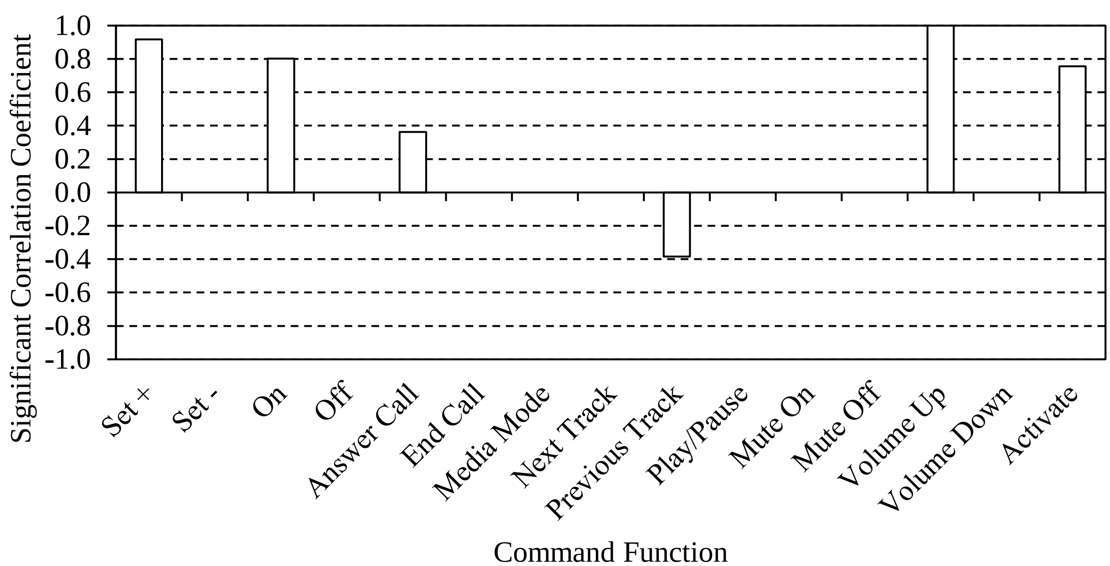



Figure 80. Significant Correlation Coefficient vs. Command Function: Volume Up

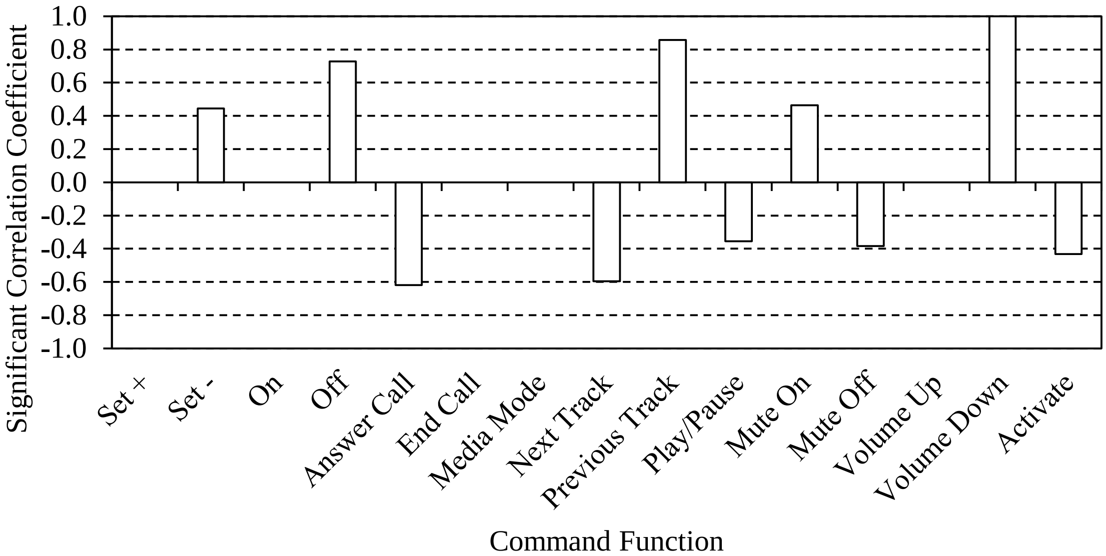



Figure 81. Significant Correlation Coefficient vs. Command Function: Volume Down

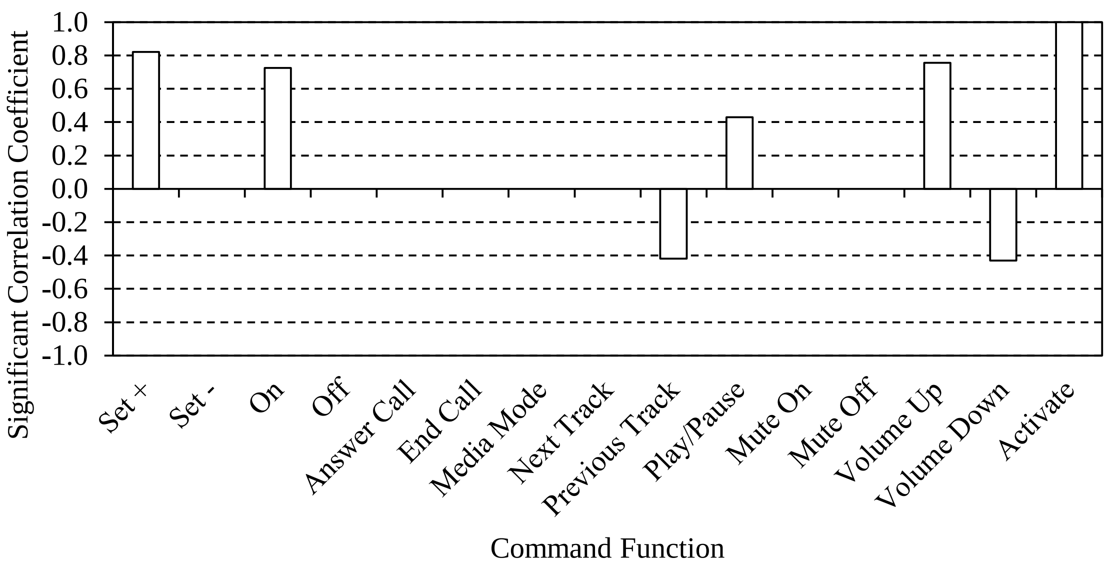



Figure 82. Significant Correlation Coefficient vs. Command Function: Activate

### 4.4.7. Post Hoc T-test

Table 10 shows a summary of the percentage of statistically significant values ($P < 0.05$) obtained from the T-test, which was conducted for all parametric data sets. This test determined whether the difference between each pair of sample sounds was significant. The T-test was run on all parametric data sets because the previous parametric tests found statistical significance in all data sets. The majority of psychoacoustic metrics (as well as the command functions) had a statistical significance in the range of 40% – 70%. The metrics that had lower statistical significance (20% - 40%) were those which one might consider to be more ambiguous in meaning. A subject may have been more likely to be confused as to what exactly a *reliable* or *solid* sound sounds like, which may have contributed to the reduced statistical significance found in the T-test. Complete T-test tabulated results for all parametric data sets can be found in Appendix E.

Table 10. Summary of T-test Results ($P < 0.05$)

| | **% Statistically Significant** |
|---:|:---:|
| *Long* | 69 |
| *Dark* | |
| *Dynamic* | |
| *Sharp* | |
| *High Freq* | 63 |
| *Intense* | 55 |
| *Expressive* | |
| *Tonal* | 71 |
| *Short* | 47 |
| *Loud* | 68 |
| *Strong* | |
| *Smooth* | 38 |
| *Low Freq* | 59 |
| *Solid* | 23 |
| *Reliable* | 24 |
| *Powerful* | 35 |
| *Command Functions* | 45 |

### 4.4.8. Post Hoc Wilcoxon Signed Ranks Test

Table 11 shows a summary of the percentage of statistically significant values ($P < 0.05$) obtained from the Wilcoxon signed ranks test, which was conducted for all non-parametric data sets. This test determined whether the difference between each pair of sample sounds was significant. The Wilcoxon test was run on all non-parametric data sets because the previous non-parametric tests found statistical significance in all data sets. All measured psychoacoustic metrics had a statistical significance in the range of 50% – 80%. Complete Wilcoxon test tabulated results for all non-parametric data sets can be found in Appendix F.

Table 11. Summary of Wilcoxon Signed Ranks Test Results ($P < 0.05$)

| | **% Statistically Significant** |
|---:|:---:|
| *Long* | |
| *Dark* | 51 |
| *Dynamic* | 81 |
| *Sharp* | 54 |
| *High Freq* | |
| *Intense* | |
| *Expressive* | 64 |
| *Tonal* | |
| *Short* | |
| *Loud* | |
| *Strong* | 57 |
| *Smooth* | |
| *Low Freq* | |
| *Solid* | |
| *Reliable* | |
| *Powerful* | |
| *Command Functions* | |

### 4.4.9. Sample Ranking

Table 12 shows the sound sample rankings for each command function. Because several command functions had multiple sound samples tied for a single rank, no command function had a ranking lower than 6th place. Sound samples in the "No Appearance" column were never chosen by any subject for that command function.

Table 12. Command Function vs. Sound Sample Rankings

| | **1st** | **2nd** | **3rd** | **4th** | **5th** | **6th** | **No Appearance** |
|---|---|---|---|---|---|---|---|
| Set + | 1 | 6 | 13 | 3, 4, 8, 9, 12 | 5, 10, 11 | | 2, 7 |
| Set - | 5 | 12 | 2, 3, 11, 13 | 1, 4 | 7, 8, 9 | | 6, 10 |
| On | 1 | 3, 6 | 4, 8, 11, 12 | 5 | 2, 7, 10 | | 9, 13 |
| Off | 5 | 2, 12 | 3, 4, 11 | 1, 7, 8, 10 | 13 | | 6, 9 |
| Answer Call | 4 | 1 | 5, 6, 9 | 8, 12 | 2, 3, 13 | | 7, 10, 11 |
| End Call | 3 | 1, 2 | 5, 11 | 4, 6, 9 | 10, 12 | | 7, 8, 13 |
| Media Mode | 10 | 1, 3, 4, 6, 7, 8, 12 | 2, 11 | 9, 13 | | | 5 |
| Next Track | 4, 11 | 8, 9 | 6, 12 | 1, 5 | 2, 3, 13 | | 7, 10 |
| Previous Track | 12 | 3, 5 | 7, 8 | 9, 11 | 1, 4, 6, 10, 13 | | 2 |
| Play/Pause | 11 | 8, 13 | 1, 4, 10, 12 | 2, 7 | 9 | | 3, 5, 6 |
| Mute On | 2 | 13 | 1, 3 | 4, 6, 8, 12 | 5, 9, 10 | | 7, 11 |
| Mute Off | 1 | 3, 4, 13 | 8, 10, 11 | 2, 6 | 7 | | 5, 9, 12 |
| Volume Up | 1 | 6 | 8 | 9 | 4, 12, 13 | 2, 3, 7, 10 | 5, 11 |
| Volume Down | 12 | 2 | 3, 5 | 9 | 1, 11 | 4, 8, 10 | 6, 7, 13 |
| Activate | 6 | 1 | 4 | 11 | 5, 7 | 2, 9, 10 | 3, 8, 12, 13 |

Table 13 shows the total rankings for each sound sample across all command functions. Each rank was given a weighting in accordance with its popularity ($1^{st}$ = 6 pts, $2^{nd}$ = 5 pts, $3^{rd}$ = 4 pts, $4^{th}$ = 3 pts, $5^{th}$ = 2 pts, $6^{th}$ = 1 pt, *No Appearance* = 0 pts), which were then summed to find a total score for each sound sample. The higher a given sample's total score, the greater its popularity. It can be seen that some sounds were significantly more popular than others (e.g. sample 1 vs sample 7), indicating a strong preference for certain types of sounds across all command functions. Further conclusions are drawn about these rankings in the following section.

Table 13. Sound Sample Rankings Across All Command Functions

| | | **Sample 1** | **Sample 2** | **Sample 3** | **Sample 4** | **Sample 5** | **Sample 6** | **Sample 7** | **Sample 8** | **Sample 9** | **Sample 10** | **Sample 11** | **Sample 12** | **Sample 13** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Rank** | 1st | 4 | 1 | 1 | 2 | 2 | 1 | 0 | 0 | 0 | 1 | 2 | 2 | 0 |
| | 2nd | 4 | 3 | 4 | 2 | 1 | 4 | 1 | 3 | 1 | 0 | 0 | 3 | 3 |
| | 3rd | 2 | 2 | 4 | 4 | 3 | 2 | 1 | 4 | 1 | 2 | 6 | 3 | 2 |
| | 4th | 3 | 2 | 1 | 4 | 2 | 3 | 2 | 4 | 6 | 1 | 2 | 3 | 1 |
| | 5th | 2 | 3 | 2 | 2 | 3 | 1 | 4 | 1 | 3 | 5 | 2 | 2 | 5 |
| | 6th | 0 | 2 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 3 | 0 | 0 | 0 |
| | Total Score | 65 | 43 | 50 | 55 | 41 | 45 | 24 | 46 | 34 | 30 | 46 | 52 | 36 |

Table 14 shows the overall rankings for all 13 sound samples based on their total scores given in Table 13. There was only one tie for $5^{th}$ place between samples 8 and 11. Further conclusions are drawn about these rankings in the following section.

Table 14. Sound Sample Overall Rankings

| **Rank** | **Sample** |
|---:|---|
| 1st | Sample 1 |
| 2nd | Sample 4 |
| 3rd | Sample 12 |
| 4th | Sample 3 |
| 5th | Sample 8 & Sample 11 |
| 6th | Sample 6 |
| 7th | Sample 2 |
| 8th | Sample 5 |
| 9th | Sample 13 |
| 10th | Sample 9 |
| 11th | Sample 10 |
| 12th | Sample 7 |

#### 4.4.10. Command Function / Sample Sound Grouping

For the given data, the command functions were grouped for two, three, and four sample sounds. Functions with highly positive correlation coefficients were grouped together, while functions with highly negative correlation coefficients were kept separate. Functions with little or no correlation to any other functions were either grouped to form their own set (three and four sounds) or added arbitrarily to other groups (two sounds). Additionally, each group was assigned a sample sound which corresponded to the most popular choice for all of the command functions within that group. The process used to assign these sounds was similar to the process used in the previous section to rank the sample sounds.

Table 15 – Table 17 show the command function groups for two, three, and four sounds, assuming the number of available sounds is two, three, or four. Grouping of highly correlated command functions tended to follow logical reasoning: functions which one may consider "affirmative" or having to do with an addition of some sort are

correlated to one another (e.g. *volume up* and *on*), as do functions which one may consider "negative" or having to do with a reduction of some sort (e.g. *volume down* and *off*).

Table 15. Command Function Groups – Four Sounds

| **Group 1** | **Group 2** | **Group 3** | **Group 4** |
|---|---|---|---|
| Set + | Set - | Answer Call | End Call |
| Volume Up | Off | Next Track | Media Mode |
| Activate | Volume Down | | Play/Pause |
| On | Previous Track | | Mute On |
| | | | Mute Off |

Table 16. Command Function Groups – Three Sounds

| **Group 1** | **Group 2** | **Group 3** |
|---|---|---|
| Set + | Set - | Answer Call |
| Volume Up | Off | End Call |
| Activate | Volume Down | Media Mode |
| On | Previous Track | Play/Pause |
| Next Track | | Mute On |
| | | Mute Off |

Table 17. Command Function Groups – Two Sounds

| **Group 1** | **Group 2** |
|---|---|
| Set + | Set - |
| Volume Up | Off |
| Activate | Volume Down |
| On | Previous Track |
| Answer Call | End Call |
| Next Track | Mute Off |
| Mute On | |
| Play/Pause | |
| Media Mode | |

Table 18 summarizes which sample sounds would be preferred to be used for each command function group based on overall popularity. The same five sample sounds ended up being assigned multiple times: sample 1, 3, 4, 6, and 12, which also exist within the top 6 places of the overall sample sound ranking in Table 13. An additional note of interest is the source of these four sounds themselves. All but one of the sounds were synthetically generated. The exception is sample 12, which is the recorded pack 2, switch 1 click. This suggests that users in general have a preference for synthetically generated sounds over the sounds naturally produced by the physical movement of push button switches.

Table 18. Summary of Group Sample Sounds

| **Number of Sounds** | **Group** | **Sample Sound** | **Description** |
|---|---|---|---|
| Four | 1 | Sample 6 | High rising tonal cue |
| | 2 | Sample 12 | Pack 2, switch 1 click |
| | 3 | Sample 4 | Low rising tonal cue |
| | 4 | Sample 1 | 2 kHz cue |
| Three | 1 | Sample 6 | High rising tonal cue |
| | 2 | Sample 12 | Pack 2 switch 1 click |
| | 3 | Sample 1 | 2 kHz cue |
| Two | 1 | Sample 1 | 2 kHz cue |
| | 2 | Sample 3 | Low falling tonal cue |

To acquire these results, Table 13 was reproduced for each group using only the command functions found in that group rather than the entire set. Each rank was again given a weighting in accordance with its popularity (*1$^{st}$* = 6 pts, *2$^{nd}$* = 5 pts, *3$^{rd}$* = 4 pts, *4$^{th}$* = 3 pts, *5$^{th}$* = 2 pts, *6$^{th}$* = 1 pt, *No appearance* = 0 pts), which were then summed to find a total score for each sound sample. The higher a given sample's total score, the greater its popularity. For each group, the sample sound with the highest total score was chosen. If one sample sound was ranked highest for more than one group, the group with the highest overall score was assigned that sound, and the other group's second choice was chosen. For example, sample 1 was the top ranked sound for the *four sounds, group 1* set with a total score of 23, as well as the top ranked sound for the *four sounds, group 3* set with a total score of 24. In that case, sample 1 was assigned to group 3 since it has the higher score, and group 1 was assigned its second highest ranked sound, sample 6. The sound sample ranking tables for each group can be found in Appendix G.

The five sounds chosen for grouping exhibited the following overall rankings: sample 1 was ranked 1$^{st}$ overall, sample 3 was ranked 4$^{th}$ overall, sample 4 was ranked 2$^{nd}$ overall, sample 6 was ranked 6$^{th}$ overall, and sample 12 was ranked 3$^{rd}$ overall. The majority of the command functions within a given group exhibited a high overall ranking for the sample sound chosen for that group. For example, the *four sounds, group 1* set was assigned sample 6, which contains the *Set +*, *Volume Up*, *Activate*, and *On* command functions. Upon returning to the original results of part 2 of the sound jury test, it can be seen in Figure 32 that those four command functions corresponded to the top four choices for sample 6. The same can be said of several other command function groups, such as *four sounds, group 2* in Figure 38, *four sounds, group 3* in Figure 30, *three sounds, group 1* in Figure 32, *three sounds, group 2* in Figure 38, and *two sounds, group 2* in Figure 29. Additionally, those sets which did not have a perfect ranking matchup still exhibited high favorability in their rankings. These rankings provide validity to the claim that they are indeed the ideal sounds for each group of command functions, given the available set of thirteen sample sounds.

Table 19 shows the average correlation coefficients for each group. Most groups have coefficients above the critical coefficient (0.355) with the exception of *four sounds, group 3* and *three sounds, group 3*, as these functions did not have strong correlations with anything. However, it is also worth noting that no grouping resulted in negative correlation. It can also be seen that the remaining groups from the *four sounds* set had much higher correlation coefficients (0.806 and 0.671) than those from the *two sounds* set (0.405 and 0.520). Thus, even though one of the four groups resulted in lower correlation than they would if they had been split into two groups, the remaining three resulted in higher correlation than they otherwise would have had. The same can be said about the *three sounds* set as well. Hence, it would still be advantageous to group the command functions for three or four sounds. Detailed tables showing each function's correlation coefficient as compared to the other functions in its group can be found in Appendix H.

Table 19. Average Correlation Coefficients – Command Function Groups

| | **Four Sounds** | **Three Sounds** | **Two Sounds** |
|---|---|---|---|
| Group 1 | 0.806 | 0.806 | 0.405 |
| Group 2 | 0.671 | 0.671 | 0.520 |
| Group 3 | 0.620 | 0.145 | |
| Group 4 | 0.016 | | |

Figure 83 and Figure 84 show the findings of how different groups of command functions should be mapped to different qualitative types of sounds. Expanding on the sample sound assignments shown in Table 18, affirmative command functions should be matched with either high frequency beeps or beeps that increase in pitch. For example, for the *set +* command, which prompts the cruise control to increase the speed of the vehicle by one mile per hour, people associate the increase in speed with an increase in sound pitch. Also, negative command functions should be matched with either clicks or low frequency beeps that decrease in pitch. For example, for the *volume down* command, which prompts the media volume to decrease by 1 unit, people associate the decrease in volume with a decrease in sound pitch.

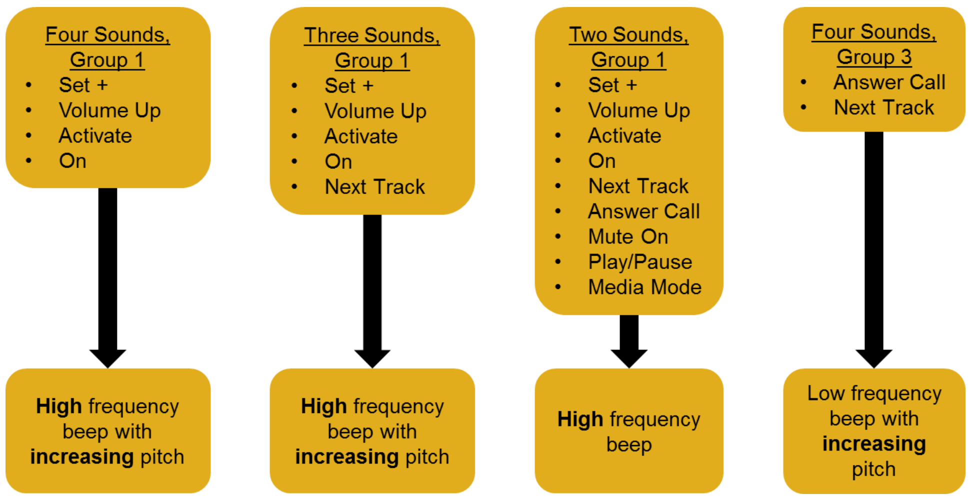



Figure 83. Affirmative Command Function Qualitative Sound Mapping

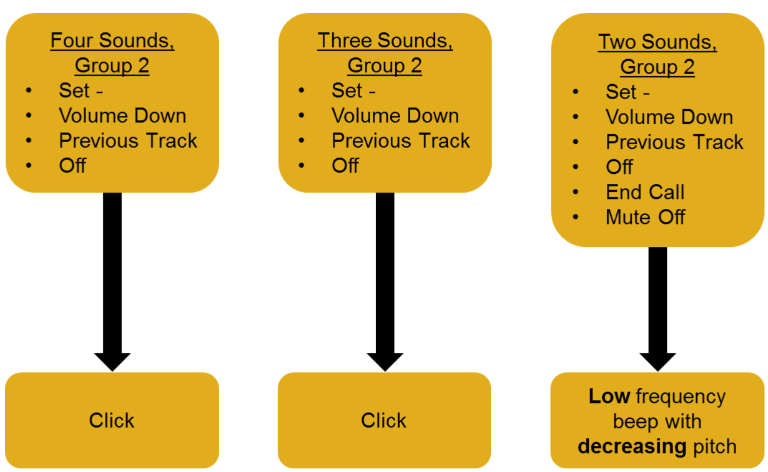



Figure 84. Negative Command Function Qualitative Sound Mapping

# CHAPTER 5. CONCLUSIONS

## 5.1. Switch Pack Output Measurement

These tests were conducted for the purpose of quantifying the audio signatures of the switch packs provided by the sponsor company as they are manually activated by a user. WAV files from five of the recorded switches ended up being used in the sound jury test as sample sounds, though only one of them (Pack 2, Switch 1) was identified as an ideal sound to be assigned to a group of command functions. Although the switch packs are used in steering wheels from current-model automotive vehicles, the results of the sound jury found that users prefer synthetically generated sounds instead. This provides validity to the value of using a haptic user interface system rather than switch packs to initiate commands. The need to perfectly mimic the sound of physical push buttons no longer becomes a stringent requirement, as generating synthetic sounds can be done by the exciter tested in this thesis.

## 5.2. Haptic Board Output Measurement

This test was conducted for the purpose of determining the transfer function and output capabilities of the haptic board provided by the sponsor company. The internal processing problem that was identified at volume settings above 35 indicates that the system would not be able to successfully reproduce button sounds at high loudness levels. However, the sound jury test did not investigate the successful initiation of command functions as they related to the quantitative loudness of sounds. Thus, it cannot be immediately discerned whether this may pose a problem, as it is plausible that sufficiently loud sounds would not be desirable to drivers anyway.

## 5.3. Exciter Output Measurement

This test was conducted for the purpose of determining the transfer function and output capabilities of the exciter provided by the sponsor company. Due to its small size, generating signals which are audible to the human ear in an automotive environment with frequency components in the range of the 190 Hz resonance frequency is not practical. For this reason, the resonance frequency is not deemed to be of concern when designing input waveforms to be played by the exciter.

## 5.4. Sound Jury

### 5.4.1. Model Performance

This test was conducted for the purpose of identifying the subjective psychoacoustic metrics of thirteen sample sounds, as well as matching these sounds to ideal command functions. The statistical variance tests conducted on the data (Levene test, ANOVA, Kruskal-Wallis test) revealed that the results gathered did not happen by chance, which allows for meaningful conclusions to be drawn from the data. This was further supported by the post hoc T-test and Wilcoxon signed rank test, which also showed the majority of the psychoacoustic metric and command function data to be statistically significant.

In an automotive environment where each command function is not activated by its own individual push button, but rather by a steering wheel mounted haptic user interface, it was deemed desirable to assign a single button sound to more than one command function. The results gathered from the sound jury test found this to be a possibility while still retaining high correlation between the command functions in each group. Thus, using these function group and sample sound pairs in a steering wheel mounted haptic user interface would effectively indicate successful initiation of the intended functions.

A preference was found for synthetically generated button sounds over those which are created by the physical movement of mechanical button switches. Thus, the ability to perfectly mimic the sound of psyical button sounds is not a stringent reqirement when they are being replicated in a haptic user interface. Furthermore, it was found through the mapping of groups of command functions to sounds that people associate the behavior of a function with the behavior of that function's sound. Affirmative functions should be matched with either high frequency beeps or beeps that increase in pitch. Negative functions should be matched with either clicks or low frequency beeps that decrease in pitch. Thus, people expect the qualitative characteristics of the sounds themselves to match the characteristics of the functions they're initiating.

These findings indicate a steering wheel mounted haptic user interface that contains the haptic board and exciter examined in this thesis to be an effective method of providing control over multiple vehicle functions to the driver without the use of physical push buttons. There is no need to have a unique button sound for each individual command function; should hardware constraints limit the number of sounds that are able to be stored, a subset of sounds may successfully be used instead. This would improve not only customer satisfaction, but would also serve to increase driver safety. Drivers would not have to take their eyes off the road to confirm successful initiation of the intended function, as the sound itself would suffice in doing so.

### 5.4.2. Recommendations for Future Work

As the jury test was conducted on the Georgia Tech campus, only Georgia Tech students were used, which may have introduced a bias in the results. In the future, a subsequent study could potentially be conducted using a wider range of demographics to remove any age or gender related prejudice in the results.

The work done for the sound jury in this thesis focused primarily on the mapping of groups of command functions to individual sample sounds, as well as to subjective psychoacoustic metrics. Further work could be done to map those subjective metrics to their objective counterparts, which would allow one to synthetically generate the objectively "ideal" sound for each command function group based on those objective metrics.

The sample sounds used in the study were not chosen to be universal (i.e. not containing every possible button sound), but rather to cover a general wide range of sound characteristics. As such, there may have been specific sounds or specific types of sounds that would have elicited different results had they been included. However, other limitations (e.g. ensuring an excessive test duration did not fatigue participants) encouraged the use of a smaller, rather than universal, set of sample sounds. In the future, a second test could be conducted that contains a more concise set of sample sounds which are delicately chosen based on results gathered in the work performed in this thesis.

# APPENDICES

# Appendix A. Sample Sound Time and Frequency Spectra

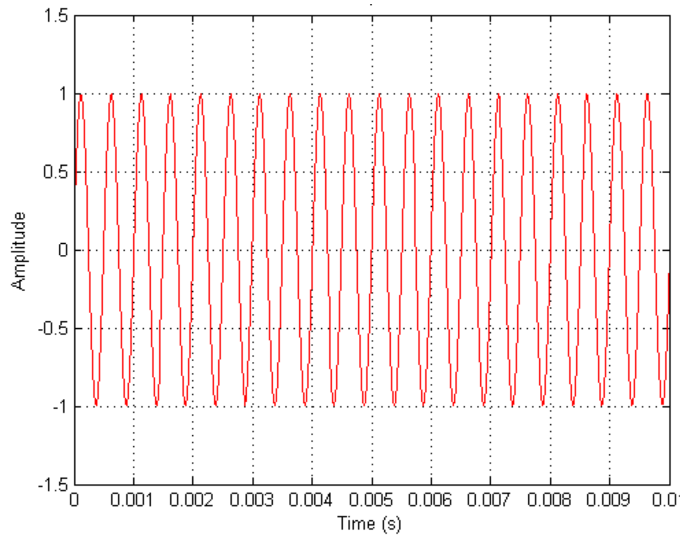



Figure 85. Sample 1 Time Spectrum

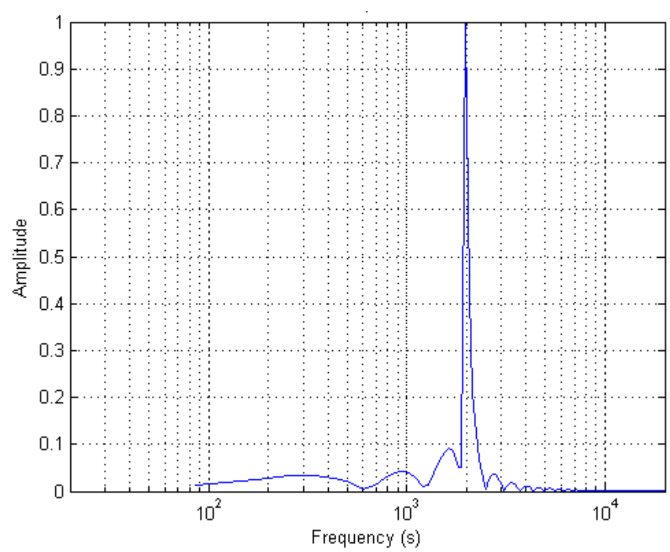



Figure 86. Sample 1 Frequency Spectrum

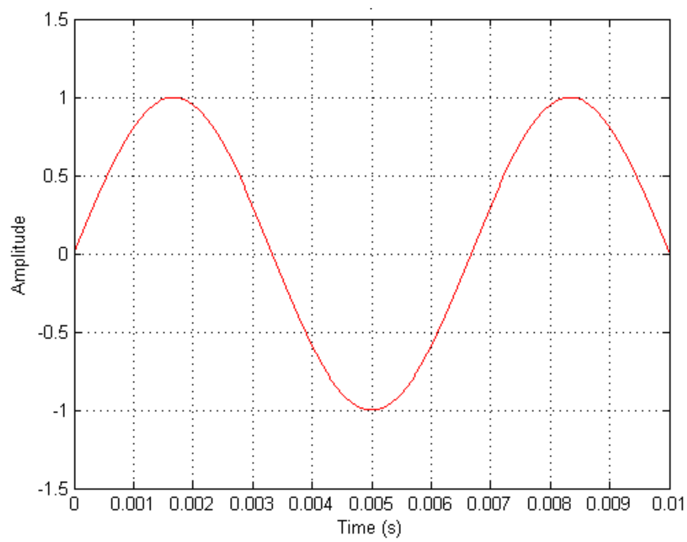



Figure 87. Sample 2 Time Spectrum

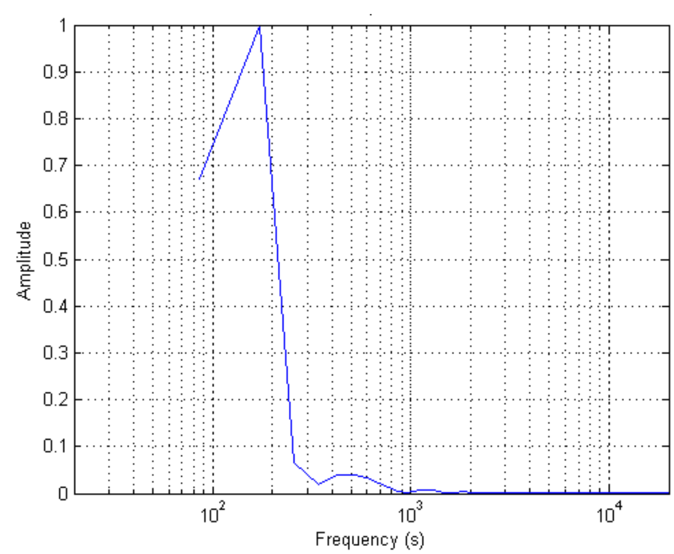



Figure 88. Sample 2 Frequency Spectrum

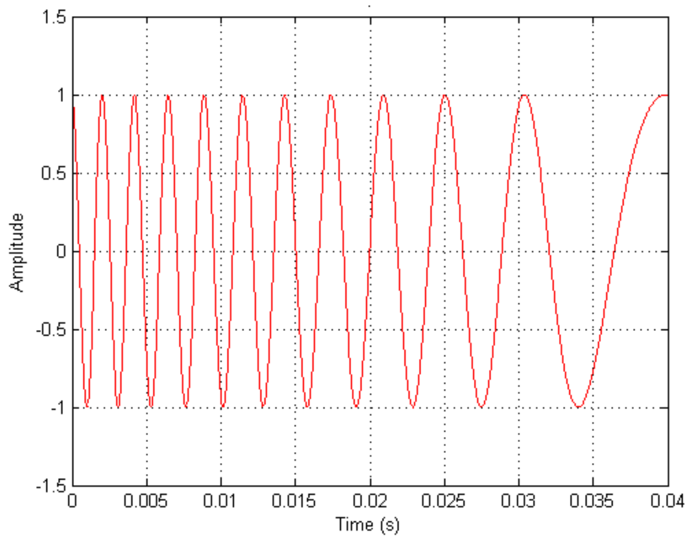



Figure 89. Sample 3 Time Spectrum

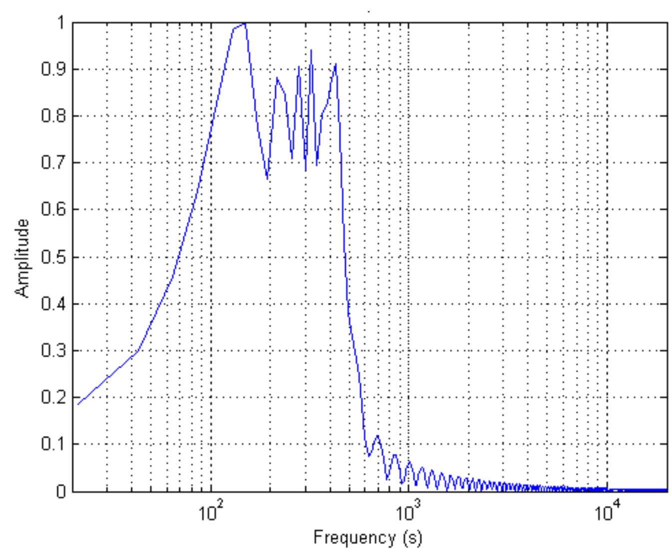



Figure 90. Sample 3 Frequency Spectrum

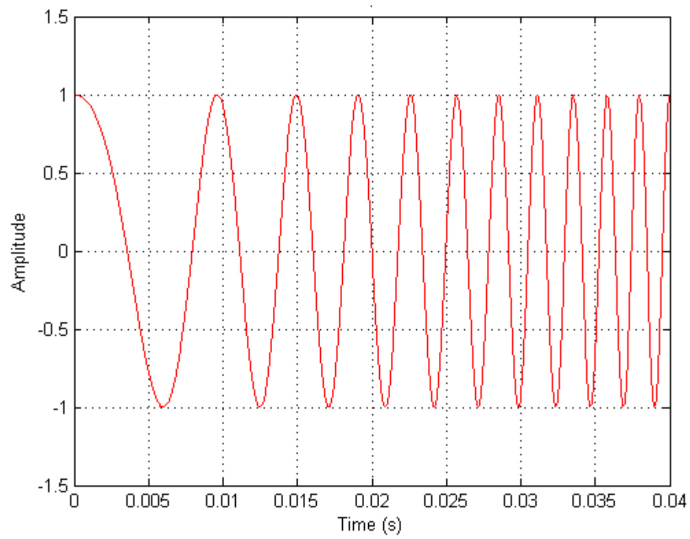



Figure 91. Sample 4 Time Spectrum

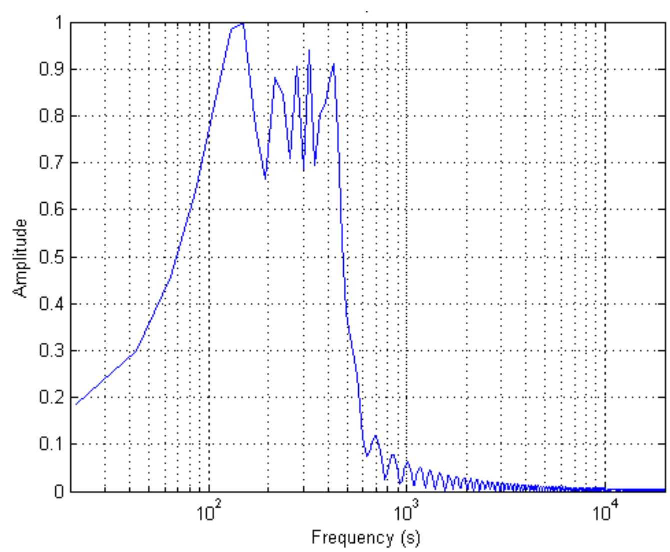



Figure 92. Sample 4 Frequency Spectrum

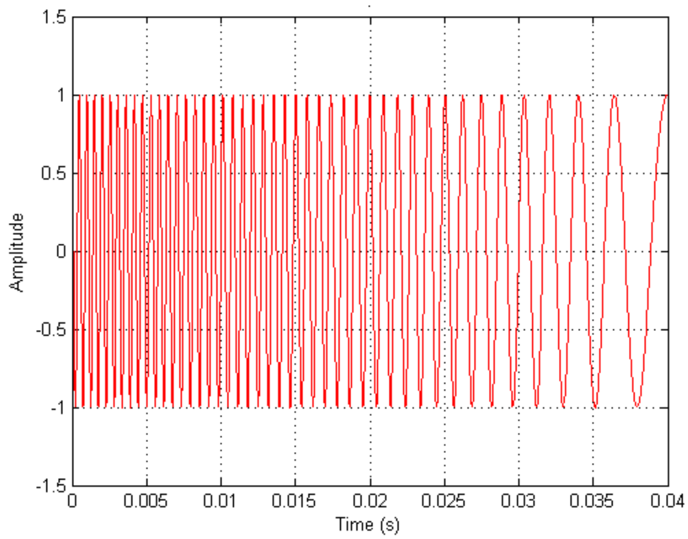



Figure 93. Sample 5 Time Spectrum

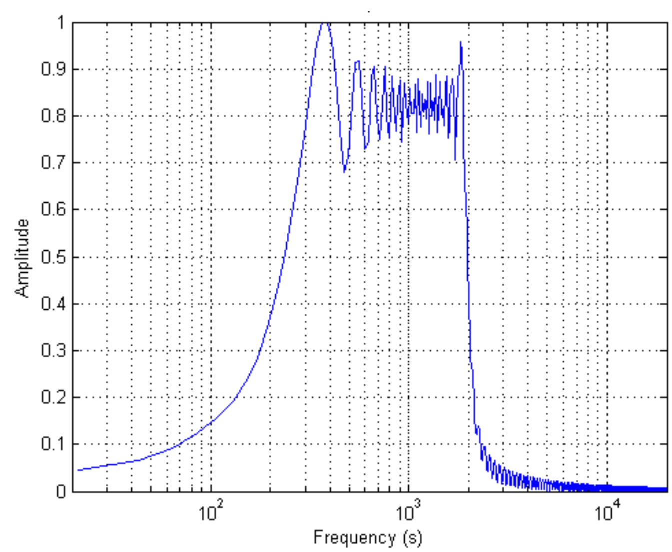



Figure 94. Sample 5 Frequency Spectrum

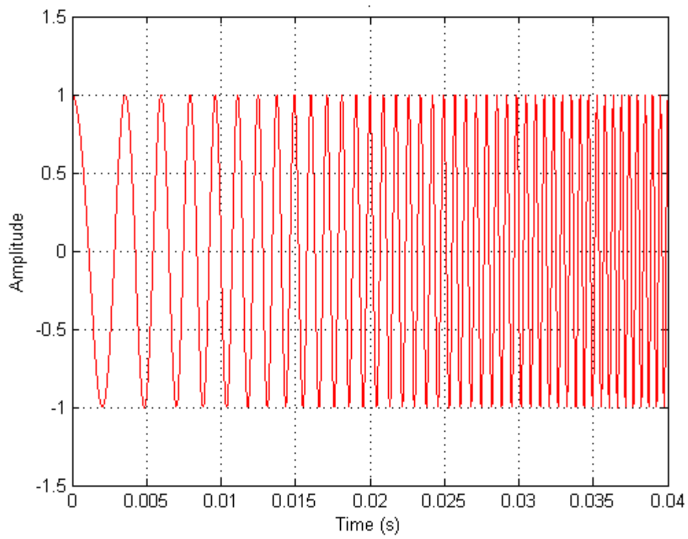



Figure 95. Sample 6 Time Spectrum

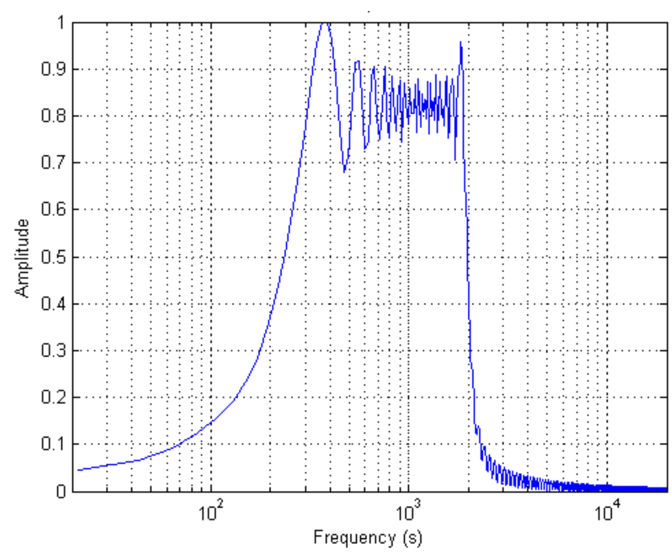



Figure 96. Sample 6 Frequency Spectrum

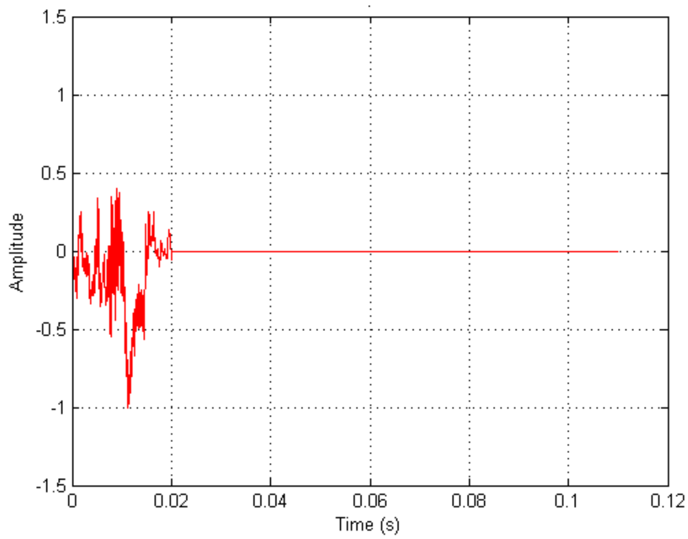



Figure 97. Sample 7 Time Spectrum

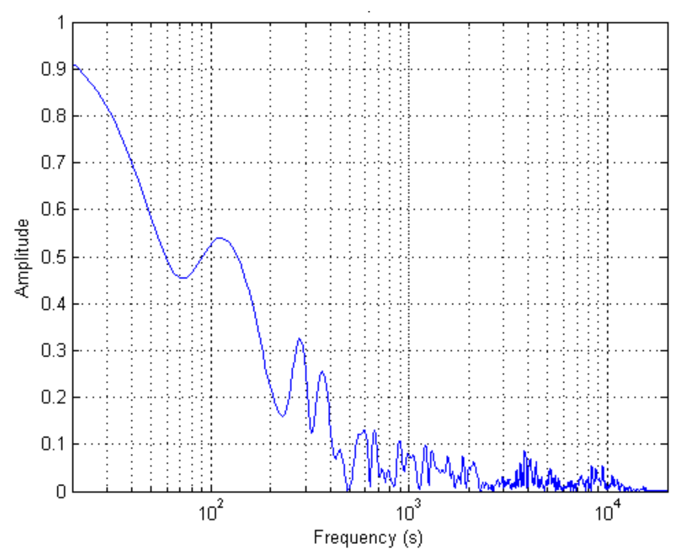



Figure 98. Sample 7 Frequency Spectrum

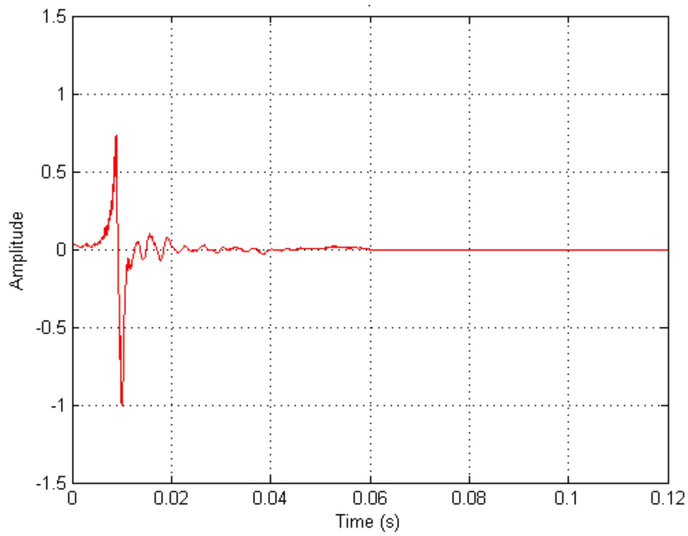



Figure 99. Sample 8 Time Spectrum

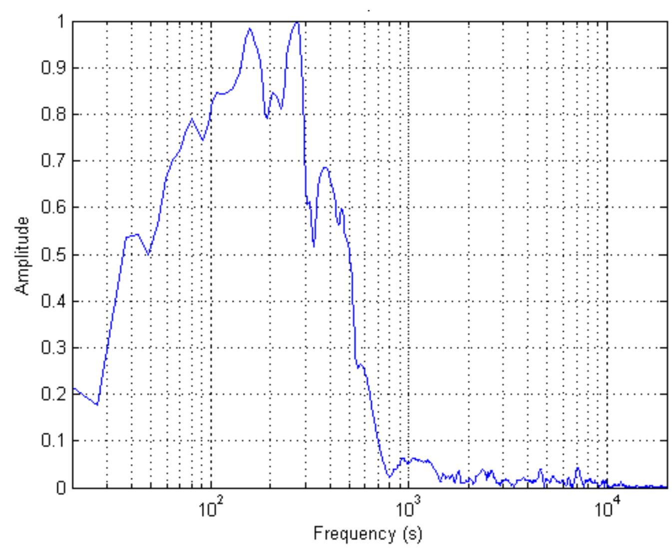



Figure 100. Sample 8 Frequency Spectrum

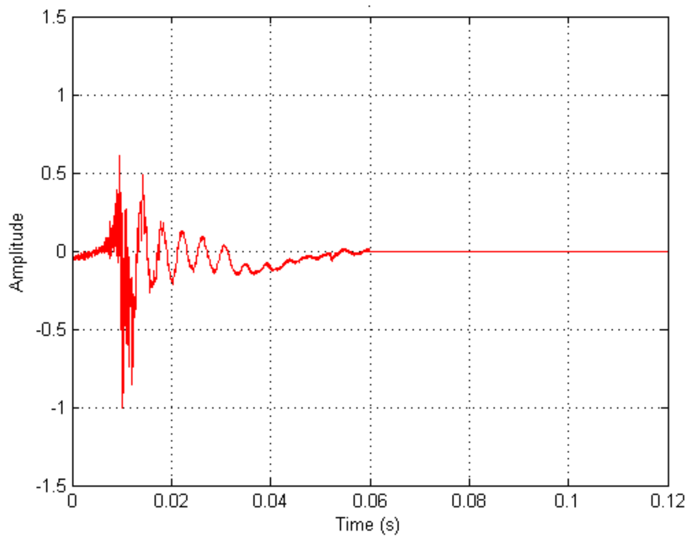



Figure 101. Sample 9 Time Spectrum

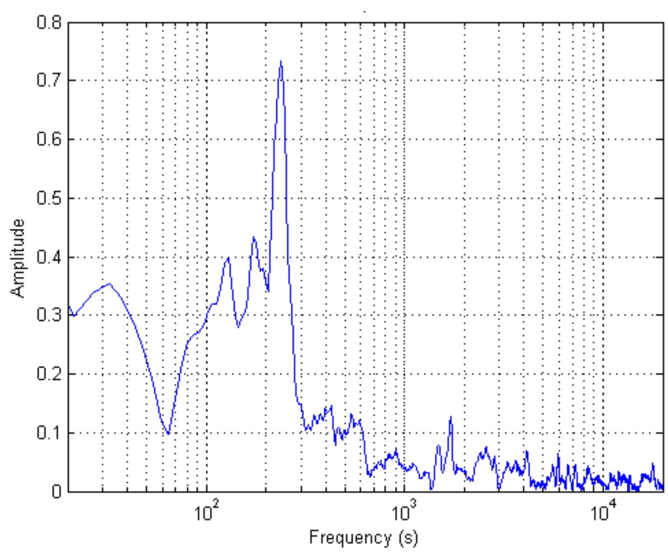



Figure 102. Sample 9 Frequency Spectrum

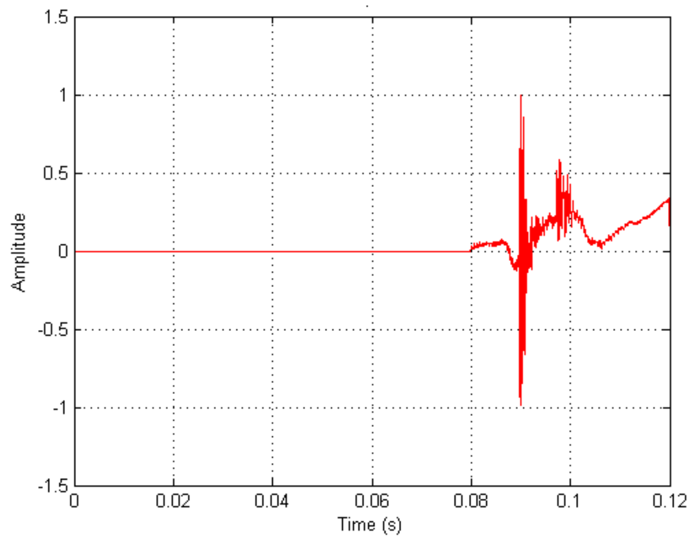



Figure 103. Sample 10 Time Spectrum

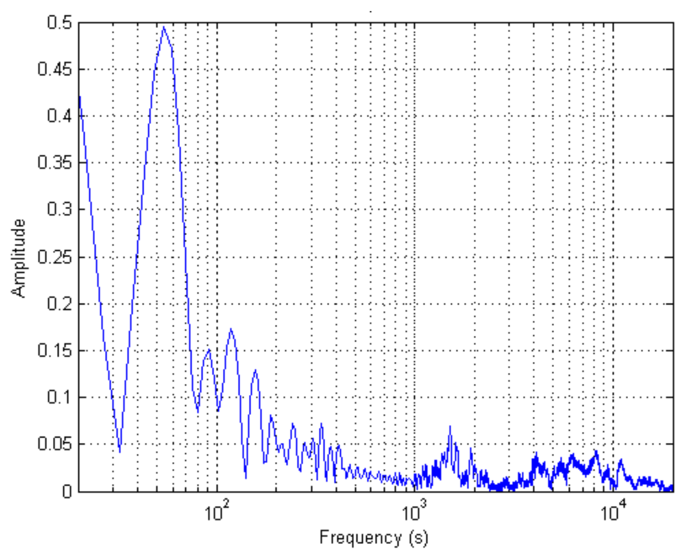



Figure 104. Sample 10 Frequency Spectrum

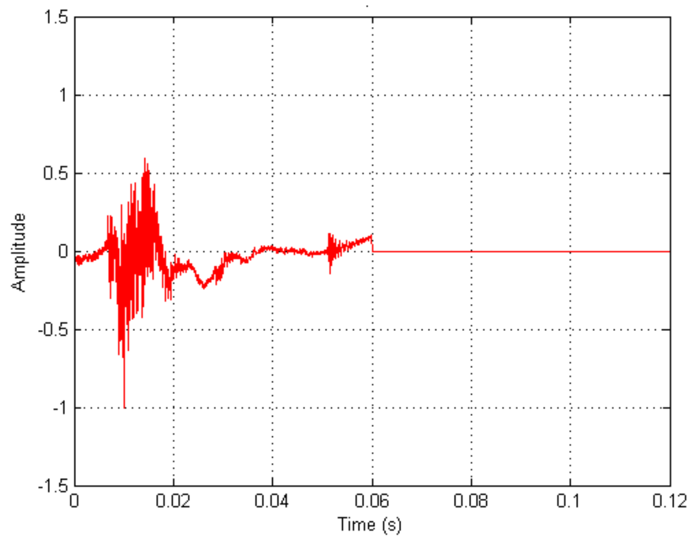



Figure 105. Sample 11 Time Spectrum

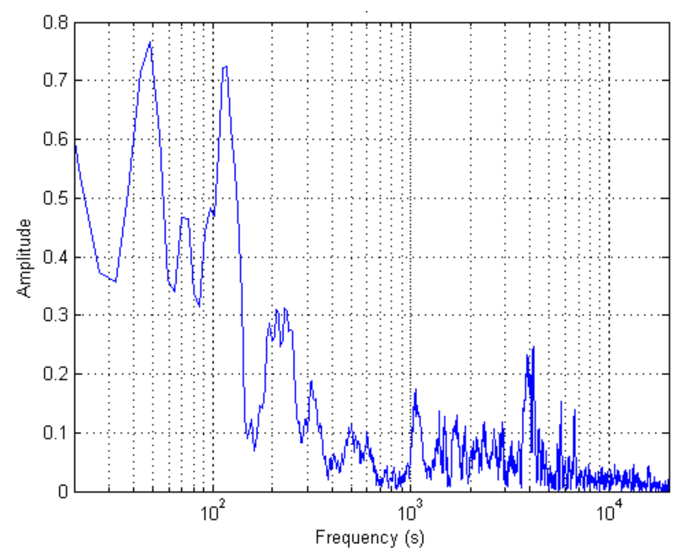



Figure 106. Sample 11 Frequency Spectrum

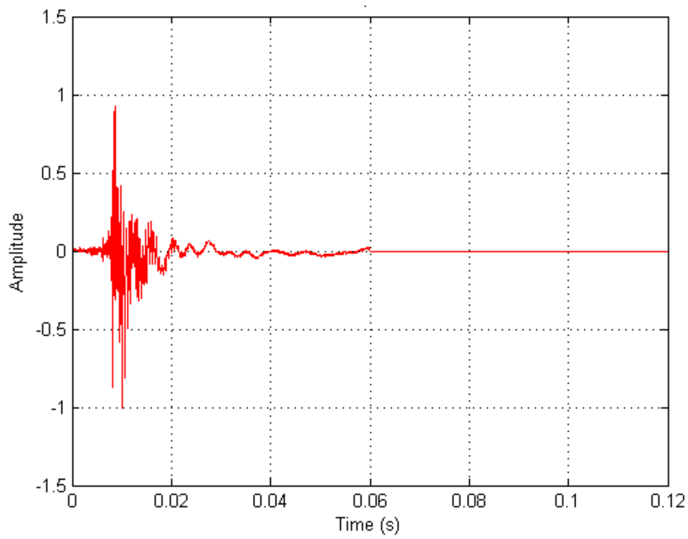



Figure 107. Sample 12 Time Spectrum

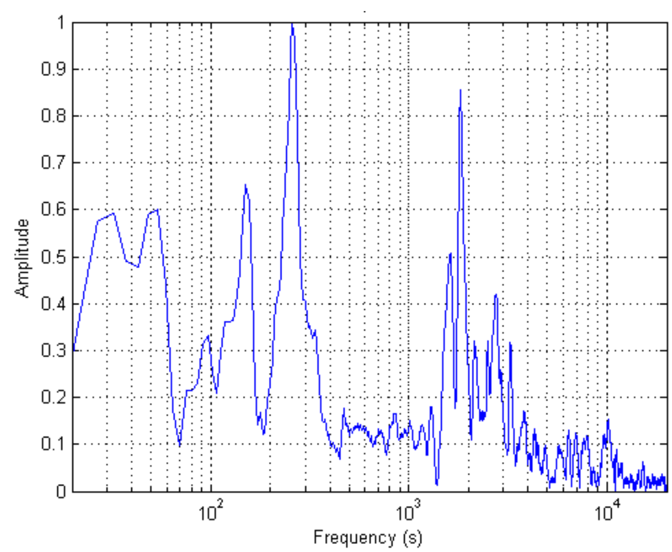



Figure 108. Sample 12 Frequency Spectrum

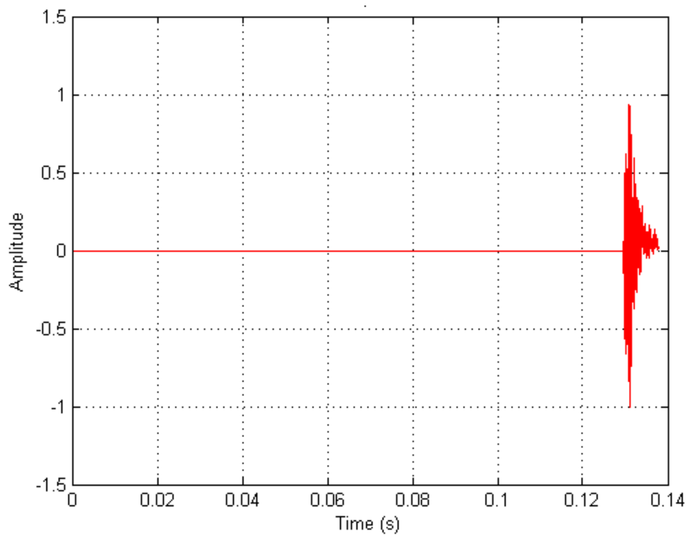



Figure 109. Sample 13 Time Spectrum

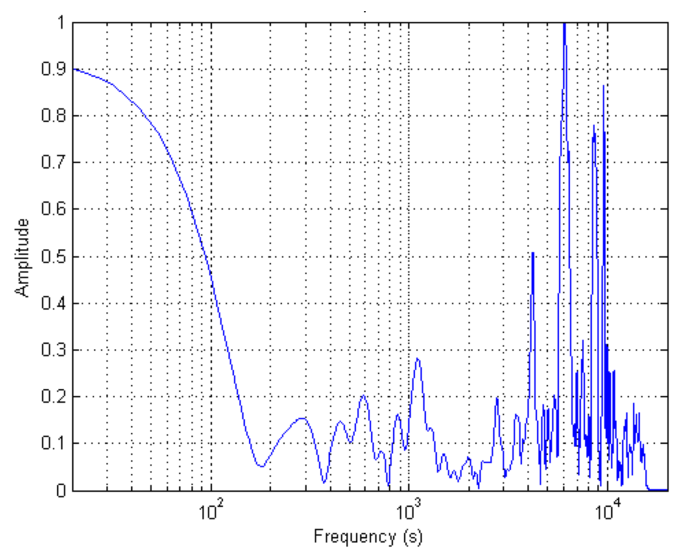



Figure 110. Sample 13 Frequency Spectrum

## Appendix B. ANOVA Results

Table 20. ANOVA Results: *Long*

ANOVA: Single Factor

| DESCRIPTION | | | | | Alpha | 0.05 | | |
|---|---|---|---|---|---|---|---|---|
| *Groups* | *Count* | *Sum* | *Mean* | *Variance* | *SS* | *Std Err* | *Lower* | *Upper* |
| Sample 1 | 31 | 110 | 3.548387097 | 0.922580645 | 27.67741935 | 0.179872506 | 3.181038433 | 3.915735761 |
| Sample 2 | 31 | 43 | 1.387096774 | 0.84516129 | 25.35483871 | 0.179872506 | 1.01974811 | 1.754445438 |
| Sample 3 | 31 | 76 | 2.451612903 | 0.855913978 | 25.67741935 | 0.179872506 | 2.084264239 | 2.818961567 |
| Sample 4 | 31 | 66 | 2.129032258 | 0.782795699 | 23.48387097 | 0.179872506 | 1.761683594 | 2.496380922 |
| Sample 5 | 31 | 126 | 4.064516129 | 0.395698925 | 11.87096774 | 0.179872506 | 3.697167465 | 4.431864793 |
| Sample 6 | 31 | 108 | 3.483870968 | 1.124731183 | 33.74193548 | 0.179872506 | 3.116522304 | 3.851219632 |
| Sample 7 | 31 | 65 | 2.096774194 | 1.490322581 | 44.70967742 | 0.179872506 | 1.72942553 | 2.464122858 |
| Sample 8 | 31 | 52 | 1.677419355 | 0.625806452 | 18.77419355 | 0.179872506 | 1.310070691 | 2.044768019 |
| Sample 9 | 31 | 76 | 2.451612903 | 1.189247312 | 35.67741935 | 0.179872506 | 2.084264239 | 2.818961567 |
| Sample 10 | 31 | 78 | 2.516129032 | 1.191397849 | 35.74193548 | 0.179872506 | 2.148780368 | 2.883477696 |
| Sample 11 | 31 | 60 | 1.935483871 | 1.195698925 | 35.87096774 | 0.179872506 | 1.568135207 | 2.302832535 |
| Sample 12 | 31 | 65 | 2.096774194 | 1.356989247 | 40.70967742 | 0.179872506 | 1.72942553 | 2.464122858 |
| Sample 13 | 31 | 126 | 4.064516129 | 1.062365591 | 31.87096774 | 0.179872506 | 3.697167465 | 4.431864793 |

ANOVA

| *Sources* | *SS* | *df* | *MS* | *F* | *P value* | *F crit* | *RMSSE* | *Omega Sq* |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 294.8933002 | 12 | 24.57444169 | 24.50148441 | 1.01139E-40 | 1.777036001 | 0.88902782 | 0.411694131 |
| Within Groups | 391.1612903 | 390 | 1.002977667 | | | | | |
| Total | 686.0545906 | 402 | 1.706603459 | | | | | |

Table 21. ANOVA Results: *High Frequency*

ANOVA: Single Factor

| DESCRIPTION | | | | | Alpha | 0.05 | | |
|---|---|---|---|---|---|---|---|---|
| *Groups* | *Count* | *Sum* | *Mean* | *Variance* | *SS* | *Std Err* | *Lower* | *Upper* |
| Sample 1 | 26 | 39 | 1.5 | 0.74 | 18.5 | 0.180935689 | 1.127355973 | 1.872644027 |
| Sample 2 | 27 | 39 | 1.444444444 | 0.333333333 | 8.666666667 | 0.177553415 | 1.079478173 | 1.809410716 |
| Sample 3 | 30 | 92 | 3.066666667 | 1.236781609 | 35.86666667 | 0.168441959 | 2.722164178 | 3.411169155 |
| Sample 4 | 29 | 87 | 3 | 0.714285714 | 20 | 0.171321518 | 2.64906378 | 3.35093622 |
| Sample 5 | 29 | 77 | 2.655172414 | 1.162561576 | 32.55172414 | 0.171321518 | 2.304236193 | 3.006108634 |
| Sample 6 | 28 | 73 | 2.607142857 | 0.988095238 | 26.67857143 | 0.174353992 | 2.249398015 | 2.964887699 |
| Sample 7 | 29 | 48 | 1.655172414 | 0.948275862 | 26.55172414 | 0.171321518 | 1.304236193 | 2.006108634 |
| Sample 8 | 30 | 59 | 1.966666667 | 0.791954023 | 22.96666667 | 0.168441959 | 1.622164178 | 2.311169155 |
| Sample 9 | 29 | 54 | 1.862068966 | 1.051724138 | 29.44827586 | 0.171321518 | 1.511132745 | 2.213005186 |
| Sample 10 | 30 | 70 | 2.333333333 | 1.057471264 | 30.66666667 | 0.168441959 | 1.988830845 | 2.677835822 |
| Sample 11 | 29 | 59 | 2.034482759 | 1.034482759 | 28.96551724 | 0.171321518 | 1.683546538 | 2.385418979 |
| Sample 12 | 29 | 44 | 1.517241379 | 0.330049261 | 9.24137931 | 0.171321518 | 1.166305159 | 1.8681776 |
| Sample 13 | 29 | 45 | 1.551724138 | 0.613300493 | 17.17241379 | 0.171321518 | 1.200787918 | 1.902660358 |

ANOVA

| *Sources* | *SS* | *df* | *MS* | *F* | *P value* | *F crit* | *RMSSE* | *Omega Sq* |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 116.8627649 | 12 | 9.738563738 | 11.44123977 | 1.60567E-19 | 1.779041669 | 0.631238343 | 0.250943647 |
| Within Groups | 307.2762726 | 361 | 0.85118081 | | | | | |
| Total | 424.1390374 | 373 | 1.137101977 | | | | | |

Table 22. ANOVA Results: *Intense*

ANOVA: Single Factor

DESCRIPTION Alpha 0.05

| Groups | Count | Sum | Mean | Variance | SS | Std Err | Lower | Upper |
|---|---|---|---|---|---|---|---|---|
| Sample 1 | 30 | 70 | 2.333333333 | 1.333333333 | 38.66666667 | 0.178576332 | 1.968103726 | 2.69856294 |
| Sample 2 | 31 | 49 | 1.580645161 | 0.51827957 | 15.5483871 | 0.175672458 | 1.221874139 | 1.939416183 |
| Sample 3 | 31 | 79 | 2.548387097 | 0.922580645 | 27.67741935 | 0.175672458 | 2.189616075 | 2.907158119 |
| Sample 4 | 31 | 71 | 2.290322581 | 1.012903226 | 30.38709677 | 0.175672458 | 1.931551558 | 2.649093603 |
| Sample 5 | 31 | 104 | 3.35483871 | 0.969892473 | 29.09677419 | 0.175672458 | 2.996067688 | 3.713609732 |
| Sample 6 | 31 | 98 | 3.161290323 | 1.47311828 | 44.19354839 | 0.175672458 | 2.8025193 | 3.520061345 |
| Sample 7 | 31 | 53 | 1.709677419 | 1.012903226 | 30.38709677 | 0.175672458 | 1.350906397 | 2.068448442 |
| Sample 8 | 31 | 53 | 1.709677419 | 0.812903226 | 24.38709677 | 0.175672458 | 1.350906397 | 2.068448442 |
| Sample 9 | 31 | 59 | 1.903225806 | 0.623655914 | 18.70967742 | 0.175672458 | 1.544454784 | 2.261996829 |
| Sample 10 | 31 | 60 | 1.935483871 | 0.662365591 | 19.87096774 | 0.175672458 | 1.576712849 | 2.294254893 |
| Sample 11 | 31 | 54 | 1.741935484 | 0.797849462 | 23.93548387 | 0.175672458 | 1.383164462 | 2.100706506 |
| Sample 12 | 31 | 50 | 1.612903226 | 0.778494624 | 23.35483871 | 0.175672458 | 1.254132204 | 1.971674248 |
| Sample 13 | 31 | 85 | 2.741935484 | 1.531182796 | 45.93548387 | 0.175672458 | 2.383164462 | 3.100706506 |

ANOVA

| Sources | SS | df | MS | F | P value | F crit | RMSSE | Omega Sq |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 132.5285668 | 12 | 11.04404724 | 11.54407677 | 6.24052E-20 | 1.777100165 | 0.610277227 | 0.239398292 |
| Within Groups | 372.1505376 | 389 | 0.956685187 | | | | | |
| Total | 504.6791045 | 401 | 1.258551383 | | | | | |

Table 23. ANOVA Results: *Tonal*

ANOVA: Single Factor

DESCRIPTION Alpha 0.05

| *Groups* | *Count* | *Sum* | *Mean* | *Variance* | *SS* | *Std Err* | *Lower* | *Upper* |
|---|---|---|---|---|---|---|---|---|
| Sample 1 | 31 | 144 | 4.64516129 | 0.23655914 | 7.096774194 | 0.173407533 | 4.291015861 | 4.999306719 |
| Sample 2 | 31 | 128 | 4.129032258 | 0.782795699 | 23.48387097 | 0.173407533 | 3.774886829 | 4.483177687 |
| Sample 3 | 31 | 79 | 2.548387097 | 0.922580645 | 27.67741935 | 0.173407533 | 2.194241668 | 2.902532526 |
| Sample 4 | 31 | 85 | 2.741935484 | 1.064516129 | 31.93548387 | 0.173407533 | 2.387790055 | 3.096080913 |
| Sample 5 | 31 | 100 | 3.225806452 | 1.113978495 | 33.41935484 | 0.173407533 | 2.871661022 | 3.579951881 |
| Sample 6 | 31 | 99 | 3.193548387 | 0.827956989 | 24.83870968 | 0.173407533 | 2.839402958 | 3.547693816 |
| Sample 7 | 31 | 121 | 3.903225806 | 0.956989247 | 28.70967742 | 0.173407533 | 3.549080377 | 4.257371236 |
| Sample 8 | 31 | 104 | 3.35483871 | 1.103225806 | 33.09677419 | 0.173407533 | 3.000693281 | 3.708984139 |
| Sample 9 | 31 | 103 | 3.322580645 | 1.025806452 | 30.77419355 | 0.173407533 | 2.968435216 | 3.676726074 |
| Sample 10 | 31 | 104 | 3.35483871 | 1.303225806 | 39.09677419 | 0.173407533 | 3.000693281 | 3.708984139 |
| Sample 11 | 31 | 95 | 3.064516129 | 1.195698925 | 35.87096774 | 0.173407533 | 2.7103707 | 3.418661558 |
| Sample 12 | 31 | 114 | 3.677419355 | 0.892473118 | 26.77419355 | 0.173407533 | 3.323273926 | 4.031564784 |
| Sample 13 | 31 | 134 | 4.322580645 | 0.692473118 | 20.77419355 | 0.173407533 | 3.968435216 | 4.676726074 |

ANOVA

| *Sources* | *SS* | *df* | *MS* | *F* | *P value* | *F crit* | *RMSSE* | *Omega Sq* |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 139.2009926 | 12 | 11.60008271 | 12.44409938 | 1.55286E-21 | 1.777036001 | 0.633579167 | 0.254158381 |
| Within Groups | 363.5483871 | 390 | 0.932175352 | | | | | |
| Total | 502.7493797 | 402 | 1.250620347 | | | | | |

Table 24. ANOVA Results: *Short*

ANOVA: Single Factor

DESCRIPTION | | | | | Alpha | 0.05 | | |
|---|---|---|---|---|---|---|---|---|---|

| *Groups* | *Count* | *Sum* | *Mean* | *Variance* | *SS* | *Std Err* | *Lower* | *Upper* |
|---|---|---|---|---|---|---|---|---|
| Sample 1 | 19 | 44 | 2.315789474 | 1.228070175 | 22.10526316 | 0.219178296 | 1.85531296 | 2.776265987 |
| Sample 2 | 18 | 33 | 1.833333333 | 0.852941176 | 14.5 | 0.225184293 | 1.358236003 | 2.308430663 |
| Sample 3 | 22 | 58 | 2.636363636 | 1.004329004 | 21.09090909 | 0.203686857 | 2.212773628 | 3.059953645 |
| Sample 4 | 24 | 65 | 2.708333333 | 1.346014493 | 30.95833333 | 0.195015319 | 2.30491341 | 3.111753256 |
| Sample 5 | 24 | 57 | 2.375 | 0.766304348 | 17.625 | 0.195015319 | 1.971580077 | 2.778419923 |
| Sample 6 | 24 | 58 | 2.416666667 | 1.123188406 | 25.83333333 | 0.195015319 | 2.013246744 | 2.82008659 |
| Sample 7 | 24 | 40 | 1.666666667 | 1.188405797 | 27.33333333 | 0.195015319 | 1.263246744 | 2.07008659 |
| Sample 8 | 24 | 37 | 1.541666667 | 0.69384058 | 15.95833333 | 0.195015319 | 1.138246744 | 1.94508659 |
| Sample 9 | 25 | 41 | 1.64 | 0.656666667 | 15.76 | 0.191075209 | 1.245640151 | 2.034359849 |
| Sample 10 | 24 | 45 | 1.875 | 0.89673913 | 20.625 | 0.195015319 | 1.471580077 | 2.278419923 |
| Sample 11 | 25 | 39 | 1.56 | 0.59 | 14.16 | 0.191075209 | 1.165640151 | 1.954359849 |
| Sample 12 | 25 | 40 | 1.6 | 0.833333333 | 20 | 0.191075209 | 1.205640151 | 1.994359849 |
| Sample 13 | 24 | 34 | 1.416666667 | 0.775362319 | 17.83333333 | 0.195015319 | 1.013246744 | 1.82008659 |

ANOVA

| *Sources* | *SS* | *df* | *MS* | *F* | *P value* | *F crit* | *RMSSE* | *Omega Sq* |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 58.65755844 | 12 | 4.88812987 | 5.355426222 | 3.41507E-08 | 1.785770671 | 0.477324404 | 0.147531079 |
| Within Groups | 263.7828389 | 289 | 0.912743387 | | | | | |
| Total | 322.4403974 | 301 | 1.071230556 | | | | | |

Table 25. ANOVA Results: *Loud*

ANOVA: Single Factor

| DESCRIPTION | | | | | Alpha | 0.05 | | |
|---|---|---|---|---|---|---|---|---|
| *Groups* | *Count* | *Sum* | *Mean* | *Variance* | *SS* | *Std Err* | *Lower* | *Upper* |
| Sample 1 | 31 | 68 | 2.193548387 | 0.761290323 | 22.83870968 | 0.153179001 | 1.880715132 | 2.506381642 |
| Sample 2 | 31 | 52 | 1.677419355 | 0.492473118 | 14.77419355 | 0.153179001 | 1.3645861 | 1.99025261 |
| Sample 3 | 31 | 83 | 2.677419355 | 0.692473118 | 20.77419355 | 0.153179001 | 2.3645861 | 2.99025261 |
| Sample 4 | 31 | 81 | 2.612903226 | 0.911827957 | 27.35483871 | 0.153179001 | 2.300069971 | 2.925736481 |
| Sample 5 | 31 | 95 | 3.064516129 | 0.862365591 | 25.87096774 | 0.153179001 | 2.751682874 | 3.377349384 |
| Sample 6 | 31 | 99 | 3.193548387 | 0.894623656 | 26.83870968 | 0.153179001 | 2.880715132 | 3.506381642 |
| Sample 7 | 31 | 54 | 1.741935484 | 0.664516129 | 19.93548387 | 0.153179001 | 1.429102229 | 2.054768739 |
| Sample 8 | 31 | 47 | 1.516129032 | 0.458064516 | 13.74193548 | 0.153179001 | 1.203295777 | 1.828962287 |
| Sample 9 | 31 | 63 | 2.032258065 | 0.498924731 | 14.96774194 | 0.153179001 | 1.71942481 | 2.345091319 |
| Sample 10 | 31 | 65 | 2.096774194 | 0.956989247 | 28.70967742 | 0.153179001 | 1.783940939 | 2.409607449 |
| Sample 11 | 31 | 56 | 1.806451613 | 0.761290323 | 22.83870968 | 0.153179001 | 1.493618358 | 2.119284868 |
| Sample 12 | 31 | 51 | 1.64516129 | 0.569892473 | 17.09677419 | 0.153179001 | 1.332328035 | 1.957994545 |
| Sample 13 | 31 | 70 | 2.258064516 | 0.931182796 | 27.93548387 | 0.153179001 | 1.945231261 | 2.570897771 |

ANOVA

| *Sources* | *SS* | *df* | *MS* | *F* | *P value* | *F crit* | *RMSSE* | *Omega Sq* |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 111.2258065 | 12 | 9.268817204 | 12.74277917 | 4.641E-22 | 1.777036001 | 0.641137577 | 0.259073159 |
| Within Groups | 283.6774194 | 390 | 0.727377998 | | | | | |
| Total | 394.9032258 | 402 | 0.982346333 | | | | | |

Table 26. ANOVA Results: *Smooth*

ANOVA: Single Factor

| DESCRIPTION | | | | | Alpha | 0.05 | | |
|---|---|---|---|---|---|---|---|---|
| *Groups* | *Count* | *Sum* | *Mean* | *Variance* | *SS* | *Std Err* | *Lower* | *Upper* |
| Sample 1 | 31 | 80 | 2.580645161 | 1.451612903 | 43.5483871 | 0.185864115 | 2.201059999 | 2.960230324 |
| Sample 2 | 31 | 65 | 2.096774194 | 0.823655914 | 24.70967742 | 0.185864115 | 1.717189031 | 2.476359356 |
| Sample 3 | 31 | 81 | 2.612903226 | 0.778494624 | 23.35483871 | 0.185864115 | 2.233318064 | 2.992488388 |
| Sample 4 | 30 | 72 | 2.4 | 0.731034483 | 21.2 | 0.188936457 | 2.013581558 | 2.786418442 |
| Sample 5 | 31 | 91 | 2.935483871 | 1.462365591 | 43.87096774 | 0.185864115 | 2.555898709 | 3.315069033 |
| Sample 6 | 31 | 94 | 3.032258065 | 1.365591398 | 40.96774194 | 0.185864115 | 2.652672902 | 3.411843227 |
| Sample 7 | 31 | 54 | 1.741935484 | 0.797849462 | 23.93548387 | 0.185864115 | 1.362350322 | 2.121520646 |
| Sample 8 | 31 | 66 | 2.129032258 | 1.116129032 | 33.48387097 | 0.185864115 | 1.749447096 | 2.50861742 |
| Sample 9 | 31 | 75 | 2.419354839 | 1.184946237 | 35.5483871 | 0.185864115 | 2.039769676 | 2.798940001 |
| Sample 10 | 30 | 63 | 2.1 | 0.920689655 | 26.7 | 0.188936457 | 1.713581558 | 2.486418442 |
| Sample 11 | 31 | 62 | 2 | 0.733333333 | 22 | 0.185864115 | 1.620414838 | 2.379585162 |
| Sample 12 | 31 | 68 | 2.193548387 | 0.961290323 | 28.83870968 | 0.185864115 | 1.813963225 | 2.573133549 |
| Sample 13 | 31 | 74 | 2.387096774 | 1.578494624 | 47.35483871 | 0.185864115 | 2.007511612 | 2.766681936 |

ANOVA

| *Sources* | *SS* | *df* | *MS* | *F* | *P value* | *F crit* | *RMSSE* | *Omega Sq* |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 50.49208431 | 12 | 4.207673692 | 3.929065451 | 1.00516E-05 | 1.77716466 | 0.356249603 | 0.080588979 |
| Within Groups | 415.5129032 | 388 | 1.070909544 | | | | | |
| Total | 466.0049875 | 400 | 1.165012469 | | | | | |

Table 27. ANOVA Results: *Low Frequency*

ANOVA: Single Factor

DESCRIPTION Alpha 0.05

| *Groups* | *Count* | *Sum* | *Mean* | *Variance* | *SS* | *Std Err* | *Lower* | *Upper* |
|---|---|---|---|---|---|---|---|---|
| Sample 1 | 30 | 50 | 1.666666667 | 1.057471264 | 30.66666667 | 0.182469334 | 1.293474976 | 2.039858358 |
| Sample 2 | 31 | 47 | 1.516129032 | 0.524731183 | 15.74193548 | 0.179502155 | 1.149536725 | 1.88272134 |
| Sample 3 | 30 | 86 | 2.866666667 | 0.947126437 | 27.46666667 | 0.182469334 | 2.493474976 | 3.239858358 |
| Sample 4 | 31 | 100 | 3.225806452 | 1.180645161 | 35.41935484 | 0.179502155 | 2.859214144 | 3.592398759 |
| Sample 5 | 31 | 81 | 2.612903226 | 1.44516129 | 43.35483871 | 0.179502155 | 2.246310918 | 2.979495533 |
| Sample 6 | 30 | 86 | 2.866666667 | 1.154022989 | 33.46666667 | 0.182469334 | 2.493474976 | 3.239858358 |
| Sample 7 | 30 | 49 | 1.633333333 | 0.516091954 | 14.96666667 | 0.182469334 | 1.260141642 | 2.006525024 |
| Sample 8 | 30 | 67 | 2.233333333 | 0.874712644 | 25.36666667 | 0.182469334 | 1.860141642 | 2.606525024 |
| Sample 9 | 30 | 64 | 2.133333333 | 1.085057471 | 31.46666667 | 0.182469334 | 1.760141642 | 2.506525024 |
| Sample 10 | 31 | 65 | 2.096774194 | 1.156989247 | 34.70967742 | 0.179502155 | 1.730181886 | 2.463366501 |
| Sample 11 | 31 | 62 | 2 | 0.866666667 | 26 | 0.179502155 | 1.633407692 | 2.366592308 |
| Sample 12 | 30 | 61 | 2.033333333 | 0.86091954 | 24.96666667 | 0.182469334 | 1.660141642 | 2.406525024 |
| Sample 13 | 31 | 61 | 1.967741935 | 1.298924731 | 38.96774194 | 0.179502155 | 1.601149628 | 2.334334243 |

ANOVA

| *Sources* | *SS* | *df* | *MS* | *F* | *P value* | *F crit* | *RMSSE* | *Omega Sq* |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 101.3261486 | 12 | 8.443845715 | 8.453552622 | 2.96983E-14 | 1.777492209 | 0.52612793 | 0.184249643 |
| Within Groups | 382.5602151 | 383 | 0.998851736 | | | | | |
| Total | 483.8863636 | 395 | 1.225028769 | | | | | |

Table 28. ANOVA Results: *Solid*

ANOVA: Single Factor

DESCRIPTION Alpha 0.05

| *Groups* | *Count* | *Sum* | *Mean* | *Variance* | *SS* | *Std Err* | *Lower* | *Upper* |
|---|---|---|---|---|---|---|---|---|
| Sample 1 | 26 | 61 | 2.346153846 | 1.275384615 | 31.88461538 | 0.221122246 | 1.890744056 | 2.801563636 |
| Sample 2 | 30 | 90 | 3 | 1.172413793 | 34 | 0.205853608 | 2.578982099 | 3.421017901 |
| Sample 3 | 30 | 81 | 2.7 | 0.975862069 | 28.3 | 0.205853608 | 2.278982099 | 3.121017901 |
| Sample 4 | 29 | 64 | 2.206896552 | 1.24137931 | 34.75862069 | 0.209372728 | 1.778015961 | 2.635777143 |
| Sample 5 | 28 | 68 | 2.428571429 | 1.439153439 | 38.85714286 | 0.213078728 | 1.991369993 | 2.865772864 |
| Sample 6 | 29 | 72 | 2.482758621 | 0.901477833 | 25.24137931 | 0.209372728 | 2.05387803 | 2.911639212 |
| Sample 7 | 30 | 67 | 2.233333333 | 1.426436782 | 41.36666667 | 0.205853608 | 1.812315433 | 2.654351234 |
| Sample 8 | 30 | 73 | 2.433333333 | 1.426436782 | 41.36666667 | 0.205853608 | 2.012315433 | 2.854351234 |
| Sample 9 | 29 | 83 | 2.862068966 | 1.480295567 | 41.44827586 | 0.209372728 | 2.433188375 | 3.290949556 |
| Sample 10 | 30 | 69 | 2.3 | 1.251724138 | 36.3 | 0.205853608 | 1.878982099 | 2.721017901 |
| Sample 11 | 28 | 66 | 2.357142857 | 1.052910053 | 28.42857143 | 0.213078728 | 1.919941422 | 2.794344293 |
| Sample 12 | 30 | 88 | 2.933333333 | 1.305747126 | 37.86666667 | 0.205853608 | 2.512315433 | 3.354351234 |
| Sample 13 | 30 | 64 | 2.133333333 | 1.567816092 | 45.46666667 | 0.205853608 | 1.712315433 | 2.554351234 |

ANOVA

| *Sources* | *SS* | *df* | *MS* | *F* | *P value* | *F crit* | *RMSSE* | *Omega Sq* |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 29.45879113 | 12 | 2.45489926 | 1.931058606 | 0.029715102 | 1.778673095 | 0.255225298 | 0.028635277 |
| Within Groups | 465.2852722 | 366 | 1.271271236 | | | | | |
| Total | 494.7440633 | 378 | 1.308846728 | | | | | |

Table 29. ANOVA Results: *Reliable*

ANOVA: Single Factor

DESCRIPTION Alpha 0.05

| Groups | Count | Sum | Mean | Variance | SS | Std Err | Lower | Upper |
|---|---|---|---|---|---|---|---|---|
| Sample 1 | 26 | 59 | 2.269230769 | 1.404615385 | 35.11538462 | 0.207091005 | 1.842718861 | 2.695742677 |
| Sample 2 | 25 | 58 | 2.32 | 1.06 | 25.44 | 0.211192215 | 1.884120692 | 2.755879308 |
| Sample 3 | 28 | 70 | 2.5 | 0.777777778 | 21 | 0.199557885 | 2.090541041 | 2.909458959 |
| Sample 4 | 26 | 64 | 2.461538462 | 0.818461538 | 20.46153846 | 0.207091005 | 2.035026554 | 2.88805037 |
| Sample 5 | 25 | 70 | 2.8 | 1 | 24 | 0.211192215 | 2.364120692 | 3.235879308 |
| Sample 6 | 26 | 62 | 2.384615385 | 0.646153846 | 16.15384615 | 0.207091005 | 1.958103477 | 2.811127293 |
| Sample 7 | 25 | 45 | 1.8 | 1 | 24 | 0.211192215 | 1.364120692 | 2.235879308 |
| Sample 8 | 27 | 69 | 2.555555556 | 1.333333333 | 34.66666667 | 0.203219803 | 2.137831267 | 2.973279844 |
| Sample 9 | 26 | 61 | 2.346153846 | 0.795384615 | 19.88461538 | 0.207091005 | 1.919641938 | 2.772665754 |
| Sample 10 | 24 | 49 | 2.041666667 | 0.911231884 | 20.95833333 | 0.215547152 | 1.595773411 | 2.487559922 |
| Sample 11 | 24 | 56 | 2.333333333 | 1.536231884 | 35.33333333 | 0.215547152 | 1.887440078 | 2.779226589 |
| Sample 12 | 24 | 69 | 2.875 | 1.940217391 | 44.625 | 0.215547152 | 2.429106744 | 3.320893256 |
| Sample 13 | 24 | 50 | 2.083333333 | 1.384057971 | 31.83333333 | 0.215547152 | 1.637440078 | 2.529226589 |

ANOVA

| Sources | SS | df | MS | F | P value | F crit | RMSSE | Omega Sq |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 25.42491841 | 12 | 2.118743201 | 1.900126452 | 0.033649033 | 1.782788876 | 0.27733608 | 0.031694452 |
| Within Groups | 353.4720513 | 317 | 1.11505379 | | | | | |
| Total | 378.8969697 | 329 | 1.151662522 | | | | | |

Table 30. ANOVA Results: *Powerful*

ANOVA: Single Factor

DESCRIPTION Alpha 0.05

| Groups | Count | Sum | Mean | Variance | SS | Std Err | Lower | Upper |
|---|---|---|---|---|---|---|---|---|
| Sample 1 | 30 | 58 | 1.933333333 | 0.96091954 | 27.86666667 | 0.181638564 | 1.561840757 | 2.304825909 |
| Sample 2 | 31 | 63 | 2.032258065 | 1.032258065 | 30.96774194 | 0.178684895 | 1.667334825 | 2.397181304 |
| Sample 3 | 30 | 79 | 2.633333333 | 0.998850575 | 28.96666667 | 0.181638564 | 2.261840757 | 3.004825909 |
| Sample 4 | 31 | 69 | 2.225806452 | 0.913978495 | 27.41935484 | 0.178684895 | 1.860883212 | 2.590729691 |
| Sample 5 | 31 | 84 | 2.709677419 | 1.146236559 | 34.38709677 | 0.178684895 | 2.34475418 | 3.074600659 |
| Sample 6 | 31 | 79 | 2.548387097 | 1.455913978 | 43.67741935 | 0.178684895 | 2.183463857 | 2.913310336 |
| Sample 7 | 29 | 48 | 1.655172414 | 0.733990148 | 20.55172414 | 0.184743722 | 1.276742054 | 2.033602774 |
| Sample 8 | 31 | 66 | 2.129032258 | 1.316129032 | 39.48387097 | 0.178684895 | 1.764109019 | 2.493955497 |
| Sample 9 | 31 | 65 | 2.096774194 | 0.890322581 | 26.70967742 | 0.178684895 | 1.731850954 | 2.461697433 |
| Sample 10 | 29 | 52 | 1.793103448 | 0.527093596 | 14.75862069 | 0.184743722 | 1.414673089 | 2.171533808 |
| Sample 11 | 30 | 55 | 1.833333333 | 0.695402299 | 20.16666667 | 0.181638564 | 1.461840757 | 2.204825909 |
| Sample 12 | 29 | 66 | 2.275862069 | 1.13546798 | 31.79310345 | 0.184743722 | 1.897431709 | 2.654292429 |
| Sample 13 | 30 | 67 | 2.233333333 | 1.012643678 | 29.36666667 | 0.181638564 | 1.861840757 | 2.604825909 |

ANOVA

| Sources | SS | df | MS | F | P value | F crit | RMSSE | Omega Sq |
|---|---|---|---|---|---|---|---|---|
| Between Groups | 38.13408763 | 12 | 3.177840636 | 3.210663108 | 0.000209264 | 1.777692893 | 0.326862801 | 0.063232871 |
| Within Groups | 376.1152762 | 380 | 0.989777043 | | | | | |
| Total | 414.2493639 | 392 | 1.056758581 | | | | | |

Table 31. ANOVA Results: Command Functions

ANOVA: Single Factor

| DESCRIPTION | | | | | Alpha | 0.05 | | |
|---|---|---|---|---|---|---|---|---|
| *Groups* | *Count* | *Sum* | *Mean* | *Variance* | *SS* | *Std Err* | *Lower* | *Upper* |
| Sample 1 | 31 | 61 | 1.967741935 | 1.165591398 | 34.96774194 | 0.162279843 | 1.6363223 | 2.299161589 |
| Sample 2 | 31 | 37 | 1.193548387 | 0.694623656 | 20.83870968 | 0.162279843 | 0.8621287 | 1.524968041 |
| Sample 3 | 31 | 43 | 1.387096774 | 0.84516129 | 25.35483871 | 0.162279843 | 1.0556771 | 1.718516428 |
| Sample 4 | 31 | 46 | 1.483870968 | 0.924731183 | 27.74193548 | 0.162279843 | 1.1524513 | 1.815290621 |
| Sample 5 | 31 | 36 | 1.161290323 | 0.606451613 | 18.19354839 | 0.162279843 | 0.8298707 | 1.492709976 |
| Sample 6 | 31 | 44 | 1.419354839 | 1.11827957 | 33.5483871 | 0.162279843 | 1.0879352 | 1.750774492 |
| Sample 7 | 31 | 16 | 0.516129032 | 0.458064516 | 13.74193548 | 0.162279843 | 0.1847094 | 0.847548686 |
| Sample 8 | 31 | 34 | 1.096774194 | 1.156989247 | 34.70967742 | 0.162279843 | 0.7653545 | 1.428193847 |
| Sample 9 | 31 | 24 | 0.774193548 | 0.513978495 | 15.41935484 | 0.162279843 | 0.4427739 | 1.105613202 |
| Sample 10 | 31 | 20 | 0.64516129 | 0.63655914 | 19.09677419 | 0.162279843 | 0.3137416 | 0.976580944 |
| Sample 11 | 31 | 35 | 1.129032258 | 0.716129032 | 21.48387097 | 0.162279843 | 0.7976126 | 1.460451912 |
| Sample 12 | 31 | 42 | 1.35483871 | 1.23655914 | 37.09677419 | 0.162279843 | 1.0234191 | 1.686258363 |
| Sample 13 | 31 | 26 | 0.838709677 | 0.539784946 | 16.19354839 | 0.162279843 | 0.50729 | 1.170129331 |

ANOVA

| *Sources* | *SS* | *df* | *MS* | *F* | *P value* | *F crit* | *RMSSE* | *Omega Sq* |
|---|---|---|---|---|---|---|---|---|
| Between Gro | 57.37965261 | 12 | 4.781637717 | 5.857142857 | 2.26243E-09 | 1.777036001 | 0.4346724 | 0.126354909 |
| Within Group | 318.3870968 | 390 | 0.816377171 | | | | | |
| Total | 375.7667494 | 402 | 0.934743158 | | | | | |

# Appendix C. Kruskal-Wallis Test Results

Table 32. Kruskal-Wallis Results: *Dark*

Kruskal-Wallis Test

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 9 | Sample 10 | Sample 10 | Sample 11 | Sample 12 | Sample 13 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| median | 1 | 3 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | |
| rank sum | 8840.5 | 2861.5 | 5941.5 | 5049.5 | 10158 | 8684 | 4850.5 | 3754.5 | 5902 | 6047.5 | 4433 | 4879 | 10004.5 | |
| count | 27 | 28 | 28 | 27 | 28 | 29 | 29 | 29 | 29 | 29 | 29 | 29 | 29 | 370 |
| r^2/n | 2894608.898 | 292435.0804 | 1260765.08 | 944350.0093 | 3685177.286 | 2600408.828 | 811287.9397 | 486078.2845 | 1201158.759 | 1261112.284 | 677637.5517 | 820849.6897 | 3451380.009 | 20387249.7 |
| H | | | | | | | | | | | | | | 669.232071 |
| df | | | | | | | | | | | | | | 12 |
| p-value | | | | | | | | | | | | | | 1.6911E-135 |
| alpha | | | | | | | | | | | | | | 0.05 |
| sig | | | | | | | | | | | | | | yes |

Table 33. Kruskal-Wallis Results: *Dynamic*

Kruskal-Wallis Test

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 10 | Sample 11 | Sample 10 | Sample 11 | Sample 12 | Sample 13 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| median | 3 | 1 | 2 | 3 | 4 | 4 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | |
| rank sum | 8840.5 | 2861.5 | 5941.5 | 5049.5 | 10158 | 8684 | 4850.5 | 3754.5 | 5902 | 6047.5 | 4433 | 4879 | 10004.5 | |
| count | 31 | 30 | 31 | 31 | 31 | 31 | 29 | 30 | 31 | 30 | 31 | 31 | 30 | 397 |
| r^2/n | 2521110.976 | 272939.4083 | 1138755.556 | 822498.3952 | 3328547.226 | 2432640.516 | 811287.9397 | 469875.675 | 1123664.645 | 1219075.208 | 633919 | 767891.6452 | 3336334.008 | 18878540.2 |
| H | | | | | | | | | | | | | | 239.7587332 |
| df | | | | | | | | | | | | | | 12 |
| p-value | | | | | | | | | | | | | | 1.86181E-44 |
| alpha | | | | | | | | | | | | | | 0.05 |
| sig | | | | | | | | | | | | | | yes |

## Table 34. Kruskal-Wallis Results: *Sharp*

Kruskal-Wallis Test

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 10 | Sample 11 | Sample 10 | Sample 11 | Sample 12 | Sample 13 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| median | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | |
| rank sum | 8840.5 | 2861.5 | 5941.5 | 5049.5 | 10158 | 8684 | 4850.5 | 3754.5 | 5902 | 6047.5 | 4433 | 4879 | 10004.5 | |
| count | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 403 |
| r^2/n | 2521110.976 | 264134.9113 | 1138755.556 | 822498.3952 | 3328547.226 | 2432640.516 | 758946.7823 | 454718.3952 | 1123664.645 | 1179750.202 | 633919 | 767891.6452 | 3228710.331 | 18655288.58 |
| H | | | | | | | | | | | | | | 162.9813464 |
| df | | | | | | | | | | | | | | 12 |
| p-value | | | | | | | | | | | | | | 1.29583E-28 |
| alpha | | | | | | | | | | | | | | 0.05 |
| sig | | | | | | | | | | | | | | yes |

## Table 35. Kruskal-Wallis Results: *Expressive*

Kruskal-Wallis Test

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 10 | Sample 11 | Sample 10 | Sample 11 | Sample 12 | Sample 13 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| median | 1 | 4 | 3 | 2 | 1 | 1 | 3 | 3 | 3 | 2.5 | 3 | 2.5 | 1 | |
| rank sum | 8840.5 | 2861.5 | 5941.5 | 5049.5 | 10158 | 8684 | 4850.5 | 3754.5 | 5902 | 6047.5 | 4433 | 4879 | 10004.5 | |
| count | 29 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 29 | 30 | 29 | 387 |
| r^2/n | 2694980.698 | 272939.4083 | 1176714.075 | 849915.0083 | 3439498.8 | 2513728.533 | 784245.0083 | 469875.675 | 1161120.133 | 1219075.208 | 677637.5517 | 793488.0333 | 3451380.009 | 19504598.14 |
| H | | | | | | | | | | | | | | 394.7467547 |
| df | | | | | | | | | | | | | | 12 |
| p-value | | | | | | | | | | | | | | 4.8998E-77 |
| alpha | | | | | | | | | | | | | | 0.05 |
| sig | | | | | | | | | | | | | | yes |

Table 36. Kruskal-Wallis Results: *Strong*

Kruskal-Wallis Test

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 10 | Sample 11 | Sample 10 | Sample 11 | Sample 12 | Sample 13 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| median | 1 | 2 | 2 | 3 | 1 | 2 | 1 | 3 | 2 | 1 | 1 | 2 | 1 | |
| rank sum | 8840.5 | 2861.5 | 5941.5 | 5049.5 | 10158 | 8684 | 4850.5 | 3754.5 | 5902 | 6047.5 | 4433 | 4879 | 10004.5 | |
| count | 30 | 30 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 401 |
| r^2/n | 2605148.008 | 272939.4083 | 1138755.556 | 822498.3952 | 3328547.226 | 2432640.516 | 758946.7823 | 454718.3952 | 1123664.645 | 1179750.202 | 633919 | 767891.6452 | 3228710.331 | 18748130.11 |
| H | | | | | | | | | | | | | | 189.6251245 |
| df | | | | | | | | | | | | | | 12 |
| p-value | | | | | | | | | | | | | | 4.48581E-34 |
| alpha | | | | | | | | | | | | | | 0.05 |
| sig | | | | | | | | | | | | | | yes |

## Appendix D. Correlation Coefficients

Table 37. Correlation Coefficients. Pearson Correlation Method Used for Parametric Data Sets. Spearman Correlation Method Used for Non-Parametric Data Sets.

| Pearson/Spearman Correlation Coefficient | | Command Fuction | | | | | | | | | | | | | | | Psychoacoustic Metric | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Set + | Set - | On | Off | Answer Call | End Call | Media Mode | Next Track | Previous Track | Play/Pause | Mute On | Mute Off | Volume Up | Volume Down | Activate | Long | Dark | Dynamic | Sharp | High Freq | Intense | Expressive | Tonal | Short | Loud | Strong | Smooth | Low Freq | Solid | Reliable | Powerful |
| Command Fuction | Set + | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Set - | -0.259 | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | On | 0.664 | 0.002 | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Off | -0.477 | 0.767 | 0.129 | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Answer Call | 0.462 | 0.032 | 0.439 | 0.002 | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | End Call | 0.100 | 0.262 | 0.450 | 0.298 | 0.054 | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Media Mode | 0.186 | -0.669 | 0.426 | -0.145 | 0.031 | 0.002 | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | |
| | Next Track | 0.138 | 0.002 | 0.272 | -0.061 | 0.494 | -0.096 | -0.124 | 1.000 | | | | | | | | | | | | | | | | | | | | | | | |
| | Previous Track | -0.252 | 0.545 | 0.152 | 0.471 | -0.185 | 0.200 | -0.115 | 0.002 | 1.000 | | | | | | | | | | | | | | | | | | | | | | |
| | Play/Pause | -0.115 | -0.181 | -0.044 | 0.015 | -0.069 | -0.467 | 0.201 | 0.271 | -0.280 | 1.000 | | | | | | | | | | | | | | | | | | | | | |
| | Mute On | 0.015 | 0.100 | -0.117 | 0.164 | -0.012 | 0.244 | -0.122 | -0.295 | -0.384 | -0.011 | 1.000 | | | | | | | | | | | | | | | | | | | | |
| | Mute Off | 0.402 | -0.300 | 0.446 | -0.160 | 0.264 | 0.282 | 0.395 | -0.026 | -0.461 | 0.397 | 0.245 | 1.000 | | | | | | | | | | | | | | | | | | | |
| | Volume Up | 0.914 | -0.434 | 0.583 | -0.538 | 0.476 | -0.057 | 0.285 | 0.162 | -0.315 | -0.033 | 0.060 | 0.331 | 1.000 | | | | | | | | | | | | | | | | | | |
| | Volume Down | -0.283 | 0.554 | 0.043 | 0.655 | -0.083 | 0.419 | -0.192 | 0.002 | 0.481 | -0.210 | 0.365 | -0.424 | -0.276 | 1.000 | | | | | | | | | | | | | | | | | |
| | Activate | 0.729 | -0.288 | 0.528 | -0.368 | 0.418 | 0.023 | 0.191 | 0.227 | -0.390 | -0.265 | -0.142 | 0.156 | 0.616 | -0.391 | 1.000 | | | | | | | | | | | | | | | | |
| Psychoacoustic Metric | Long | -0.189 | 0.110 | 0.244 | 0.215 | 0.273 | 0.392 | 0.112 | 0.430 | 0.319 | -0.316 | -0.432 | 0.011 | -0.310 | -0.008 | 0.100 | 1.000 | | | | | | | | | | | | | | | |
| | Dark | -0.629 | 0.045 | -0.298 | 0.177 | -0.800 | -0.095 | 0.072 | -0.334 | 0.009 | 0.346 | -0.017 | 0.039 | -0.429 | 0.084 | -0.438 | -0.286 | 1.000 | | | | | | | | | | | | | | |
| | Dynamic | 0.011 | -0.110 | 0.278 | 0.065 | 0.123 | 0.162 | 0.147 | 0.209 | 0.261 | -0.340 | -0.309 | 0.115 | -0.267 | -0.150 | 0.160 | 0.924 | -0.357 | 1.000 | | | | | | | | | | | | | |
| | Sharp | 0.513 | 0.014 | 0.011 | -0.300 | 0.336 | 0.050 | -0.181 | -0.249 | -0.074 | -0.292 | 0.092 | 0.177 | 0.165 | -0.252 | 0.207 | -0.033 | -0.709 | 0.237 | 1.000 | | | | | | | | | | | | |
| | High Freq | 0.604 | 0.333 | 0.335 | -0.091 | 0.422 | 0.076 | -0.301 | 0.035 | 0.057 | -0.379 | -0.191 | 0.100 | 0.323 | -0.265 | 0.575 | 0.166 | -0.824 | 0.412 | 0.877 | 1.000 | | | | | | | | | | | |
| | Intense | 0.455 | 0.361 | 0.205 | -0.079 | 0.276 | 0.223 | -0.390 | -0.085 | 0.070 | -0.552 | -0.054 | 0.089 | 0.177 | -0.248 | 0.463 | 0.257 | -0.710 | 0.561 | 0.888 | 0.931 | 1.000 | | | | | | | | | | |
| | Expressive | 0.322 | -0.129 | 0.366 | -0.024 | 0.482 | 0.095 | 0.150 | 0.442 | 0.257 | -0.346 | -0.096 | 0.046 | 0.125 | -0.052 | 0.089 | 0.840 | -0.600 | 0.872 | 0.239 | 0.440 | 0.521 | 1.000 | | | | | | | | | |
| | Tonal | 0.356 | 0.141 | 0.567 | 0.186 | 0.613 | 0.610 | 0.161 | 0.039 | 0.039 | -0.583 | -0.011 | 0.290 | 0.198 | 0.002 | 0.488 | 0.607 | -0.478 | 0.582 | 0.287 | 0.545 | 0.586 | 0.547 | 1.000 | | | | | | | | |
| | Short | 0.278 | -0.026 | -0.098 | -0.109 | -0.085 | -0.336 | -0.099 | -0.416 | -0.339 | 0.306 | 0.439 | 0.087 | 0.384 | -0.001 | -0.010 | -0.966 | 0.237 | -0.897 | 0.061 | -0.046 | -0.169 | -0.818 | -0.437 | 1.000 | | | | | | | |
| | Loud | 0.414 | 0.236 | 0.263 | -0.075 | 0.380 | 0.351 | -0.219 | -0.006 | 0.031 | -0.660 | -0.119 | 0.081 | 0.135 | -0.208 | 0.560 | 0.483 | -0.692 | 0.632 | 0.778 | 0.864 | 0.933 | 0.567 | 0.774 | -0.390 | 1.000 | | | | | | |
| | Strong | 0.660 | 0.067 | 0.424 | -0.138 | 0.719 | 0.388 | -0.147 | 0.172 | 0.091 | -0.590 | 0.287 | 0.073 | 0.384 | 0.092 | 0.219 | 0.327 | -0.835 | 0.434 | 0.690 | 0.775 | 0.792 | 0.644 | 0.566 | -0.327 | 0.780 | 1.000 | | | | | |
| | Smooth | -0.240 | 0.053 | 0.136 | 0.334 | 0.336 | 0.135 | 0.155 | 0.536 | 0.142 | 0.009 | 0.255 | -0.061 | -0.110 | 0.418 | -0.153 | 0.484 | 0.000 | 0.226 | -0.551 | -0.327 | -0.256 | 0.489 | 0.199 | -0.482 | -0.112 | 0.077 | 1.000 | | | | |
| | Low Freq | -0.645 | -0.232 | -0.336 | 0.201 | -0.431 | 0.060 | 0.241 | -0.115 | -0.046 | 0.204 | 0.308 | -0.164 | -0.385 | 0.384 | -0.539 | -0.123 | 0.851 | -0.339 | -0.846 | -0.969 | -0.852 | -0.432 | -0.422 | 0.014 | -0.760 | -0.725 | 0.373 | 1.000 | | | |
| | Solid | -0.166 | 0.147 | -0.051 | 0.217 | -0.164 | 0.366 | -0.099 | 0.055 | 0.283 | -0.362 | 0.387 | -0.498 | -0.066 | 0.844 | -0.233 | -0.045 | 0.104 | -0.176 | -0.397 | -0.403 | -0.322 | 0.030 | -0.083 | -0.048 | -0.235 | 0.154 | 0.457 | 0.517 | 1.000 | | |
| | Reliable | 0.043 | 0.595 | 0.359 | 0.512 | 0.301 | 0.255 | -0.237 | 0.490 | 0.607 | -0.217 | -0.004 | -0.308 | -0.018 | 0.642 | -0.070 | 0.412 | -0.379 | 0.368 | -0.099 | 0.225 | 0.205 | 0.633 | 0.266 | -0.371 | 0.209 | 0.473 | 0.618 | -0.171 | 0.479 | 1.000 | |
| | Powerful | 0.217 | 0.508 | 0.235 | 0.189 | 0.237 | 0.421 | -0.359 | 0.124 | 0.448 | -0.632 | 0.118 | -0.133 | 0.015 | 0.275 | 0.137 | 0.457 | -0.566 | 0.434 | 0.389 | 0.605 | 0.742 | 0.641 | 0.546 | -0.428 | 0.727 | 0.769 | 0.316 | -0.492 | 0.229 | 0.693 | 1.000 |

Table 38. Significant Correlation Coefficients. Pearson Correlation Method Used for Parametric Data Sets. Spearman Correlation Method Used for Non-Parametric Data Sets.

| **Pearson/Spearman Correlation Coefficient** | | Command Fuction | | | | | | | | | | | | | | | Psychoacoustic Metric | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Set + | Set - | On | Off | Answer Call | End Call | Media Mode | Next Track | Previous Track | Play/Pause | Mute On | Mute Off | Volume Up | Volume Down | Activate | Long | Dark | Dynamic | Sharp | High Freq | Intense | Expressive | Tonal | Short | Loud | Strong | Smooth | Low Freq | Solid | Reliable | Powerful |
| Command Fuction | Set + | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Set - | | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | On | 0.664 | | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Off | -0.477 | 0.767 | | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Answer Call | 0.462 | | 0.439 | | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | End Call | | | 0.450 | | | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Media Mode | | -0.669 | 0.426 | | | | 1.000 | | | | | | | | | | | | | | | | | | | | | | | | |
| | Next Track | | | | | 0.494 | | | 1.000 | | | | | | | | | | | | | | | | | | | | | | | |
| | Previous Track | | 0.545 | | 0.471 | | | | | 1.000 | | | | | | | | | | | | | | | | | | | | | | |
| | Play/Pause | | | | | | -0.467 | | | | 1.000 | | | | | | | | | | | | | | | | | | | | | |
| | Mute On | | | | | | | | | -0.384 | | 1.000 | | | | | | | | | | | | | | | | | | | | |
| | Mute Off | 0.402 | | 0.446 | | | | 0.395 | | -0.461 | 0.397 | | 1.000 | | | | | | | | | | | | | | | | | | | |
| | Volume Up | 0.914 | -0.434 | 0.583 | -0.538 | 0.476 | | | | | | | | 1.000 | | | | | | | | | | | | | | | | | | |
| | Volume Down | | 0.554 | | 0.655 | | 0.419 | | | 0.481 | | 0.365 | -0.424 | | 1.000 | | | | | | | | | | | | | | | | | |
| | Activate | 0.729 | | 0.528 | -0.368 | 0.418 | | | | -0.390 | | | | 0.616 | -0.391 | 1.000 | | | | | | | | | | | | | | | | |
| Psychoacoustic Metric | Long | | | | | | 0.392 | | 0.430 | | | -0.432 | | | | | 1.000 | | | | | | | | | | | | | | | |
| | Dark | -0.629 | | | | -0.800 | | | | | | | | -0.429 | | -0.438 | | 1.000 | | | | | | | | | | | | | | |
| | Dynamic | | | | | | | | | | | | | | | | 0.924 | | 1.000 | | | | | | | | | | | | | |
| | Sharp | 0.513 | | | | | | | | | | | | | | | | -0.709 | | 1.000 | | | | | | | | | | | | |
| | High Freq | 0.604 | | | | 0.422 | | | | | | | | | | 0.575 | | -0.824 | 0.412 | 0.877 | 1.000 | | | | | | | | | | | |
| | Intense | 0.455 | 0.361 | | | | | -0.390 | | | -0.552 | | | | | 0.463 | | -0.710 | 0.561 | 0.888 | 0.931 | 1.000 | | | | | | | | | | |
| | Expressive | | | | | 0.482 | | | 0.442 | | | | | | | | 0.840 | -0.600 | 0.872 | | 0.440 | 0.521 | 1.000 | | | | | | | | | |
| | Tonal | 0.356 | | 0.567 | | 0.613 | 0.610 | | | | -0.583 | | | | | 0.488 | 0.607 | -0.478 | 0.582 | | 0.545 | 0.586 | 0.547 | 1.000 | | | | | | | | |
| | Short | | | | | | | | | | | | | | | | -0.966 | | -0.897 | | | | -0.818 | | 1.000 | | | | | | | |
| | Loud | 0.414 | | | | 0.380 | | | | | -0.660 | | | | | 0.560 | 0.483 | -0.692 | 0.632 | 0.778 | 0.864 | 0.933 | 0.567 | 0.774 | -0.390 | 1.000 | | | | | | |
| | Strong | 0.660 | | 0.424 | | 0.719 | 0.388 | | | | -0.590 | | | 0.384 | | | | -0.835 | 0.434 | 0.690 | 0.775 | 0.792 | 0.644 | 0.566 | | 0.780 | 1.000 | | | | | |
| | Smooth | | | | | | | | 0.536 | | | | | | 0.418 | | 0.484 | | | -0.551 | | | 0.489 | | -0.482 | | | 1.000 | | | | |
| | Low Freq | -0.645 | | | | -0.431 | | | | | | | | -0.385 | 0.384 | -0.539 | | 0.851 | | -0.846 | -0.969 | -0.852 | -0.432 | -0.422 | | -0.760 | -0.725 | 0.373 | 1.000 | | | |
| | Solid | | | | | | | | | | | | -0.498 | | 0.844 | | | | | -0.397 | -0.403 | | | | | | | 0.457 | 0.517 | 1.000 | | |
| | Reliable | | 0.595 | | 0.512 | | | | 0.490 | 0.607 | | | | | 0.642 | | 0.412 | | | | | | 0.633 | | | | 0.473 | 0.618 | | 0.479 | 1.000 | |
| | Powerful | | 0.508 | | | | 0.421 | | | 0.448 | -0.632 | | | | | | 0.457 | -0.566 | 0.434 | 0.389 | 0.605 | 0.742 | 0.641 | 0.546 | -0.428 | 0.727 | 0.769 | | -0.492 | | 0.693 | 1.000 |

# Appendix E. Post Hoc T-test Results

Table 39. *Long* T-test results (top) and significant results only (bottom)

**Long**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 5E-12 | | | | | | | | | | | | |
| Sample 3 | 7E-05 | 7E-05 | | | | | | | | | | | |
| Sample 4 | 7E-08 | 9E-04 | 8E-02 | | | | | | | | | | |
| Sample 5 | 0.016 | 0.000 | 0.000 | 0.000 | | | | | | | | | |
| Sample 6 | 0.751 | 0.000 | 0.001 | 0.000 | 0.009 | | | | | | | | |
| Sample 7 | 8E-06 | 6E-03 | 1E-01 | 9E-01 | 3E-09 | 5E-05 | | | | | | | |
| Sample 8 | 7E-09 | 2E-01 | 7E-04 | 3E-02 | 3E-14 | 2E-07 | 7E-02 | | | | | | |
| Sample 9 | 2E-04 | 3E-05 | 1E+00 | 1E-01 | 4E-08 | 5E-04 | 5E-02 | 1E-04 | | | | | |
| Sample 10 | 4E-04 | 5E-05 | 8E-01 | 8E-02 | 4E-09 | 6E-04 | 4E-02 | 2E-03 | 8E-01 | | | | |
| Sample 11 | 3E-07 | 6E-02 | 5E-02 | 4E-01 | 4E-10 | 1E-06 | 5E-01 | 3E-01 | 3E-02 | 2E-02 | | | |
| Sample 12 | 6E-06 | 8E-03 | 2E-01 | 9E-01 | 1E-09 | 9E-05 | 1E+00 | 2E-02 | 7E-02 | 9E-02 | 5E-01 | | |
| Sample 13 | 0.037 | 0.000 | 0.000 | 0.000 | 1.000 | 0.022 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |

**Long: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 5E-12 | | | | | | | | | | | | |
| Sample 3 | 7E-05 | 7E-05 | | | | | | | | | | | |
| Sample 4 | 7E-08 | 0.001 | | | | | | | | | | | |
| Sample 5 | 0.016 | 6E-14 | 3E-11 | 3E-12 | | | | | | | | | |
| Sample 6 | | 3E-10 | 6E-04 | 2E-06 | 0.009 | | | | | | | | |
| Sample 7 | 8E-06 | 0.006 | | | 3E-09 | 5E-05 | | | | | | | |
| Sample 8 | 7E-09 | | 7E-04 | 0.032 | 3E-14 | 2E-07 | | | | | | | |
| Sample 9 | 2E-04 | 3E-05 | | | 4E-08 | 5E-04 | | 0.000 | | | | | |
| Sample 10 | 4E-04 | 5E-05 | | | 4E-09 | 6E-04 | 0.035 | 0.002 | | | | | |
| Sample 11 | 3E-07 | | | | 4E-10 | 1E-06 | | | 0.033 | 0.015 | | | |
| Sample 12 | 6E-06 | 0.008 | | | 1E-09 | 9E-05 | | 0.021 | | | | | |
| Sample 13 | 0.037 | 5E-11 | 3E-08 | 6E-10 | | 0.022 | 1E-10 | 1E-12 | 3E-08 | 4E-08 | 2E-10 | 6E-10 | |

Table 40. *High Freq* T-test results (top) and significant results only (bottom)

**High Freq**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 1.000 | | | | | | | | | | | | |
| Sample 3 | 4E-07 | 3E-07 | | | | | | | | | | | |
| Sample 4 | 1E-05 | 9E-08 | 7E-01 | | | | | | | | | | |
| Sample 5 | 7E-06 | 2E-06 | 5E-02 | 2E-01 | | | | | | | | | |
| Sample 6 | 5E-06 | 8E-05 | 9E-03 | 8E-02 | 7E-01 | | | | | | | | |
| Sample 7 | 0.723 | 0.199 | 0.000 | 0.000 | 0.001 | 0.001 | | | | | | | |
| Sample 8 | 0.056 | 0.004 | 0.000 | 0.000 | 0.008 | 0.012 | 0.187 | | | | | | |
| Sample 9 | 0.205 | 0.016 | 0.000 | 0.000 | 0.001 | 0.006 | 0.364 | 0.459 | | | | | |
| Sample 10 | 0.003 | 0.000 | 0.004 | 0.006 | 0.326 | 0.364 | 0.005 | 0.086 | 0.053 | | | | |
| Sample 11 | 0.062 | 0.004 | 0.000 | 0.000 | 0.013 | 0.029 | 0.164 | 0.832 | 0.408 | 0.143 | | | |
| Sample 12 | 1.000 | 0.746 | 0.000 | 0.000 | 0.000 | 0.000 | 0.363 | 0.006 | 0.039 | 0.001 | 0.002 | | |
| Sample 13 | 0.788 | 0.502 | 0.000 | 0.000 | 0.000 | 0.000 | 0.621 | 0.045 | 0.130 | 0.001 | 0.032 | 0.839 | |

**High Freq: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | | | | | | | | | | | | | |
| Sample 3 | 4E-07 | 3E-07 | | | | | | | | | | | |
| Sample 4 | 1E-05 | 9E-08 | | | | | | | | | | | |
| Sample 5 | 7E-06 | 2E-06 | 0.045 | | | | | | | | | | |
| Sample 6 | 5E-06 | 8E-05 | 0.009 | | | | | | | | | | |
| Sample 7 | | | 5E-06 | 6E-06 | 1E-03 | 8E-04 | | | | | | | |
| Sample 8 | | 0.004 | 0.000 | 0.000 | 0.008 | 0.012 | | | | | | | |
| Sample 9 | | 0.016 | 3E-04 | 1E-04 | 0.001 | 0.006 | | | | | | | |
| Sample 10 | 0.003 | 1E-04 | 0.004 | 0.006 | | | 0.005 | | | | | | |
| Sample 11 | | 0.004 | 5E-04 | 7E-05 | 0.013 | 0.029 | | | | | | | |
| Sample 12 | | | 1E-06 | 5E-08 | 8E-06 | 1E-05 | | 0.006 | 0.039 | 5E-04 | 0.002 | | |
| Sample 13 | | | 2E-06 | 6E-06 | 1E-05 | 1E-05 | | 0.045 | | 0.001 | 0.032 | | |

Table 41. *Intense* T-test results (top) and significant results only (bottom)

**Intense**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.001 | | | | | | | | | | | | |
| Sample 3 | 0.345 | 0.000 | | | | | | | | | | | |
| Sample 4 | 1.000 | 0.002 | 0.265 | | | | | | | | | | |
| Sample 5 | 2E-05 | 2E-09 | 2E-03 | 3E-06 | | | | | | | | | |
| Sample 6 | 0.000 | 0.000 | 0.039 | 0.000 | 0.325 | | | | | | | | |
| Sample 7 | 0.065 | 0.614 | 0.001 | 0.031 | 0.000 | 0.000 | | | | | | | |
| Sample 8 | 0.059 | 0.525 | 0.002 | 0.026 | 0.000 | 0.000 | 1.000 | | | | | | |
| Sample 9 | 0.091 | 0.086 | 0.002 | 0.097 | 0.000 | 0.000 | 0.395 | 0.296 | | | | | |
| Sample 10 | 0.177 | 0.094 | 0.006 | 0.102 | 0.000 | 0.000 | 0.293 | 0.229 | 0.856 | | | | |
| Sample 11 | 0.024 | 0.420 | 0.001 | 0.011 | 0.000 | 0.000 | 0.879 | 0.851 | 0.407 | 0.246 | | | |
| Sample 12 | 0.006 | 0.851 | 0.000 | 0.003 | 0.000 | 0.000 | 0.682 | 0.647 | 0.130 | 0.096 | 0.423 | | |
| Sample 13 | 0.090 | 0.000 | 0.423 | 0.080 | 0.026 | 0.108 | 0.001 | 0.000 | 0.000 | 0.001 | 0.001 | 0.000 | |

**Intense: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.001 | | | | | | | | | | | | |
| Sample 3 | | 1E-05 | | | | | | | | | | | |
| Sample 4 | | 0.002 | | | | | | | | | | | |
| Sample 5 | 2E-05 | 2E-09 | 0.002 | 3E-06 | | | | | | | | | |
| Sample 6 | 0.000 | 1E-06 | 0.039 | 0.000 | | | | | | | | | |
| Sample 7 | | | 0.001 | 0.031 | 8E-08 | 2E-05 | | | | | | | |
| Sample 8 | | | 2E-03 | 0.026 | 6E-07 | 6E-05 | | | | | | | |
| Sample 9 | | | 0.002 | | 1E-06 | 3E-05 | | | | | | | |
| Sample 10 | | | 0.006 | | 2E-06 | 3E-05 | | | | | | | |
| Sample 11 | 0.024 | | 0.001 | 0.011 | 6E-08 | 1E-05 | | | | | | | |
| Sample 12 | 0.006 | | 2E-04 | 0.003 | 1E-09 | 4E-07 | | | | | | | |
| Sample 13 | | 2E-05 | | | 0.026 | | 8E-04 | 4E-04 | 0.000 | 0.001 | 7E-04 | 9E-06 | |

Table 42. *Tonal* T-test results (top) and significant results only (bottom)

**Tonal**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.003 | | | | | | | | | | | | |
| Sample 3 | 6E-13 | 7E-10 | | | | | | | | | | | |
| Sample 4 | 3E-12 | 2E-08 | 4E-01 | | | | | | | | | | |
| Sample 5 | 2E-07 | 7E-04 | 2E-02 | 3E-02 | | | | | | | | | |
| Sample 6 | 5E-09 | 7E-05 | 5E-03 | 2E-02 | 9E-01 | | | | | | | | |
| Sample 7 | 2E-04 | 3E-01 | 6E-07 | 4E-06 | 4E-03 | 5E-04 | | | | | | | |
| Sample 8 | 3E-08 | 3E-04 | 9E-04 | 3E-03 | 6E-01 | 4E-01 | 2E-03 | | | | | | |
| Sample 9 | 2E-08 | 4E-06 | 7E-04 | 4E-03 | 7E-01 | 4E-01 | 2E-03 | 9E-01 | | | | | |
| Sample 10 | 3E-07 | 3E-03 | 7E-04 | 1E-02 | 5E-01 | 4E-01 | 9E-03 | 1E+00 | 9E-01 | | | | |
| Sample 11 | 3E-09 | 9E-05 | 3E-02 | 1E-01 | 5E-01 | 5E-01 | 2E-05 | 2E-01 | 3E-01 | 1E-01 | | | |
| Sample 12 | 2E-06 | 2E-02 | 5E-07 | 2E-05 | 5E-02 | 3E-02 | 3E-01 | 1E-01 | 1E-01 | 1E-01 | 2E-03 | | |
| Sample 13 | 0.023 | 0.363 | 0.000 | 0.000 | 0.000 | 0.000 | 0.003 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | |

**Tonal: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.003 | | | | | | | | | | | | |
| Sample 3 | 6E-13 | 7E-10 | | | | | | | | | | | |
| Sample 4 | 3E-12 | 2E-08 | | | | | | | | | | | |
| Sample 5 | 2E-07 | 7E-04 | 0.015 | 0.030 | | | | | | | | | |
| Sample 6 | 5E-09 | 7E-05 | 0.005 | 0.017 | | | | | | | | | |
| Sample 7 | 2E-04 | | 6E-07 | 4E-06 | 0.004 | 0.000 | | | | | | | |
| Sample 8 | 3E-08 | 0.000 | 0.001 | 0.003 | | | 0.002 | | | | | | |
| Sample 9 | 2E-08 | 0.000 | 0.001 | 0.004 | | | 0.002 | | | | | | |
| Sample 10 | 3E-07 | 0.003 | 0.001 | 0.012 | | | 0.009 | | | | | | |
| Sample 11 | 3E-09 | 0.000 | 0.033 | | | | 0.000 | | | | | | |
| Sample 12 | 2E-06 | 0.024 | 5E-07 | 2E-05 | 0.046 | 0.030 | | | | | 0.002 | | |
| Sample 13 | 0.023 | | 7E-10 | 2E-08 | 2E-05 | 5E-07 | 0.003 | 4E-05 | 2E-05 | 2E-05 | 3E-07 | 0.001 | |

Table 43. *Short* T-test results (top) and significant results only (bottom)

**Short**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.237 | | | | | | | | | | | | |
| Sample 3 | 0.332 | 0.023 | | | | | | | | | | | |
| Sample 4 | 0.173 | 0.019 | 0.427 | | | | | | | | | | |
| Sample 5 | 0.234 | 0.008 | 0.666 | 0.186 | | | | | | | | | |
| Sample 6 | 0.381 | 0.033 | 0.705 | 0.249 | 1.000 | | | | | | | | |
| Sample 7 | 0.104 | 0.260 | 0.002 | 0.000 | 0.000 | 0.005 | | | | | | | |
| Sample 8 | 0.037 | 0.172 | 0.000 | 0.000 | 0.000 | 0.000 | 0.266 | | | | | | |
| Sample 9 | 0.061 | 0.805 | 0.002 | 0.001 | 0.000 | 0.002 | 0.692 | 0.802 | | | | | |
| Sample 10 | 0.210 | 0.806 | 0.001 | 0.000 | 0.006 | 0.069 | 0.203 | 0.031 | 0.299 | | | | |
| Sample 11 | 0.028 | 0.163 | 0.001 | 0.001 | 0.000 | 0.000 | 0.426 | 0.802 | 0.491 | 0.148 | | | |
| Sample 12 | 0.066 | 0.608 | 0.002 | 0.002 | 0.001 | 0.016 | 0.857 | 0.647 | 0.802 | 0.299 | 0.377 | | |
| Sample 13 | 0.003 | 0.055 | 0.001 | 0.003 | 0.002 | 0.001 | 0.133 | 0.186 | 0.162 | 0.024 | 0.295 | 0.017 | |

**Short: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | | | | | | | | | | | | | |
| Sample 3 | | 0.023 | | | | | | | | | | | |
| Sample 4 | | 0.019 | | | | | | | | | | | |
| Sample 5 | | 0.008 | | | | | | | | | | | |
| Sample 6 | | 0.033 | | | | | | | | | | | |
| Sample 7 | | | 0.002 | 0.000 | 0.000 | 0.005 | | | | | | | |
| Sample 8 | 0.037 | | 3E-04 | 1E-04 | 4E-06 | 3E-04 | | | | | | | |
| Sample 9 | | | 0.002 | 0.001 | 0.000 | 0.002 | | | | | | | |
| Sample 10 | | | 0.001 | 0.000 | 0.006 | | | 0.031 | | | | | |
| Sample 11 | 0.028 | | 9E-04 | 8E-04 | 0.000 | 0.000 | | | | | | | |
| Sample 12 | | | 2E-03 | 2E-03 | 0.001 | 0.016 | | | | | | | |
| Sample 13 | 0.003 | | 1E-03 | 3E-03 | 2E-03 | 1E-03 | | | | 0.024 | | 0.017 | |

Table 44. *Loud* T-test results (top) and significant results only (bottom)

**Loud**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.001 | | | | | | | | | | | | |
| Sample 3 | 0.014 | 0.000 | | | | | | | | | | | |
| Sample 4 | 0.017 | 0.000 | 0.712 | | | | | | | | | | |
| Sample 5 | 2E-05 | 4E-08 | 4E-02 | 1E-02 | | | | | | | | | |
| Sample 6 | 1E-06 | 2E-09 | 1E-02 | 2E-03 | 3E-01 | | | | | | | | |
| Sample 7 | 0.032 | 0.745 | 0.000 | 0.000 | 0.000 | 0.000 | | | | | | | |
| Sample 8 | 1E-03 | 4E-01 | 4E-07 | 2E-06 | 8E-09 | 3E-10 | 9E-02 | | | | | | |
| Sample 9 | 0.378 | 0.054 | 0.001 | 0.006 | 0.000 | 0.000 | 0.095 | 0.002 | | | | | |
| Sample 10 | 0.693 | 0.062 | 0.012 | 0.027 | 0.000 | 0.000 | 0.032 | 0.000 | 0.730 | | | | |
| Sample 11 | 0.056 | 0.514 | 0.000 | 0.001 | 0.000 | 0.000 | 0.662 | 0.037 | 0.198 | 0.059 | | | |
| Sample 12 | 0.011 | 0.851 | 0.000 | 0.000 | 0.000 | 0.000 | 0.586 | 0.380 | 0.031 | 0.011 | 0.305 | | |
| Sample 13 | 0.751 | 0.004 | 0.062 | 0.110 | 0.000 | 0.000 | 0.009 | 0.001 | 0.256 | 0.432 | 0.037 | 0.005 | |

**Loud: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.001 | | | | | | | | | | | | |
| Sample 3 | 0.014 | 7E-07 | | | | | | | | | | | |
| Sample 4 | 0.017 | 1E-05 | | | | | | | | | | | |
| Sample 5 | 2E-05 | 4E-08 | 0.037 | 0.011 | | | | | | | | | |
| Sample 6 | 1E-06 | 2E-09 | 0.011 | 0.002 | | | | | | | | | |
| Sample 7 | 0.032 | | 5E-05 | 1E-04 | 1E-07 | 2E-08 | | | | | | | |
| Sample 8 | 1E-03 | | 4E-07 | 2E-06 | 8E-09 | 3E-10 | | | | | | | |
| Sample 9 | | | 0.001 | 0.006 | 4E-06 | 2E-06 | | 0.002 | | | | | |
| Sample 10 | | | 0.012 | 0.027 | 2E-05 | 3E-05 | 0.032 | 0.000 | | | | | |
| Sample 11 | | | 1E-04 | 0.001 | 3E-07 | 2E-08 | | 0.037 | | | | | |
| Sample 12 | 0.011 | | 1E-07 | 2E-05 | 3E-09 | 4E-09 | | | 0.031 | 0.011 | | | |
| Sample 13 | | 0.004 | | | 0.000 | 3E-05 | 0.009 | 1E-03 | | | 0.037 | 0.005 | |

Table 45. *Smooth* T-test results (top) and significant results only (bottom)

**Smooth**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.041 | | | | | | | | | | | | |
| Sample 3 | 0.884 | 0.004 | | | | | | | | | | | |
| Sample 4 | 0.396 | 0.153 | 0.297 | | | | | | | | | | |
| Sample 5 | 0.227 | 0.006 | 0.224 | 0.006 | | | | | | | | | |
| Sample 6 | 0.105 | 0.004 | 0.079 | 0.009 | 0.639 | | | | | | | | |
| Sample 7 | 0.007 | 0.086 | 0.000 | 0.008 | 0.000 | 0.000 | | | | | | | |
| Sample 8 | 0.114 | 0.861 | 0.066 | 0.371 | 0.009 | 0.013 | 0.110 | | | | | | |
| Sample 9 | 0.567 | 0.096 | 0.385 | 1.000 | 0.062 | 0.044 | 0.004 | 0.222 | | | | | |
| Sample 10 | 0.095 | 0.876 | 0.040 | 0.305 | 0.002 | 0.001 | 0.115 | 0.758 | 0.186 | | | | |
| Sample 11 | 0.029 | 0.598 | 0.005 | 0.025 | 0.001 | 0.001 | 0.234 | 0.525 | 0.030 | 0.791 | | | |
| Sample 12 | 0.148 | 0.610 | 0.040 | 0.363 | 0.012 | 0.010 | 0.065 | 0.758 | 0.335 | 0.807 | 0.363 | | |
| Sample 13 | 0.522 | 0.239 | 0.394 | 0.889 | 0.035 | 0.021 | 0.019 | 0.361 | 0.895 | 0.152 | 0.195 | 0.495 | |

**Smooth: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.041 | | | | | | | | | | | | |
| Sample 3 | | 0.004 | | | | | | | | | | | |
| Sample 4 | | | | | | | | | | | | | |
| Sample 5 | | 0.006 | | 0.006 | | | | | | | | | |
| Sample 6 | | 4E-03 | | 0.009 | | | | | | | | | |
| Sample 7 | 0.007 | | 1E-04 | 0.008 | 3E-05 | 2E-05 | | | | | | | |
| Sample 8 | | | | | 0.009 | 0.013 | | | | | | | |
| Sample 9 | | | | | | 0.044 | 0.004 | | | | | | |
| Sample 10 | | | 0.040 | | 0.002 | 0.001 | | | | | | | |
| Sample 11 | 0.029 | | 0.005 | 0.025 | 0.001 | 0.001 | | | 0.030 | | | | |
| Sample 12 | | | 0.040 | | 0.012 | 0.010 | | | | | | | |
| Sample 13 | | | | | 0.035 | 0.021 | 0.019 | | | | | | |

Table 46. *Low Freq* T-test results (top) and significant results only (bottom)

**Low Freq**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.442 | | | | | | | | | | | | |
| Sample 3 | 1E-06 | 8E-08 | | | | | | | | | | | |
| Sample 4 | 3E-07 | 4E-09 | 1E-01 | | | | | | | | | | |
| Sample 5 | 0.001 | 0.000 | 0.375 | 0.009 | | | | | | | | | |
| Sample 6 | 7E-05 | 1E-06 | 1E+00 | 1E-01 | 5E-01 | | | | | | | | |
| Sample 7 | 0.876 | 0.442 | 0.000 | 0.000 | 0.001 | 0.000 | | | | | | | |
| Sample 8 | 0.009 | 0.001 | 0.006 | 0.001 | 0.173 | 0.011 | 0.001 | | | | | | |
| Sample 9 | 0.041 | 0.005 | 0.002 | 0.000 | 0.074 | 0.016 | 0.014 | 0.655 | | | | | |
| Sample 10 | 0.105 | 0.017 | 0.005 | 0.000 | 0.065 | 0.016 | 0.070 | 0.326 | 0.573 | | | | |
| Sample 11 | 0.078 | 0.016 | 0.000 | 0.000 | 0.009 | 0.003 | 0.031 | 0.339 | 0.610 | 0.682 | | | |
| Sample 12 | 0.028 | 0.001 | 0.001 | 0.000 | 0.011 | 0.001 | 0.003 | 0.537 | 1.000 | 0.813 | 0.722 | | |
| Sample 13 | 0.224 | 0.050 | 0.001 | 0.000 | 0.033 | 0.000 | 0.102 | 0.409 | 0.564 | 0.614 | 0.889 | 0.662 | |

**Low Freq: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | | | | | | | | | | | | | |
| Sample 3 | 1E-06 | 8E-08 | | | | | | | | | | | |
| Sample 4 | 3E-07 | 4E-09 | | | | | | | | | | | |
| Sample 5 | 0.001 | 7E-05 | | 0.009 | | | | | | | | | |
| Sample 6 | 7E-05 | 1E-06 | | | | | | | | | | | |
| Sample 7 | | | 8E-07 | 1E-07 | 9E-04 | 1E-05 | | | | | | | |
| Sample 8 | 0.009 | 0.001 | 0.006 | 0.001 | | 0.011 | 0.001 | | | | | | |
| Sample 9 | 0.041 | 0.005 | 0.002 | 7E-05 | | 0.016 | 0.014 | | | | | | |
| Sample 10 | | 0.017 | 0.005 | 2E-04 | | 0.016 | | | | | | | |
| Sample 11 | | 0.016 | 0.000 | 2E-06 | 0.009 | 0.003 | 0.031 | | | | | | |
| Sample 12 | 0.028 | 0.001 | 0.001 | 2E-05 | 0.011 | 0.001 | 0.003 | | | | | | |
| Sample 13 | | | 0.001 | 3E-05 | 0.033 | 0.000 | | | | | | | |

Table 47. *Solid* T-test results (top) and significant results only (bottom)

**Solid**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.005 | | | | | | | | | | | | |
| Sample 3 | 0.409 | 0.147 | | | | | | | | | | | |
| Sample 4 | 0.495 | 0.012 | 0.070 | | | | | | | | | | |
| Sample 5 | 0.557 | 0.036 | 0.525 | 0.147 | | | | | | | | | |
| Sample 6 | 0.870 | 0.090 | 0.515 | 0.294 | 0.758 | | | | | | | | |
| Sample 7 | 0.430 | 0.003 | 0.055 | 1.000 | 0.517 | 0.279 | | | | | | | |
| Sample 8 | 0.890 | 0.013 | 0.440 | 0.326 | 0.816 | 0.823 | 0.375 | | | | | | |
| Sample 9 | 0.276 | 0.380 | 0.518 | 0.056 | 0.295 | 0.248 | 0.030 | 0.054 | | | | | |
| Sample 10 | 0.689 | 0.006 | 0.161 | 0.636 | 0.751 | 0.545 | 0.798 | 0.475 | 0.024 | | | | |
| Sample 11 | 0.890 | 0.012 | 0.223 | 0.791 | 0.805 | 0.656 | 0.758 | 0.587 | 0.039 | 0.670 | | | |
| Sample 12 | 0.198 | 0.620 | 0.371 | 0.021 | 0.227 | 0.156 | 0.023 | 0.119 | 0.787 | 0.048 | 0.012 | | |
| Sample 13 | 0.125 | 0.004 | 0.033 | 0.602 | 0.249 | 0.079 | 0.546 | 0.164 | 0.008 | 0.573 | 0.215 | 0.005 | |

**Solid: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.005 | | | | | | | | | | | | |
| Sample 3 | | | | | | | | | | | | | |
| Sample 4 | | 0.012 | | | | | | | | | | | |
| Sample 5 | | 0.036 | | | | | | | | | | | |
| Sample 6 | | | | | | | | | | | | | |
| Sample 7 | | 0.003 | | | | | | | | | | | |
| Sample 8 | | 0.013 | | | | | | | | | | | |
| Sample 9 | | | | | | | 0.030 | | | | | | |
| Sample 10 | | 0.006 | | | | | | | 0.024 | | | | |
| Sample 11 | | 0.012 | | | | | | | 0.039 | | | | |
| Sample 12 | | | | 0.021 | | | 0.023 | | | 0.048 | 0.012 | | |
| Sample 13 | | 0.004 | 0.033 | | | | | | 0.008 | | | 0.005 | |

Table 48. *Reliable* T-test results (top) and significant results only (bottom)

**Reliable**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.814 | | | | | | | | | | | | |
| Sample 3 | 0.467 | 0.731 | | | | | | | | | | | |
| Sample 4 | 0.445 | 0.601 | 0.840 | | | | | | | | | | |
| Sample 5 | 0.005 | 0.025 | 0.131 | 0.245 | | | | | | | | | |
| Sample 6 | 0.364 | 0.714 | 0.449 | 0.647 | 0.076 | | | | | | | | |
| Sample 7 | 0.066 | 0.029 | 0.007 | 0.010 | 0.001 | 0.013 | | | | | | | |
| Sample 8 | 0.188 | 0.070 | 0.896 | 0.892 | 0.246 | 0.596 | 0.002 | | | | | | |
| Sample 9 | 0.492 | 0.503 | 0.434 | 0.447 | 0.049 | 0.870 | 0.016 | 0.232 | | | | | |
| Sample 10 | 0.505 | 0.348 | 0.069 | 0.056 | 0.032 | 0.245 | 0.365 | 0.045 | 0.283 | | | | |
| Sample 11 | 0.874 | 0.680 | 0.775 | 0.684 | 0.124 | 1.000 | 0.025 | 0.283 | 0.824 | 0.396 | | | |
| Sample 12 | 0.114 | 0.056 | 0.236 | 0.427 | 0.866 | 0.124 | 0.000 | 0.328 | 0.069 | 0.004 | 0.042 | | |
| Sample 13 | 0.396 | 0.544 | 0.233 | 0.213 | 0.012 | 0.388 | 0.073 | 0.031 | 0.601 | 0.672 | 0.458 | 0.014 | |

**Reliable: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | | | | | | | | | | | | | |
| Sample 3 | | | | | | | | | | | | | |
| Sample 4 | | | | | | | | | | | | | |
| Sample 5 | 0.005 | 0.025 | | | | | | | | | | | |
| Sample 6 | | | | | | | | | | | | | |
| Sample 7 | | 0.029 | 0.007 | 0.010 | 1E-03 | 0.013 | | | | | | | |
| Sample 8 | | | | | | | 0.002 | | | | | | |
| Sample 9 | | | | | 0.049 | | 0.016 | | | | | | |
| Sample 10 | | | | | 0.032 | | | 0.045 | | | | | |
| Sample 11 | | | | | | | 0.025 | | | | | | |
| Sample 12 | | | | | | | 0.000 | | | 0.004 | 0.042 | | |
| Sample 13 | | | | | 0.012 | | | 0.031 | | | | 0.014 | |

Table 49. *Powerful* T-test results (top) and significant results only (bottom)

**Powerful**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.546 | | | | | | | | | | | | |
| Sample 3 | 0.002 | 0.006 | | | | | | | | | | | |
| Sample 4 | 0.119 | 0.455 | 0.043 | | | | | | | | | | |
| Sample 5 | 0.000 | 0.027 | 0.588 | 0.030 | | | | | | | | | |
| Sample 6 | 0.012 | 0.111 | 0.865 | 0.086 | 0.455 | | | | | | | | |
| Sample 7 | 0.130 | 0.070 | 0.000 | 0.021 | 0.000 | 0.002 | | | | | | | |
| Sample 8 | 0.409 | 0.693 | 0.079 | 0.732 | 0.068 | 0.223 | 0.045 | | | | | | |
| Sample 9 | 0.580 | 0.778 | 0.030 | 0.564 | 0.011 | 0.105 | 0.019 | 0.897 | | | | | |
| Sample 10 | 0.378 | 0.199 | 0.000 | 0.025 | 0.000 | 0.002 | 0.424 | 0.094 | 0.229 | | | | |
| Sample 11 | 0.663 | 0.433 | 0.001 | 0.076 | 0.000 | 0.006 | 0.297 | 0.213 | 0.182 | 0.712 | | | |
| Sample 12 | 0.164 | 0.161 | 0.130 | 0.885 | 0.070 | 0.164 | 0.021 | 0.712 | 0.433 | 0.021 | 0.068 | | |
| Sample 13 | 0.119 | 0.269 | 0.083 | 1.000 | 0.016 | 0.203 | 0.001 | 0.717 | 0.455 | 0.020 | 0.103 | 0.791 | |

**Powerful: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | | | | | | | | | | | | | |
| Sample 3 | 0.002 | 0.006 | | | | | | | | | | | |
| Sample 4 | | | 0.043 | | | | | | | | | | |
| Sample 5 | 0.000 | 0.027 | | 0.030 | | | | | | | | | |
| Sample 6 | 0.012 | | | | | | | | | | | | |
| Sample 7 | | | 4E-05 | 0.021 | 2E-05 | 0.002 | | | | | | | |
| Sample 8 | | | | | | | 0.045 | | | | | | |
| Sample 9 | | | 0.030 | | 0.011 | | 0.019 | | | | | | |
| Sample 10 | | | 6E-05 | 0.025 | 1E-04 | 0.002 | | | | | | | |
| Sample 11 | | | 0.001 | | 0.000 | 0.006 | | | | | | | |
| Sample 12 | | | | | | | 0.021 | | | 0.021 | | | |
| Sample 13 | | | | | 0.016 | | 0.001 | | | 0.020 | | | |

Table 50. All Command Function T-test results (top) and significant results only (bottom)

**All Command Functions**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.003 | | | | | | | | | | | | |
| Sample 3 | 0.010 | 0.462 | | | | | | | | | | | |
| Sample 4 | 0.023 | 0.163 | 0.630 | | | | | | | | | | |
| Sample 5 | 0.001 | 0.861 | 0.325 | 0.209 | | | | | | | | | |
| Sample 6 | 0.074 | 0.371 | 0.899 | 0.801 | 0.244 | | | | | | | | |
| Sample 7 | 1E-06 | 2E-03 | 3E-04 | 6E-04 | 2E-03 | 7E-04 | | | | | | | |
| Sample 8 | 0.010 | 0.712 | 0.271 | 0.211 | 0.782 | 0.270 | 0.006 | | | | | | |
| Sample 9 | 1E-04 | 6E-02 | 1E-02 | 1E-02 | 3E-02 | 1E-02 | 1E-01 | 1E-01 | | | | | |
| Sample 10 | 3E-05 | 2E-02 | 4E-03 | 2E-03 | 2E-02 | 4E-03 | 5E-01 | 1E-01 | 5E-01 | | | | |
| Sample 11 | 0.006 | 0.790 | 0.325 | 0.170 | 0.891 | 0.293 | 0.004 | 0.904 | 0.086 | 0.004 | | | |
| Sample 12 | 0.049 | 0.572 | 0.908 | 0.654 | 0.527 | 0.835 | 0.002 | 0.392 | 0.020 | 0.005 | 0.304 | | |
| Sample 13 | 4E-04 | 9E-02 | 4E-02 | 1E-02 | 1E-01 | 2E-02 | 1E-01 | 3E-01 | 7E-01 | 3E-01 | 7E-02 | 2E-02 | |

**All Comand Functions: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.003 | | | | | | | | | | | | |
| Sample 3 | 0.010 | | | | | | | | | | | | |
| Sample 4 | 0.023 | | | | | | | | | | | | |
| Sample 5 | 0.001 | | | | | | | | | | | | |
| Sample 6 | | | | | | | | | | | | | |
| Sample 7 | 1E-06 | 2E-03 | 3E-04 | 6E-04 | 2E-03 | 7E-04 | | | | | | | |
| Sample 8 | 0.010 | | | | | | 0.006 | | | | | | |
| Sample 9 | 1E-04 | | 0.014 | 0.011 | 0.031 | 0.014 | | | | | | | |
| Sample 10 | 3E-05 | 0.019 | 0.004 | 2E-03 | 0.021 | 0.004 | | | | | | | |
| Sample 11 | 0.006 | | | | | | 0.004 | | | 0.004 | | | |
| Sample 12 | 0.049 | | | | | | 2E-03 | | 0.020 | 0.005 | | | |
| Sample 13 | 4E-04 | | 0.035 | 0.011 | | 0.017 | | | | | | 0.018 | |

## Appendix F. Post Hoc Wilcoxon Signed Ranks Test Results

Table 51. *Dark* Wilcoxon signed ranks test results (top) and significant results only (bottom)

**Dark**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 1E-04 | | | | | | | | | | | | |
| Sample 3 | 0.011 | 0.002 | | | | | | | | | | | |
| Sample 4 | 0.735 | 0.000 | 0.002 | | | | | | | | | | |
| Sample 5 | 0.398 | 0.000 | 0.001 | 0.594 | | | | | | | | | |
| Sample 6 | 0.178 | 0.000 | 0.001 | 0.345 | 0.593 | | | | | | | | |
| Sample 7 | 2E-03 | 9E-02 | 1E-01 | 5E-04 | 1E-03 | 1E-03 | | | | | | | |
| Sample 8 | 0.008 | 0.008 | 0.925 | 0.002 | 0.006 | 0.004 | 0.132 | | | | | | |
| Sample 9 | 0.333 | 0.000 | 0.061 | 0.110 | 0.099 | 0.099 | 0.001 | 0.019 | | | | | |
| Sample 10 | 0.041 | 0.003 | 0.776 | 0.007 | 0.012 | 0.009 | 0.084 | 0.551 | 0.056 | | | | |
| Sample 11 | 0.012 | 0.024 | 0.783 | 0.005 | 0.007 | 0.007 | 0.156 | 0.820 | 0.016 | 0.453 | | | |
| Sample 12 | 0.397 | 0.000 | 0.072 | 0.142 | 0.133 | 0.078 | 0.001 | 0.033 | 0.735 | 0.055 | 0.011 | | |
| Sample 13 | 0.158 | 0.004 | 0.277 | 0.083 | 0.021 | 0.018 | 0.030 | 0.255 | 0.532 | 0.485 | 0.196 | 0.480 | |

**Dark: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 1E-04 | | | | | | | | | | | | |
| Sample 3 | 0.011 | 2E-03 | | | | | | | | | | | |
| Sample 4 | | 6E-05 | 2E-03 | | | | | | | | | | |
| Sample 5 | | 2E-04 | 8E-04 | | | | | | | | | | |
| Sample 6 | | 1E-04 | 7E-04 | | | | | | | | | | |
| Sample 7 | 2E-03 | | | 5E-04 | 1E-03 | 1E-03 | | | | | | | |
| Sample 8 | 0.008 | 0.008 | | 0.002 | 0.006 | 0.004 | | | | | | | |
| Sample 9 | | 2E-04 | | | | | 0.001 | 0.019 | | | | | |
| Sample 10 | 0.041 | 0.003 | | 0.007 | 0.012 | 0.009 | | | | | | | |
| Sample 11 | 0.012 | 0.024 | | 0.005 | 0.007 | 0.007 | | | 0.016 | | | | |
| Sample 12 | | 2E-04 | | | | | 0.001 | 0.033 | | | 0.011 | | |
| Sample 13 | | 4E-03 | | | 0.021 | 0.018 | 0.030 | | | | | | |

Table 52. *Dynamic* Wilcoxon signed ranks test results (top) and significant results only (bottom)

**Dynamic**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 6E-06 | | | | | | | | | | | | |
| Sample 3 | 2E-03 | 2E-05 | | | | | | | | | | | |
| Sample 4 | 0.047 | 0.000 | 0.012 | | | | | | | | | | |
| Sample 5 | 0.003 | 0.000 | 0.000 | 0.000 | | | | | | | | | |
| Sample 6 | 0.019 | 0.000 | 0.000 | 0.000 | 0.307 | | | | | | | | |
| Sample 7 | 2E-05 | 1E-03 | 3E-03 | 9E-05 | 3E-06 | 4E-06 | | | | | | | |
| Sample 8 | 2E-05 | 5E-03 | 2E-04 | 3E-05 | 2E-06 | 3E-06 | 2E-01 | | | | | | |
| Sample 9 | 9E-05 | 3E-04 | 2E-02 | 1E-03 | 3E-06 | 3E-06 | 3E-01 | 3E-02 | | | | | |
| Sample 10 | 1E-03 | 1E-04 | 2E-01 | 1E-02 | 8E-06 | 2E-05 | 4E-02 | 4E-03 | 3E-01 | | | | |
| Sample 11 | 1E-04 | 1E-04 | 5E-02 | 1E-03 | 2E-06 | 4E-06 | 1E-01 | 4E-03 | 8E-01 | 5E-01 | | | |
| Sample 12 | 6E-04 | 6E-05 | 2E-01 | 2E-03 | 1E-06 | 2E-06 | 9E-02 | 3E-03 | 4E-01 | 9E-01 | 4E-01 | | |
| Sample 13 | 0.935 | 0.000 | 0.001 | 0.122 | 0.000 | 0.007 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | |

**Dynamic: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 6E-06 | | | | | | | | | | | | |
| Sample 3 | 2E-03 | 2E-05 | | | | | | | | | | | |
| Sample 4 | 0.047 | 4E-06 | 0.012 | | | | | | | | | | |
| Sample 5 | 0.003 | 2E-06 | 2E-06 | 2E-05 | | | | | | | | | |
| Sample 6 | 0.019 | 2E-06 | 8E-06 | 7E-05 | | | | | | | | | |
| Sample 7 | 2E-05 | 1E-03 | 0.003 | 9E-05 | 3E-06 | 4E-06 | | | | | | | |
| Sample 8 | 2E-05 | 0.005 | 0.000 | 3E-05 | 2E-06 | 3E-06 | | | | | | | |
| Sample 9 | 9E-05 | 0.000 | 0.022 | 1E-03 | 3E-06 | 3E-06 | | 0.035 | | | | | |
| Sample 10 | 1E-03 | 1E-04 | | 0.012 | 8E-06 | 2E-05 | 0.039 | 0.004 | | | | | |
| Sample 11 | 1E-04 | 1E-04 | 0.047 | 1E-03 | 2E-06 | 4E-06 | | 0.004 | | | | | |
| Sample 12 | 6E-04 | 6E-05 | | 0.002 | 1E-06 | 2E-06 | | 0.003 | | | | | |
| Sample 13 | | 8E-06 | 0.001 | | 2E-04 | 0.007 | 4E-05 | 2E-05 | 2E-04 | 5E-04 | 1E-04 | 6E-04 | |

Table 53. *Sharp* Wilcoxon signed ranks test results (top) and significant results only (bottom)

**Sharp**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 0.345 | | | | | | | | | | | | |
| Sample 3 | 4E-05 | 8E-05 | | | | | | | | | | | |
| Sample 4 | 6E-05 | 2E-04 | 7E-01 | | | | | | | | | | |
| Sample 5 | 1E-03 | 5E-03 | 2E-01 | 1E-01 | | | | | | | | | |
| Sample 6 | 2E-03 | 6E-03 | 4E-02 | 5E-02 | 4E-01 | | | | | | | | |
| Sample 7 | 0.110 | 0.328 | 0.001 | 0.001 | 0.035 | 0.053 | | | | | | | |
| Sample 8 | 0.004 | 0.019 | 0.013 | 0.008 | 0.362 | 0.670 | 0.060 | | | | | | |
| Sample 9 | 0.008 | 0.019 | 0.007 | 0.012 | 0.198 | 0.352 | 0.133 | 0.666 | | | | | |
| Sample 10 | 4E-03 | 2E-02 | 2E-02 | 2E-02 | 2E-01 | 6E-01 | 9E-02 | 9E-01 | 7E-01 | | | | |
| Sample 11 | 0.001 | 0.008 | 0.093 | 0.052 | 0.705 | 0.841 | 0.050 | 0.551 | 0.209 | 0.308 | | | |
| Sample 12 | 0.043 | 0.213 | 0.003 | 0.003 | 0.034 | 0.164 | 0.616 | 0.424 | 0.490 | 0.209 | 0.016 | | |
| Sample 13 | 0.361 | 0.116 | 0.000 | 0.000 | 0.000 | 0.001 | 0.028 | 0.002 | 0.001 | 0.001 | 0.001 | 0.021 | |

**Sharp: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | | | | | | | | | | | | | |
| Sample 3 | 4E-05 | 8E-05 | | | | | | | | | | | |
| Sample 4 | 6E-05 | 2E-04 | | | | | | | | | | | |
| Sample 5 | 1E-03 | 0.005 | | | | | | | | | | | |
| Sample 6 | 2E-03 | 0.006 | 0.037 | 0.049 | | | | | | | | | |
| Sample 7 | | | 7E-04 | 1E-03 | 0.035 | | | | | | | | |
| Sample 8 | 0.004 | 0.019 | 0.013 | 0.008 | | | | | | | | | |
| Sample 9 | 0.008 | 0.019 | 0.007 | 0.012 | | | | | | | | | |
| Sample 10 | 4E-03 | 0.017 | 0.015 | 0.021 | | | | | | | | | |
| Sample 11 | 0.001 | 0.008 | | | | | | | | | | | |
| Sample 12 | 0.043 | | 0.003 | 0.003 | 0.034 | | | | | | 0.016 | | |
| Sample 13 | | | 4E-05 | 9E-05 | 3E-04 | 7E-04 | 0.028 | 0.002 | 1E-03 | 1E-03 | 7E-04 | 0.021 | |

Table 54. *Expressive* Wilcoxon signed ranks test results (top) and significant results only (bottom)

**Expressive**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 8E-06 | | | | | | | | | | | | |
| Sample 3 | 2E-03 | 4E-05 | | | | | | | | | | | |
| Sample 4 | 0.068 | 0.000 | 0.006 | | | | | | | | | | |
| Sample 5 | 0.131 | 0.000 | 0.000 | 0.000 | | | | | | | | | |
| Sample 6 | 0.131 | 0.000 | 0.000 | 0.000 | 1.000 | | | | | | | | |
| Sample 7 | 2E-04 | 2E-03 | 8E-02 | 1E-03 | 1E-05 | 1E-05 | | | | | | | |
| Sample 8 | 4E-04 | 3E-04 | 2E-01 | 6E-04 | 1E-05 | 8E-06 | 9E-01 | | | | | | |
| Sample 9 | 0.000 | 0.000 | 0.781 | 0.012 | 0.000 | 0.000 | 0.235 | 0.266 | | | | | |
| Sample 10 | 6E-03 | 2E-04 | 8E-01 | 6E-02 | 1E-04 | 1E-04 | 5E-02 | 1E-01 | 6E-01 | | | | |
| Sample 11 | 0.001 | 0.000 | 0.948 | 0.007 | 0.000 | 0.000 | 0.232 | 0.048 | 0.740 | 0.670 | | | |
| Sample 12 | 2E-03 | 7E-05 | 9E-01 | 3E-02 | 4E-05 | 3E-05 | 1E-01 | 7E-02 | 5E-01 | 9E-01 | 7E-01 | | |
| Sample 13 | 0.657 | 0.000 | 0.001 | 0.008 | 0.182 | 0.182 | 0.000 | 0.000 | 0.001 | 0.002 | 0.001 | 0.001 | |

**Expressive: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 8E-06 | | | | | | | | | | | | |
| Sample 3 | 2E-03 | 4E-05 | | | | | | | | | | | |
| Sample 4 | | 3E-06 | 0.006 | | | | | | | | | | |
| Sample 5 | | 2E-06 | 2E-05 | 2E-04 | | | | | | | | | |
| Sample 6 | | 2E-06 | 2E-05 | 3E-04 | | | | | | | | | |
| Sample 7 | 2E-04 | 2E-03 | | 1E-03 | 1E-05 | 1E-05 | | | | | | | |
| Sample 8 | 4E-04 | 3E-04 | | 0.001 | 1E-05 | 8E-06 | | | | | | | |
| Sample 9 | 0.000 | 0.000 | | 0.012 | 6E-05 | 4E-05 | | | | | | | |
| Sample 10 | 6E-03 | 2E-04 | | | 1E-04 | 1E-04 | | | | | | | |
| Sample 11 | 0.001 | 6E-05 | | 0.007 | 7E-05 | 5E-05 | | 0.048 | | | | | |
| Sample 12 | 2E-03 | 7E-05 | | 0.029 | 4E-05 | 3E-05 | | | | | | | |
| Sample 13 | | 4E-06 | 7E-04 | 0.008 | | | 9E-05 | 1E-04 | 7E-04 | 2E-03 | 1E-03 | 1E-03 | |

Table 55. *Strong* Wilcoxon signed ranks test results (top) and significant results only (bottom)

**Strong**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 3E-04 | | | | | | | | | | | | |
| Sample 3 | 0.004 | 0.301 | | | | | | | | | | | |
| Sample 4 | 1E-04 | 3E-01 | 3E-02 | | | | | | | | | | |
| Sample 5 | 0.014 | 0.014 | 0.179 | 0.001 | | | | | | | | | |
| Sample 6 | 0.013 | 0.039 | 0.301 | 0.000 | 0.459 | | | | | | | | |
| Sample 7 | 0.824 | 0.000 | 0.002 | 0.000 | 0.019 | 0.004 | | | | | | | |
| Sample 8 | 6E-05 | 7E-01 | 5E-02 | 5E-01 | 2E-03 | 2E-03 | 1E-05 | | | | | | |
| Sample 9 | 0.037 | 0.033 | 0.339 | 0.003 | 0.723 | 0.974 | 0.010 | 0.001 | | | | | |
| Sample 10 | 0.308 | 0.000 | 0.028 | 0.000 | 0.396 | 0.049 | 0.083 | 0.000 | 0.196 | | | | |
| Sample 11 | 0.056 | 0.006 | 0.159 | 0.001 | 0.977 | 0.679 | 0.007 | 0.000 | 0.687 | 0.249 | | | |
| Sample 12 | 3E-04 | 8E-01 | 3E-01 | 2E-01 | 7E-03 | 2E-02 | 1E-04 | 3E-01 | 5E-02 | 1E-03 | 6E-03 | | |
| Sample 13 | 0.799 | 0.000 | 0.009 | 0.000 | 0.013 | 0.004 | 0.636 | 0.000 | 0.039 | 0.213 | 0.028 | 0.000 | |

**Strong: Significant Values**

| | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample 1 | | | | | | | | | | | | | |
| Sample 2 | 3E-04 | | | | | | | | | | | | |
| Sample 3 | 0.004 | | | | | | | | | | | | |
| Sample 4 | 1E-04 | | 0.033 | | | | | | | | | | |
| Sample 5 | 0.014 | 0.014 | | 6E-04 | | | | | | | | | |
| Sample 6 | 0.013 | 0.039 | | 4E-04 | | | | | | | | | |
| Sample 7 | | 1E-04 | 2E-03 | 1E-05 | 0.019 | 0.004 | | | | | | | |
| Sample 8 | 6E-05 | | | | 0.002 | 0.002 | 1E-05 | | | | | | |
| Sample 9 | 0.037 | 0.033 | | 3E-03 | | | 0.010 | 0.001 | | | | | |
| Sample 10 | | 5E-04 | 0.028 | 8E-05 | | 0.049 | | 4E-05 | | | | | |
| Sample 11 | | 0.006 | | 0.001 | | | 0.007 | 0.000 | | | | | |
| Sample 12 | 3E-04 | | | | 0.007 | 0.022 | 1E-04 | | | 0.001 | 0.006 | | |
| Sample 13 | | 2E-04 | 0.009 | 5E-05 | 0.013 | 0.004 | | 6E-05 | 0.039 | | 0.028 | 2E-04 | |

## Appendix G. Detailed Sample Sound Rankings for Command Function Groups

Table 56. Sound Sample Rankings: *Four Sounds, Group 1*

| 4 Sounds, Group 1 | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1st | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2nd | 1 | 0 | 1 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3rd | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | 1 |
| 4th | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 0 | 1 | 1 | 0 |
| 5th | 0 | 1 | 0 | 1 | 2 | 0 | 2 | 0 | 0 | 2 | 1 | 1 | 1 |
| 6th | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 0 | 0 | 0 |
| Total Score | 23 | 4 | 9 | 13 | 7 | 21 | 5 | 11 | 7 | 6 | 9 | 9 | 6 |

Table 57. Sound Sample Rankings: *Four Sounds, Group 2*

| 4 Sounds, Group 2 | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1st | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 2nd | 0 | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 3rd | 0 | 1 | 3 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 2 | 0 | 1 |
| 4th | 2 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 0 | 0 |
| 5th | 2 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 2 |
| 6th | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| Total Score | 10 | 14 | 17 | 10 | 21 | 2 | 9 | 10 | 8 | 6 | 13 | 22 | 8 |

Table 58. Sound Sample Rankings: *Four Sounds, Group 3*

| | | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 Sounds, Group 3 | 1st | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 2nd | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| | 3rd | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| | 4th | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| | 5th | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| | 6th | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total Score | | 8 | 4 | 4 | 12 | 7 | 8 | 0 | 8 | 9 | 0 | 6 | 7 | 4 |

Table 59. Sound Sample Rankings: *Four Sounds, Group 4*

| | | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 Sounds, Group 4 | 1st | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| | 2nd | 2 | 1 | 2 | 2 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 3 |
| | 3rd | 2 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 2 | 3 | 1 | 0 |
| | 4th | 0 | 2 | 0 | 2 | 0 | 3 | 1 | 1 | 2 | 0 | 0 | 1 | 1 |
| | 5th | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 2 | 0 | 1 | 0 |
| | 6th | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total Score | | 24 | 21 | 20 | 20 | 6 | 14 | 10 | 17 | 10 | 18 | 18 | 14 | 18 |

Table 60. Sound Sample Rankings: *Three Sounds, Group 1*

| | | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 Sounds, Group 1 | 1st | 3 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 2nd | 1 | 0 | 1 | 0 | 0 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| | 3rd | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 2 | 0 | 0 | 1 | 2 | 1 |
| | 4th | 1 | 0 | 1 | 1 | 2 | 0 | 0 | 1 | 2 | 0 | 1 | 1 | 0 |
| | 5th | 0 | 2 | 1 | 1 | 2 | 0 | 2 | 0 | 0 | 2 | 1 | 1 | 2 |
| | 6th | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 0 | 0 | 0 |
| Total Score | | 26 | 6 | 11 | 19 | 10 | 25 | 5 | 16 | 12 | 6 | 15 | 13 | 8 |

Table 61. Sound Sample Rankings: *Three Sounds, Group 2*

| | | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 Sounds, Group 2 | 1st | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| | 2nd | 0 | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| | 3rd | 0 | 1 | 3 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 2 | 0 | 1 |
| | 4th | 2 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 0 | 0 |
| | 5th | 2 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 2 |
| | 6th | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| Total Score | | 10 | 14 | 17 | 10 | 21 | 2 | 9 | 10 | 8 | 6 | 13 | 22 | 8 |

Table 62. Sound Sample Rankings: *Three Sounds, Group 3*

| | | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 Sounds, Group 3 | 1st | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| | 2nd | 3 | 1 | 2 | 2 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 3 |
| | 3rd | 2 | 1 | 1 | 1 | 2 | 1 | 0 | 1 | 1 | 2 | 3 | 1 | 0 |
| | 4th | 0 | 2 | 0 | 2 | 0 | 3 | 1 | 2 | 2 | 0 | 0 | 2 | 1 |
| | 5th | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 2 | 2 | 0 | 1 | 1 |
| | 6th | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total Score | | 29 | 23 | 22 | 26 | 10 | 18 | 10 | 20 | 14 | 18 | 18 | 17 | 20 |

Table 63. Sound Sample Rankings: *Two Sounds, Group 1*

| | | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 Sounds, Group 1 | 1st | 3 | 1 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| | 2nd | 3 | 0 | 2 | 1 | 0 | 4 | 1 | 3 | 1 | 0 | 0 | 1 | 2 |
| | 3rd | 2 | 1 | 1 | 3 | 1 | 2 | 0 | 2 | 1 | 1 | 2 | 3 | 1 |
| | 4th | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 3 | 3 | 0 | 1 | 3 | 1 |
| | 5th | 0 | 3 | 2 | 1 | 3 | 0 | 2 | 0 | 2 | 3 | 1 | 1 | 3 |
| | 6th | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 0 | 0 | 0 |
| Total Score | | 44 | 21 | 22 | 37 | 16 | 37 | 13 | 32 | 23 | 12 | 25 | 28 | 23 |

Table 64. Sound Sample Rankings: *Two Sounds, Group 2*

| | | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Sample 6 | Sample 7 | Sample 8 | Sample 9 | Sample 10 | Sample 11 | Sample 12 | Sample 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 Sounds, Group 2 | 1st | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| | 2nd | 1 | 3 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 |
| | 3rd | 0 | 1 | 3 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 4 | 0 | 1 |
| | 4th | 2 | 1 | 0 | 2 | 0 | 2 | 1 | 1 | 3 | 1 | 1 | 0 | 0 |
| | 5th | 2 | 0 | 0 | 1 | 0 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 |
| | 6th | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| Total Score | | 21 | 22 | 28 | 18 | 25 | 8 | 11 | 14 | 11 | 12 | 21 | 24 | 13 |

## Appendix H. Detailed Average Correlation Coefficients for Command Function Groups

Table 65. Detailed Average Correlation Coefficients: *Four Sounds, Group 1*

| | Set + | Volume Up | Activate | On |
|---|---|---|---|---|
| Set + | 1.000 | | | |
| Volume Up | 0.917 | 1.000 | | |
| Activate | 0.821 | 0.756 | 1.000 | |
| On | 0.814 | 0.802 | 0.725 | 1.000 |
| Avg Correlation | 0.806 | | | |

Table 66. Detailed Average Correlation Coefficients: *Four Sounds, Group 2*

| | Set - | Off | Volume Down | Previous Track |
|---|---|---|---|---|
| Set - | 1.000 | | | |
| Off | 0.742 | 1.000 | | |
| Volume Down | 0.444 | 0.729 | 1.000 | |
| Previous Track | 0.537 | 0.714 | 0.858 | 1.000 |
| Avg Correlation | 0.671 | | | |

Table 67. Detailed Average Correlation Coefficients: *Four Sounds, Group 3*

| | Answer Call | Next Track |
|---|---|---|
| Answer Call | 1.000 | |
| Next Track | 0.620 | 1.000 |
| Avg Correlation | 0.620 | |

Table 68. Detailed Average Correlation Coefficients: *Four Sounds, Group 4*

| | End Call | Media Mode | Play/Pause | Mute On | Mute Off |
|---|---|---|---|---|---|
| End Call | 1.000 | | | | |
| Media Mode | | 1.000 | | | |
| Play/Pause | | | 1.000 | | |
| Mute On | | -0.383 | | 1.000 | |
| Mute Off | | | 0.415 | | 1.000 |
| Avg Correlation | 0.016 | | | | |

Table 69. Detailed Average Correlation Coefficients: *Three Sounds, Group 1*

| | Set + | Volume Up | Activate | On | Next Track |
|---|---|---|---|---|---|
| Set + | 1.000 | | | | |
| Volume Up | 0.917 | 1.000 | | | |
| Activate | 0.821 | 0.756 | 1.000 | | |
| On | 0.814 | 0.802 | 0.725 | 1.000 | |
| Next Track | | | | | 1.000 |
| Avg Correlation | 0.806 | | | | |

Table 70. Detailed Average Correlation Coefficients: *Three Sounds, Group 2*

| | Set - | Off | Volume Down | Previous Track |
|---|---|---|---|---|
| Set - | 1.000 | | | |
| Off | 0.742 | 1.000 | | |
| Volume Down | 0.444 | 0.729 | 1.000 | |
| Previous Track | 0.537 | 0.714 | 0.858 | 1.000 |
| Avg Correlation | 0.671 | | | |

Table 71. Detailed Average Correlation Coefficients: *Three Sounds, Group 3*

| | Answer Call | End Call | Media Mode | Play/Pause | Mute On | Mute Off |
|---|---|---|---|---|---|---|
| Answer Call | 1.000 | | | | | |
| Next Track | 0.620 | | | | | |
| End Call | | 1.000 | | | | |
| Media Mode | | | 1.000 | | | |
| Play/Pause | | | | 1.000 | | |
| Mute On | -0.361 | | -0.383 | | 1.000 | |
| Mute Off | 0.434 | | | 0.415 | | 1.000 |
| Avg Correlation | 0.145 | | | | | |

Table 72. Detailed Average Correlation Coefficients: *Two Sounds, Group 1*

| | Set + | Volume Up | Activate | On | Answer Call | Next Track | Mute On | Play/Pause | Media Mode |
|---|---|---|---|---|---|---|---|---|---|
| Set + | 1.000 | | | | | | | | |
| Volume Up | 0.917 | 1.000 | | | | | | | |
| Activate | 0.821 | 0.756 | 1.000 | | | | | | |
| On | 0.814 | 0.802 | 0.725 | 1.000 | | | | | |
| Answer Call | | 0.363 | | | 1.000 | | | | |
| Next Track | | | | | 0.620 | 1.000 | | | |
| Mute On | | | | | -0.361 | -0.614 | 1.000 | | |
| Play/Pause | | | 0.429 | 0.374 | | | | 1.000 | |
| Media Mode | | | | | | | -0.383 | | 1.000 |
| Avg Correlation | 0.405 | | | | | | | | |

Table 73. Detailed Average Correlation Coefficients: *Two Sounds, Group 2*

| | Set - | Off | Volume Down | Previous Track | End Call | Mute Off |
|---|---|---|---|---|---|---|
| Set - | 1.000 | | | | | |
| Off | 0.742 | 1.000 | | | | |
| Volume Down | 0.444 | 0.729 | 1.000 | | | |
| Previous Track | 0.537 | 0.714 | 0.858 | 1.000 | | |
| End Call | | | | | 1.000 | |
| Mute Off | | | -0.385 | | | 1.000 |
| Avg Correlation | 0.520 | | | | | |

## Appendix I. Amplitude vs Time & Frequency Plots –Haptic Board Voltage Measurement

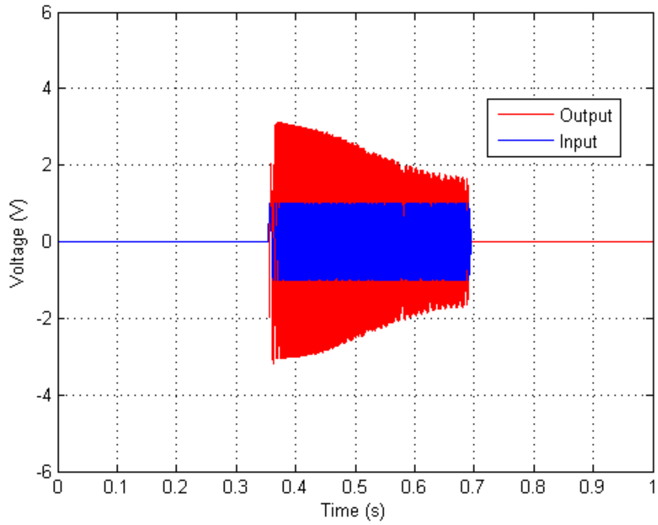



Figure 111. Haptic Board Voltage Output Time Plots – Sine Sweep, 6 dB gain, 35 Volume, No-load Impedance Condition

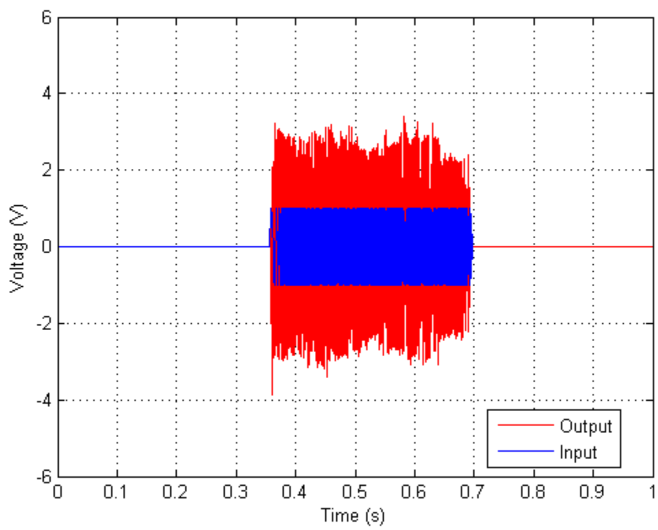



Figure 112. Haptic Board Voltage Output Time Plots – Sine Sweep, 6 dB gain, 36 Volume, No-load Impedance Condition

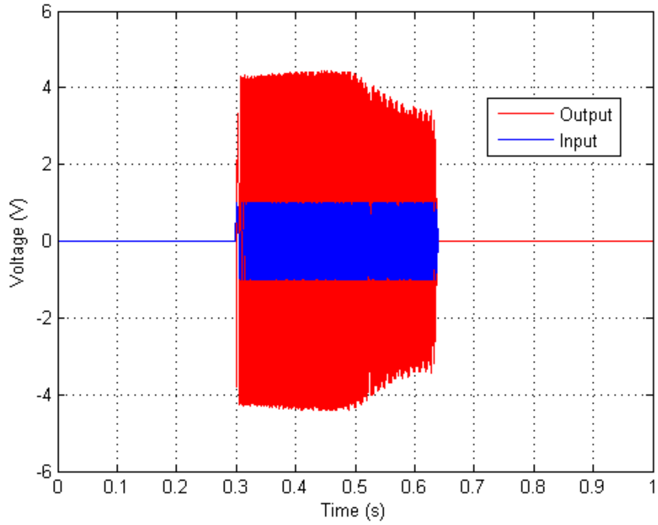



Figure 113. Haptic Board Voltage Output Time Plots – Sine Sweep, 12 dB gain, 35 Volume, No-load Impedance Condition

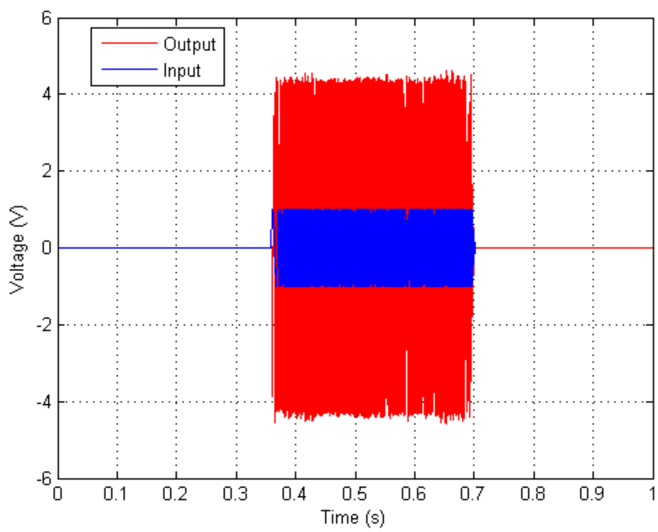



Figure 114. Haptic Board Voltage Output Time Plots – Sine Sweep, 12 dB gain, 36 Volume, No-load Impedance Condition

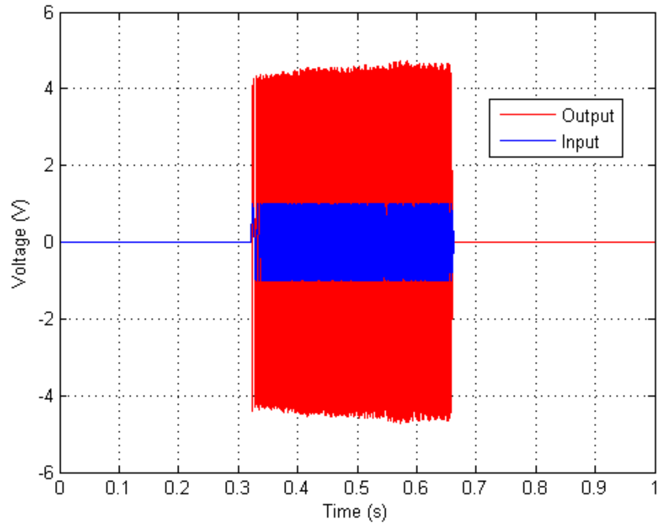



Figure 115. Haptic Board Voltage Output Time Plots – Sine Sweep, 18 dB gain, 35 Volume, No-load Impedance Condition

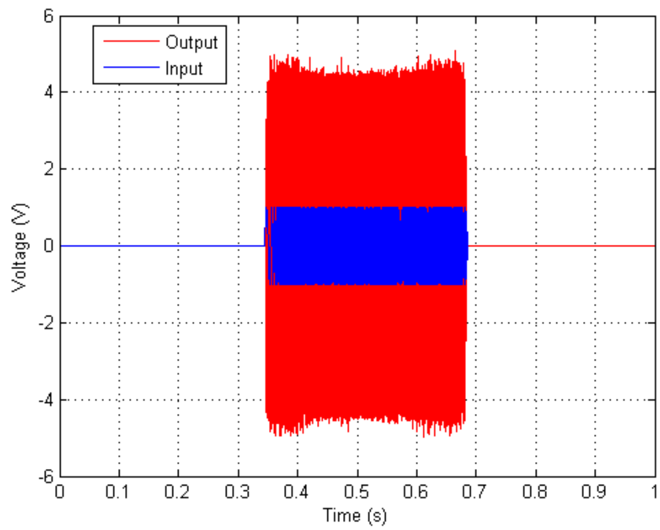



Figure 116. Haptic Board Voltage Output Time Plots – Sine Sweep, 18 dB gain, 36 Volume, No-load Impedance Condition

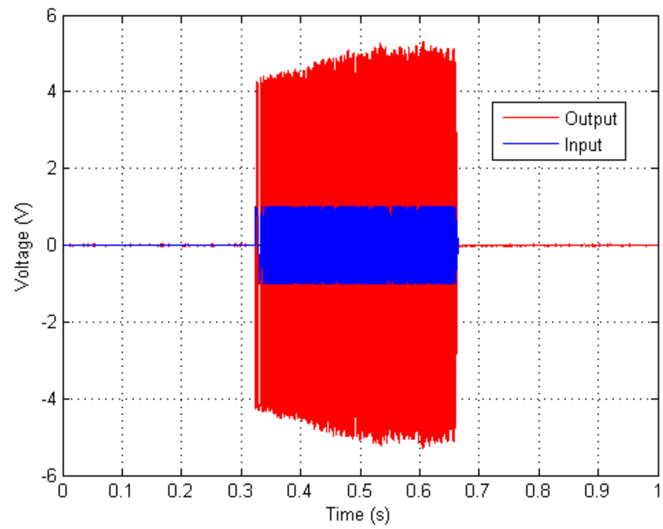



Figure 117. Haptic Board Voltage Output Time Plots – Sine Sweep, 24 dB gain, 35 Volume, No-load Impedance Condition

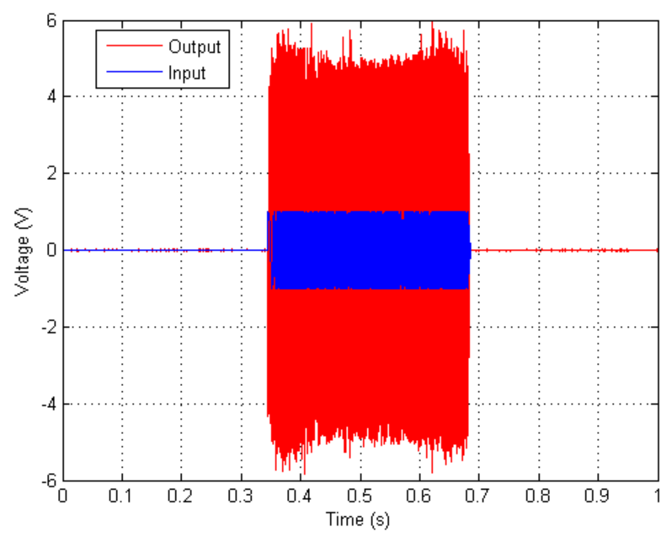



Figure 118. Haptic Board Voltage Output Time Plots – Sine Sweep, 24 dB gain, 36 Volume No-load Impedance Condition

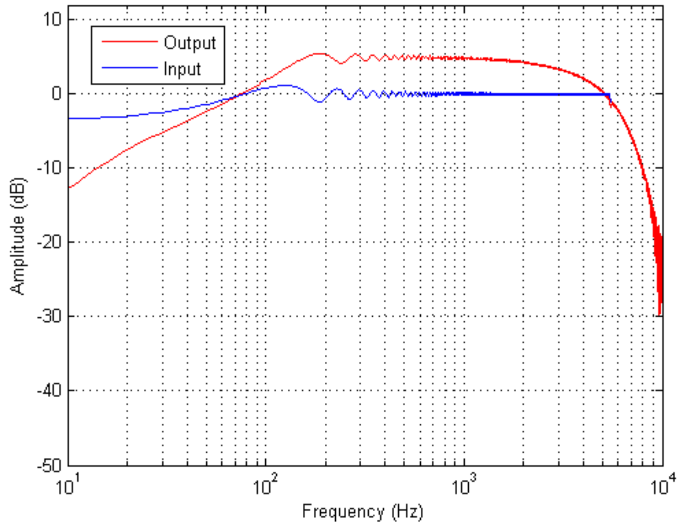



Figure 119. Haptic Board Voltage Output Frequency Plots – Sine Sweep, 6 dB gain, 35 Volume, No-load Impedance Condition

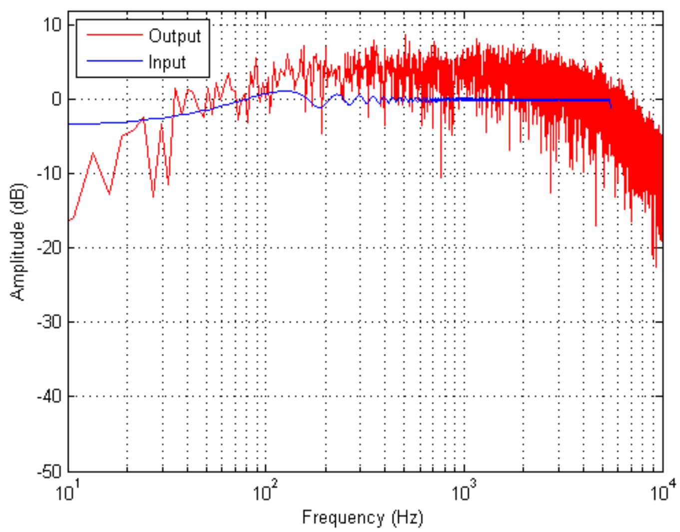



Figure 120. Haptic Board Voltage Output Frequency Plots – Sine Sweep, 6 dB gain, 36 Volume, No-load Impedance Condition

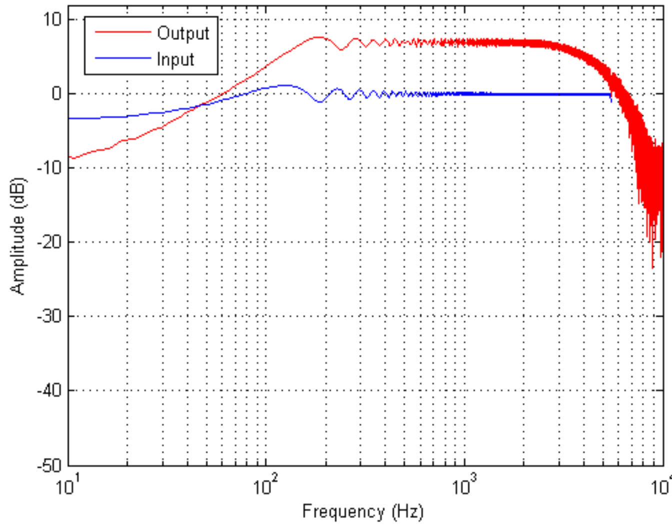



Figure 121. Haptic Board Voltage Output Frequency Plots – Sine Sweep, 12 dB gain, 35 Volume, No-load Impedance Condition

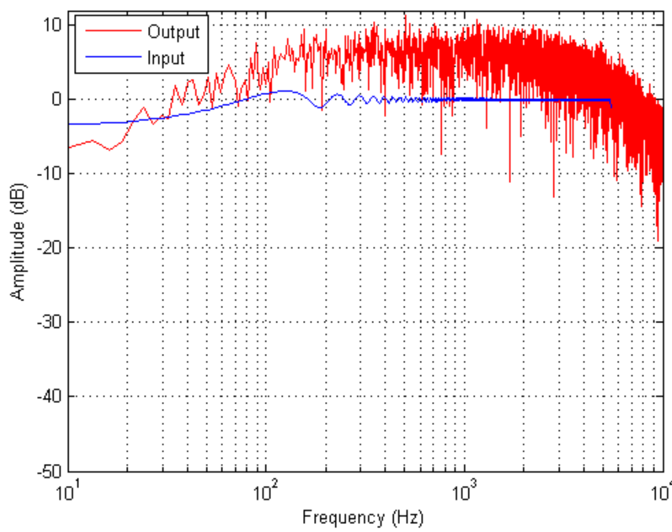



Figure 122. Haptic Board Voltage Output Frequency Plots – Sine Sweep, 12 dB gain, 36 Volume, No-load Impedance Condition

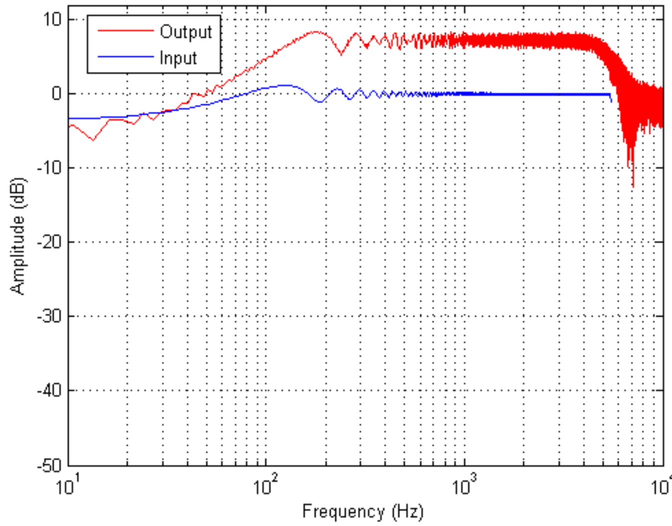



Figure 123. Haptic Board Voltage Output Frequency Plots – Sine Sweep, 18 dB gain, 35 Volume, No-load Impedance Condition

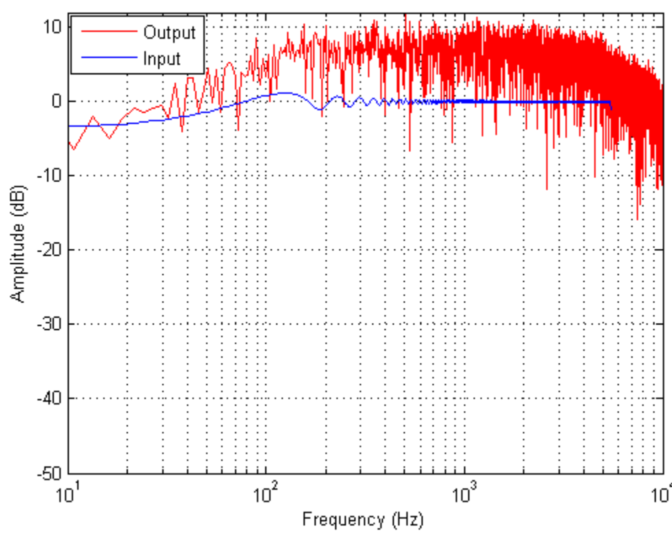



Figure 124. Haptic Board Voltage Output Frequency Plots – Sine Sweep, 18 dB gain, 36 Volume, No-load Impedance Condition

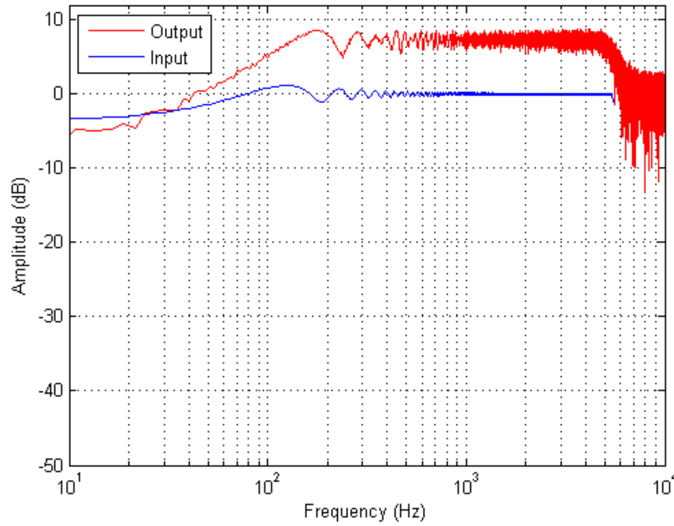



Figure 125. Haptic Board Voltage Output Frequency Plots – Sine Sweep, 24 dB gain, 35 Volume, No-load Impedance Condition

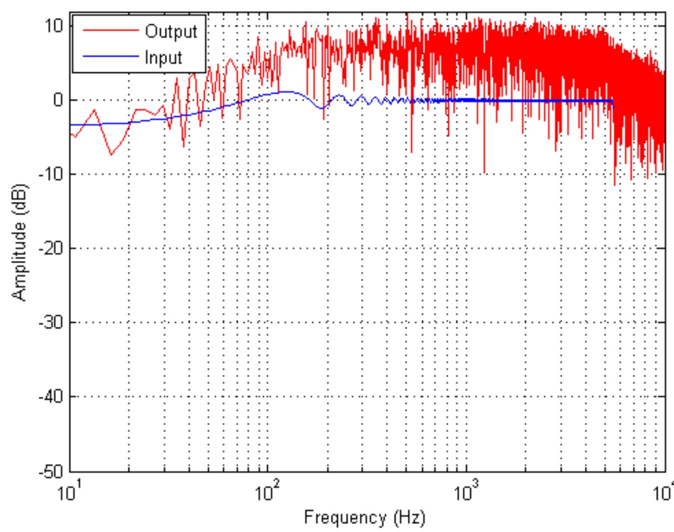



Figure 126. Haptic Board Voltage Output Frequency Plots – Sine Sweep, 24 dB gain, 36 Volume, No-load Impedance Condition

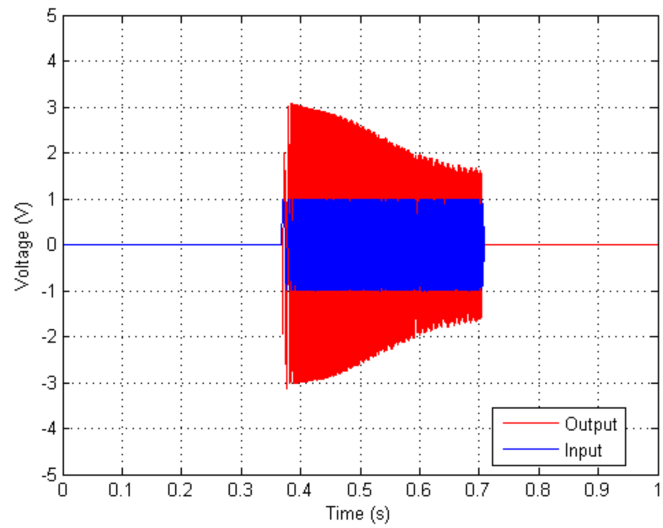



Figure 127. Time Spectrum –Haptic Board Voltage Measurement – Exciter loaded – Sine Sweep, 6 dB Gain, 35 Volume

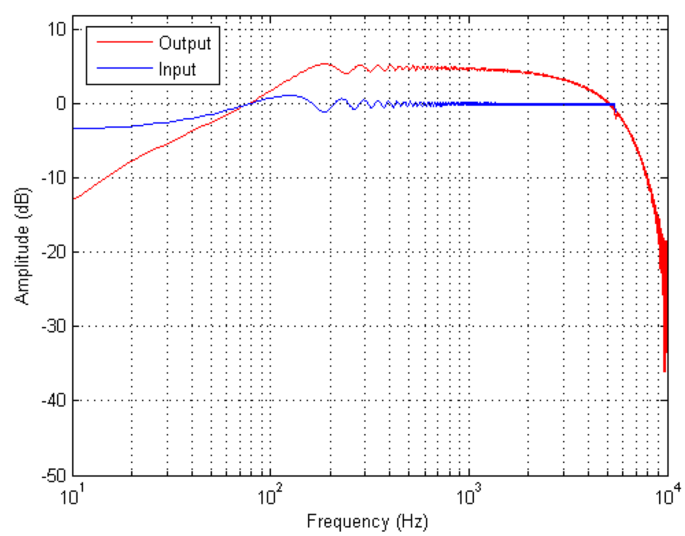



Figure 128. Frequency Spectrum –Haptic Board Voltage Measurement – Exciter loaded – Sine Sweep, 6 dB Gain, 35 Volume

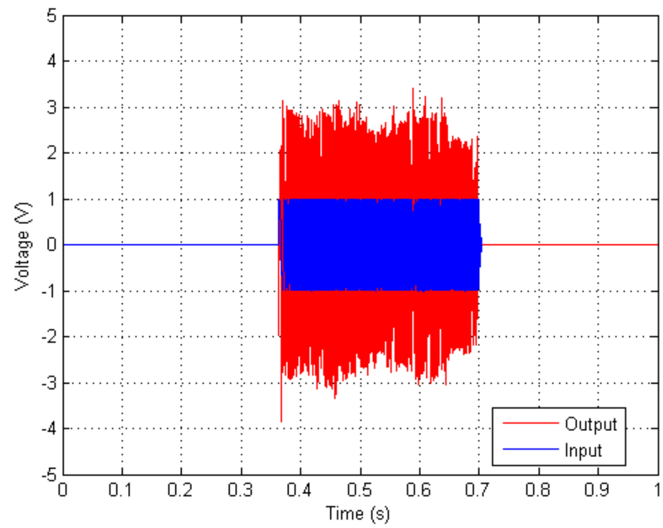



Figure 129. Time Spectrum –Haptic Board Voltage Measurement – Exciter loaded – Sine Sweep, 6 dB Gain, 36 Volume

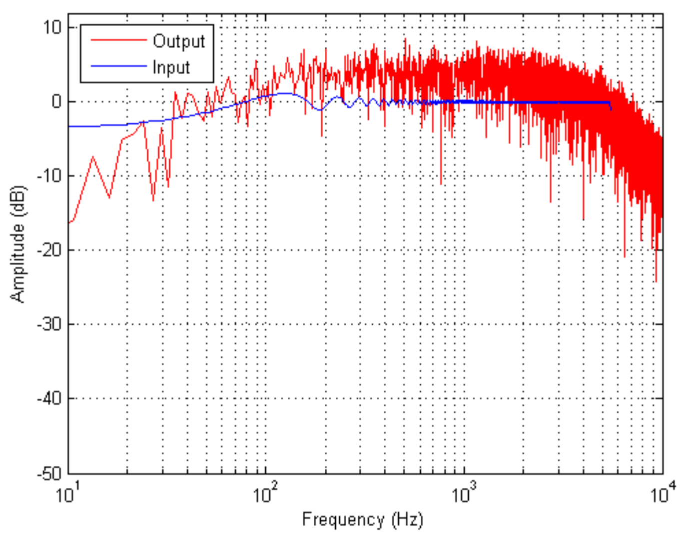



Figure 130. Frequency Spectrum –Haptic Board Voltage Measurement – Exciter loaded – Sine Sweep, 6 dB Gain, 36 Volume

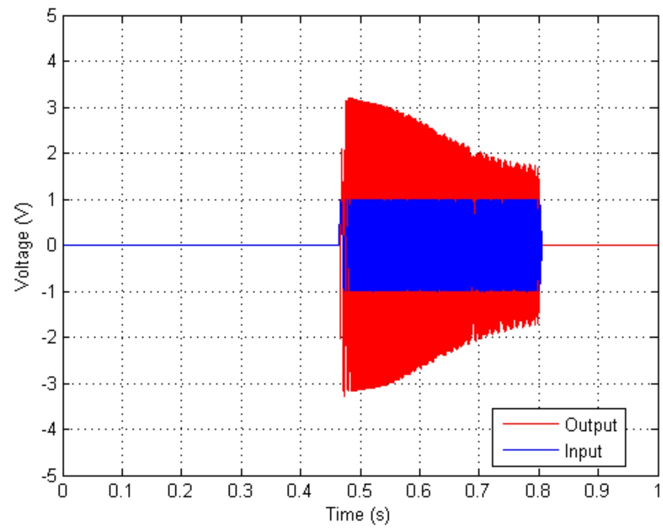



Figure 131. Time Spectrum –Haptic Board Voltage Measurement – Resistor loaded – Sine Sweep, 6 dB Gain, 35 Volume

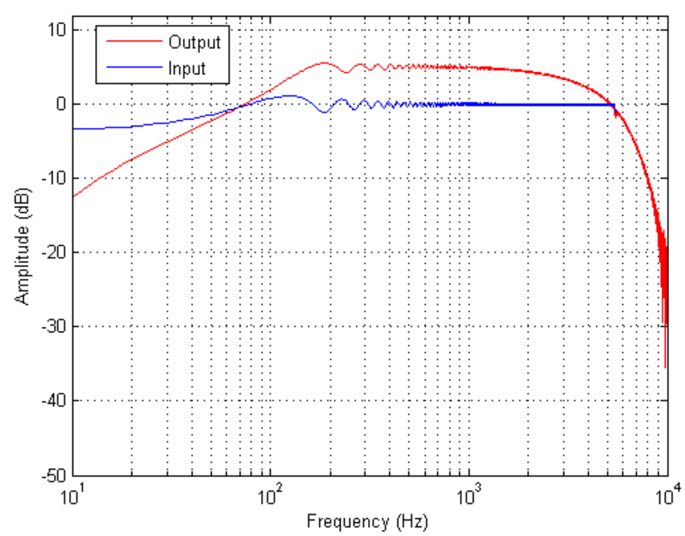



Figure 132. Frequency Spectrum –Haptic Board Voltage Measurement – Resistor loaded – Sine Sweep, 6 dB Gain, 35 Volume

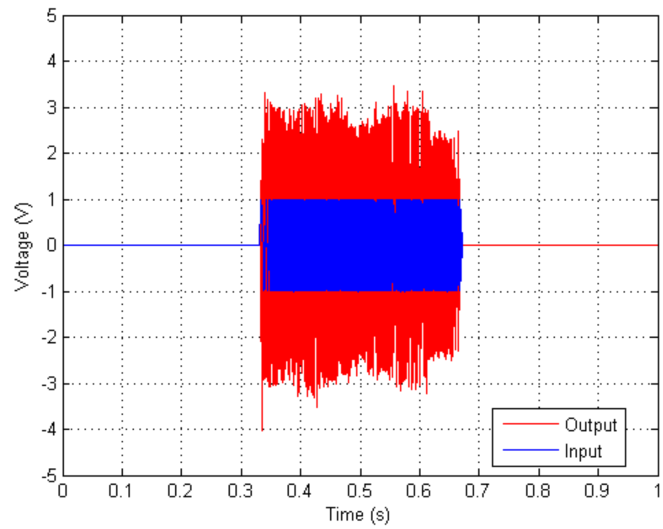



Figure 133. Time Spectrum –Haptic Board Voltage Measurement – Resistor loaded – Sine Sweep, 6 dB Gain, 36 Volume

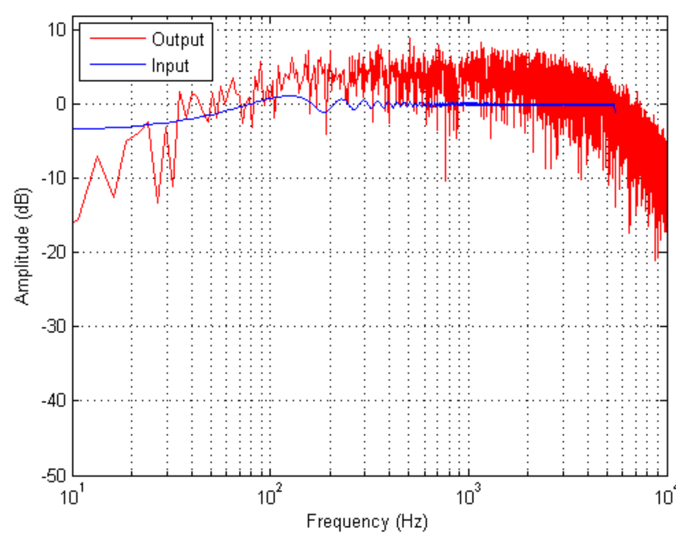



Figure 134. Frequency Spectrum –Haptic Board Voltage Measurement – Resistor loaded – Sine Sweep, 6 dB Gain, 36 Volume

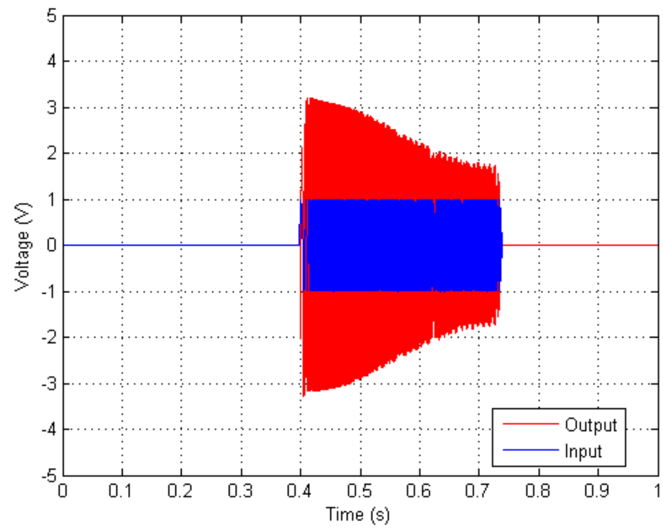



Figure 135. Time Spectrum –Haptic Board Voltage Measurement – No load – Sine Sweep, 6 dB Gain, 35 Volume

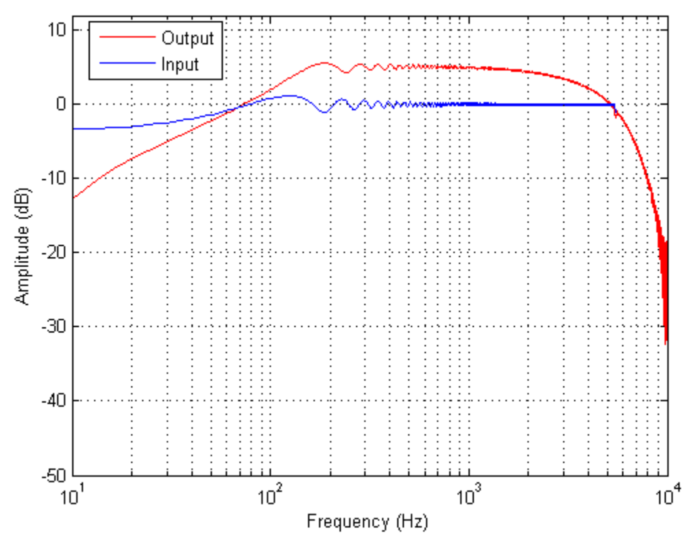



Figure 136. Frequency Spectrum –Haptic Board Voltage Measurement – No load – Sine Sweep, 6 dB Gain, 35 Volume

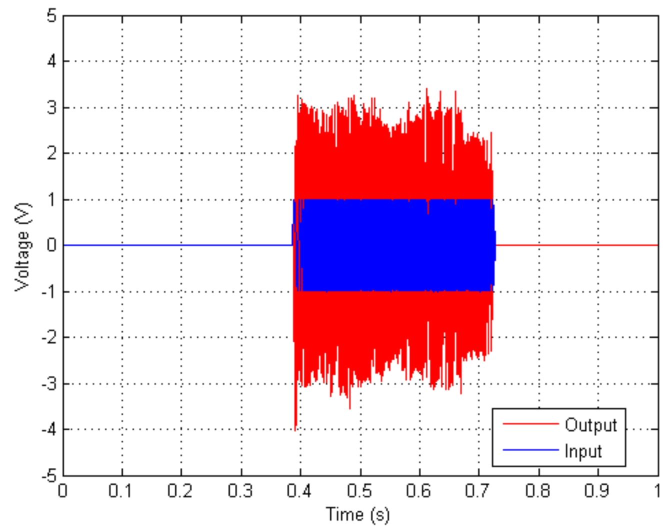



Figure 137. Time Spectrum –Haptic Board Voltage Measurement – No load – Sine Sweep, 6 dB Gain, 36 Volume

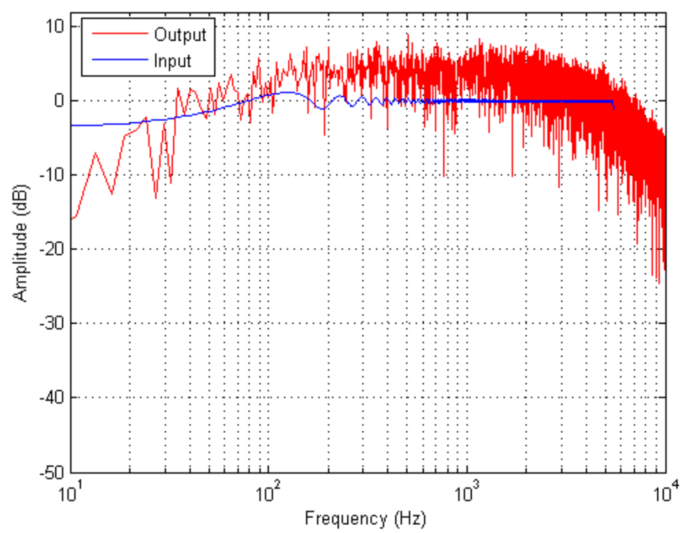



Figure 138. Frequency Spectrum –Haptic Board Voltage Measurement – No load – Sine Sweep, 6 dB Gain, 36 Volume

## Appendix J. Amplitude vs Time & Frequency Plots Exciter Voltage Measurement

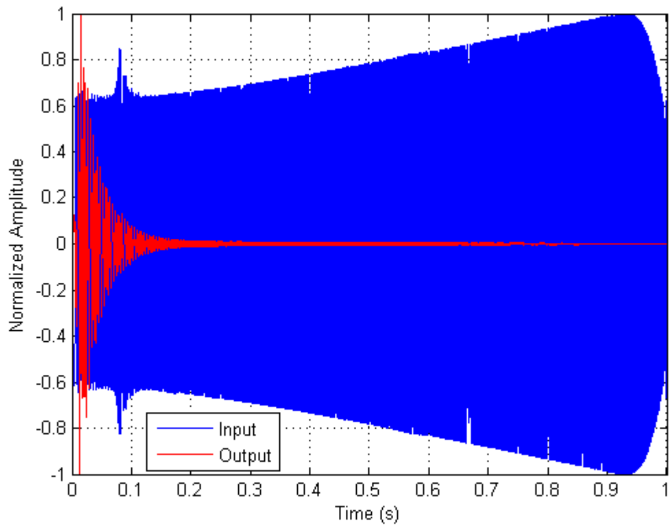



Figure 139. Exciter Voltage Output Time Spectrum – Sine Sweep, 4 V pk-pk

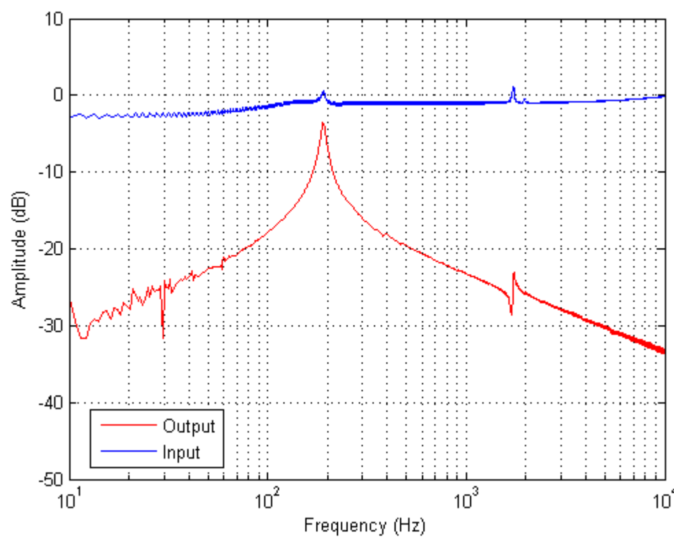



Figure 140. Exciter Voltage Output Frequency Spectrum – Sine Sweep, 4 V pk-pk

## Appendix K. Amplitude vs Time & Frequency Plots - Switch Pack Audio Response

### Appendix K.1. Pack 1

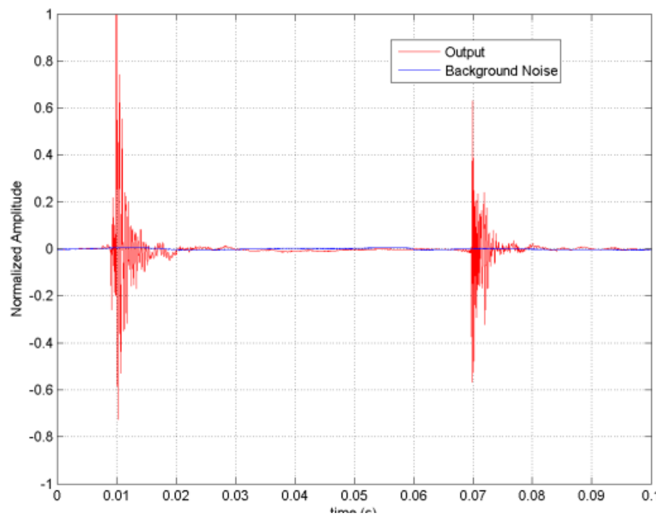



Figure 141. Audio Time Spectrum – Pack 1 Switch 1

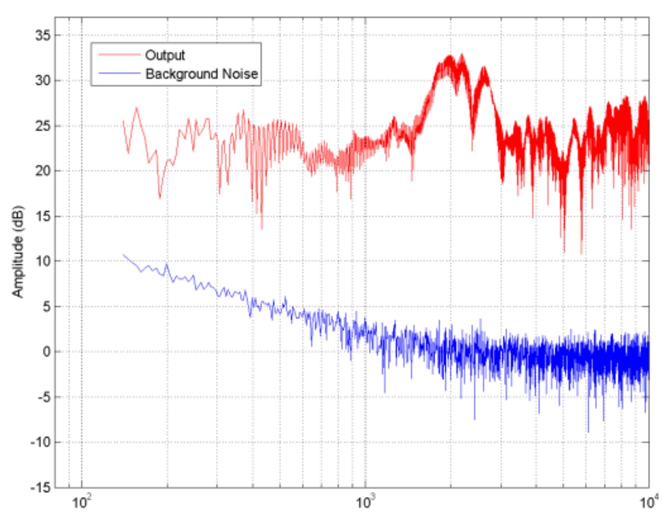



Figure 142. Audio Frequency Spectrum – Pack 1 Switch 1

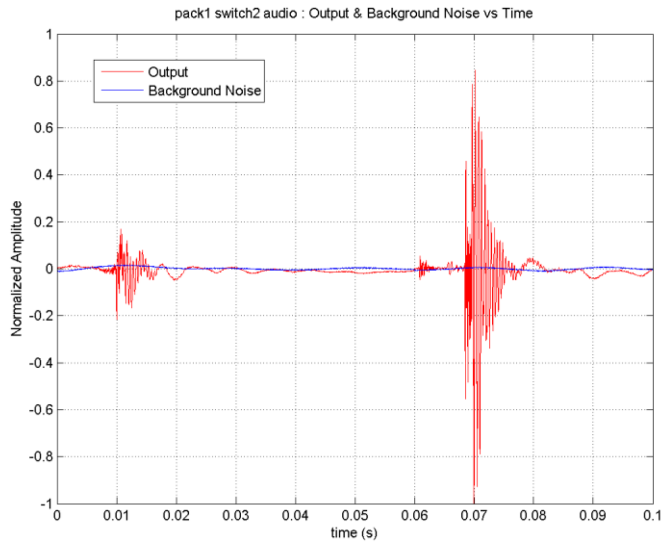



Figure 143. Audio Time Spectrum – Pack 1 Switch 2

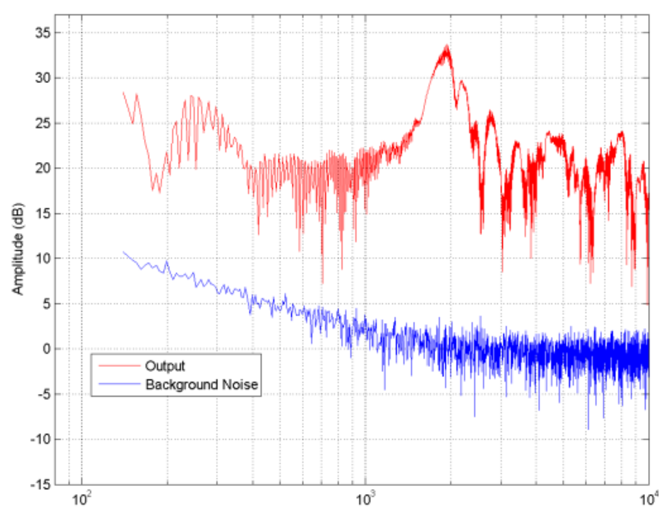



Figure 144. Audio Frequency Spectrum – Pack 1 Switch 2

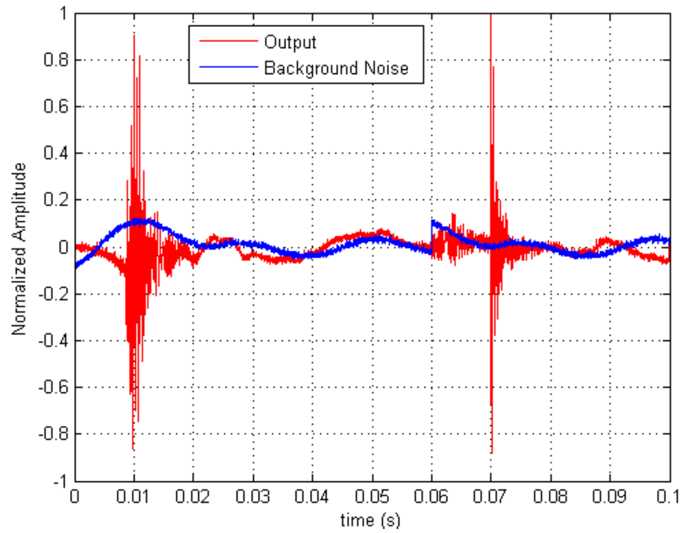



Figure 145. Audio Time Spectrum – Pack 1 Switch 3

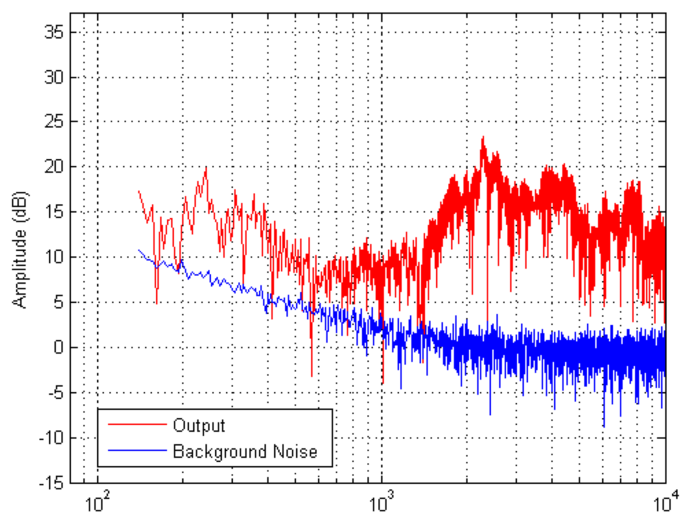



Figure 146. Audio Frequency Spectrum – Pack 1 Switch 3

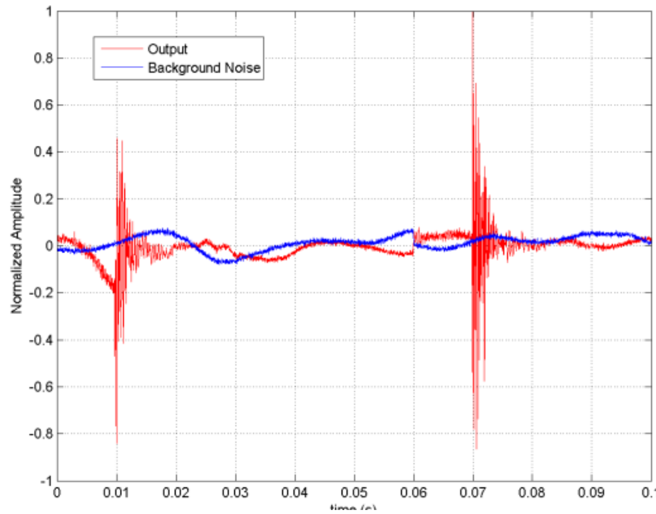



Figure 147. Audio Time Spectrum – Pack 1 Switch 4

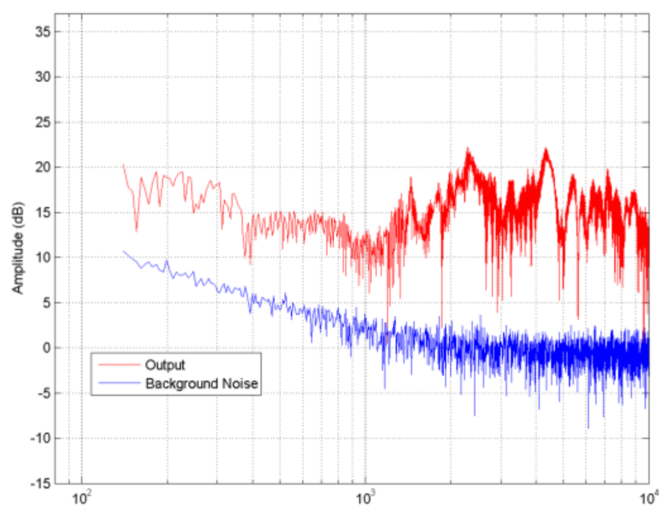



Figure 148. Audio Frequency Spectrum – Pack 1 Switch 4

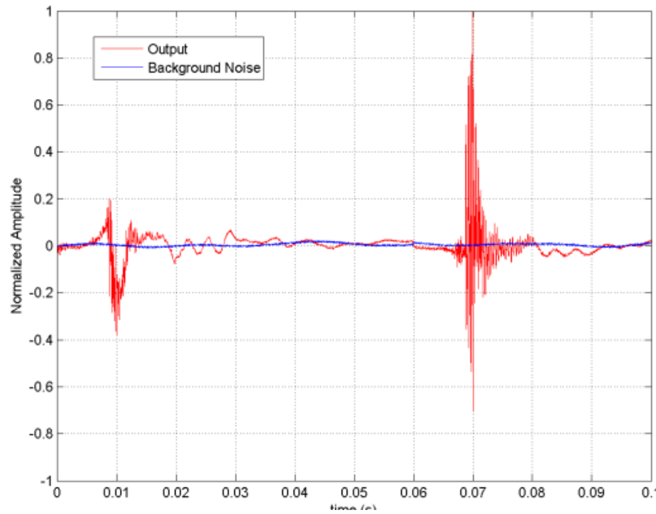



Figure 149. Audio Time Spectrum – Pack 1 Switch 5

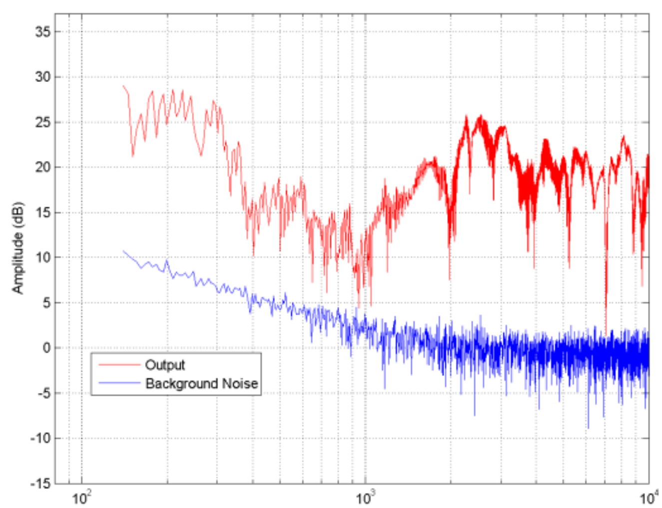



Figure 150. Audio Frequency Spectrum – Pack 1 Switch 5

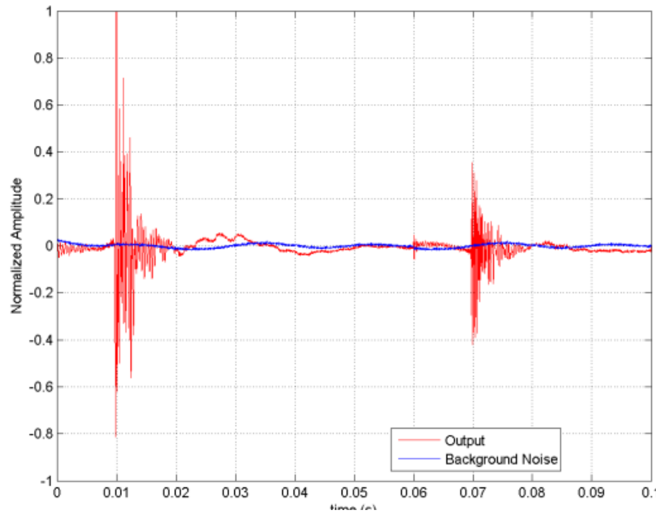



Figure 151. Audio Time Spectrum – Pack 1 Switch 6

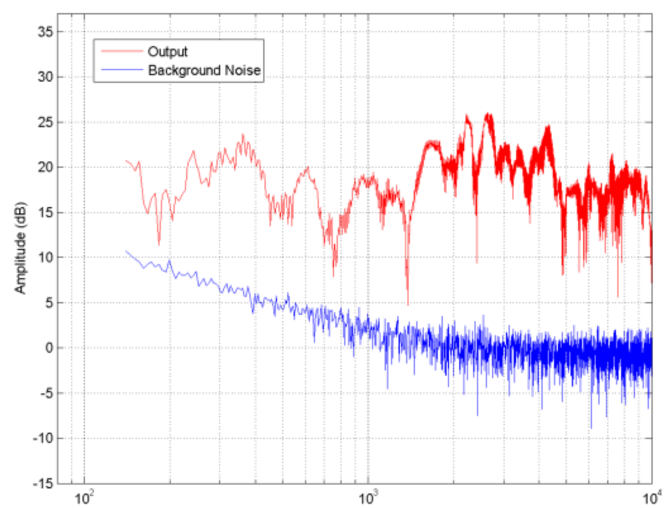



Figure 152. Audio Frequency Spectrum – Pack 1 Switch 6

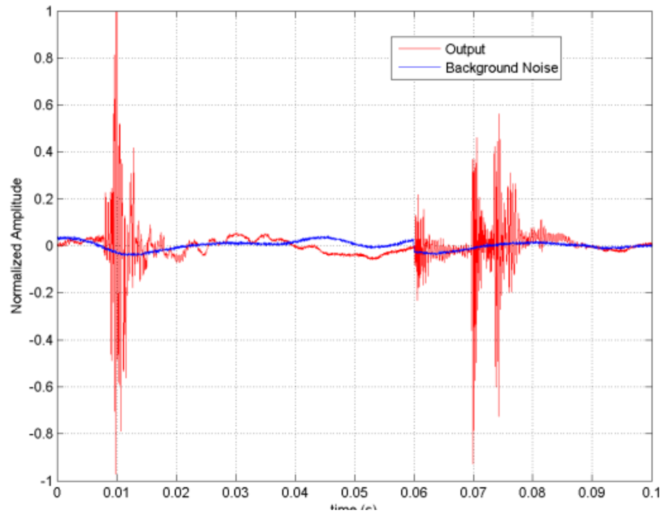



Figure 153. Audio Time Spectrum – Pack 1 Switch 7

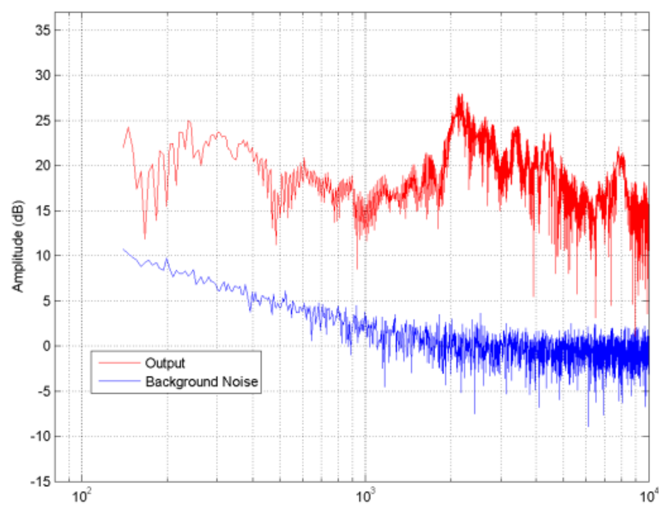



Figure 154. Audio Frequency Spectrum – Pack 1 Switch 7

## Appendix K.2. Pack 2

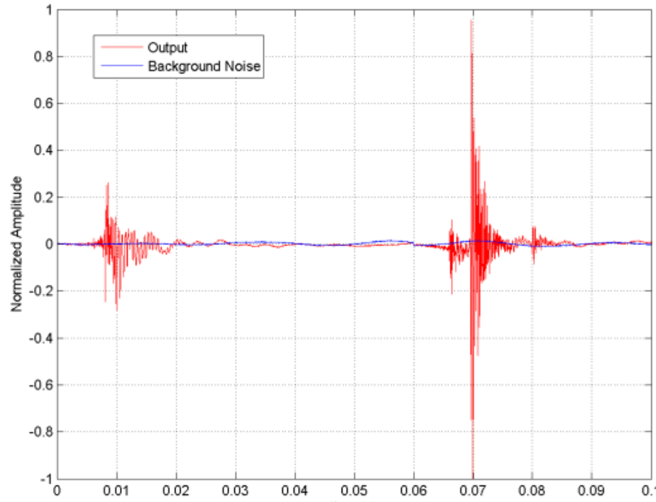



Figure 155. Audio Time Spectrum – Pack 2 Switch 1

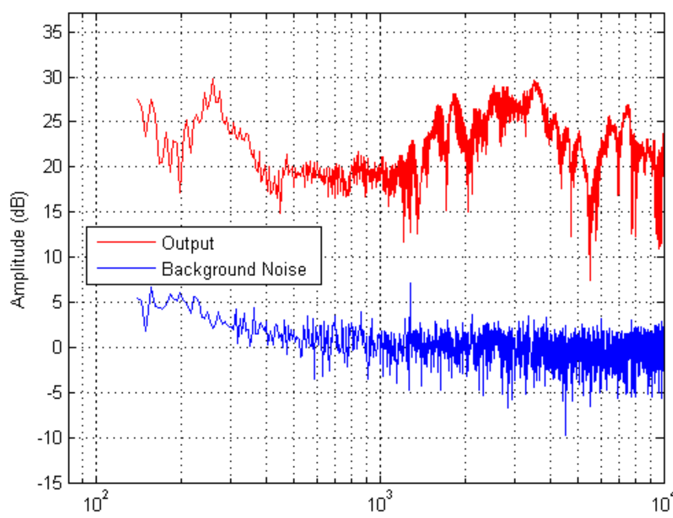



Figure 156. Audio Frequency Spectrum – Pack 2 Switch 1

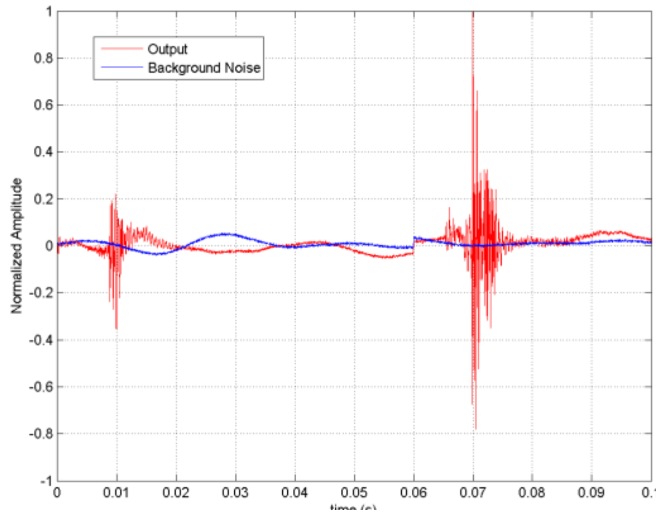



Figure 157. Audio Time Spectrum – Pack 2 Switch 2

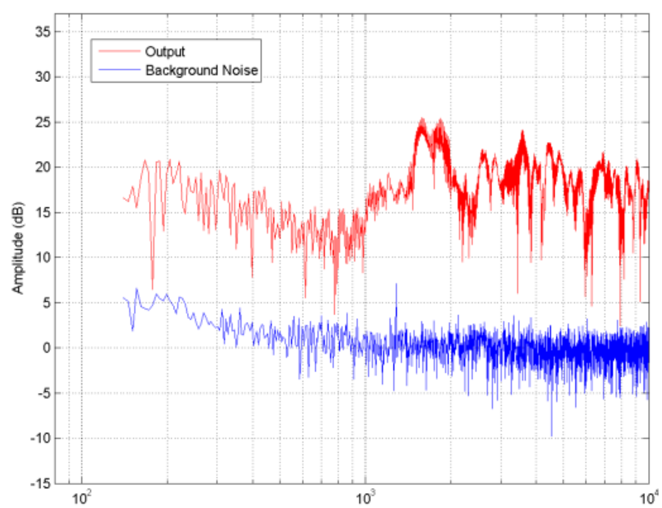



Figure 158. Audio Frequency Spectrum – Pack 2 Switch 2

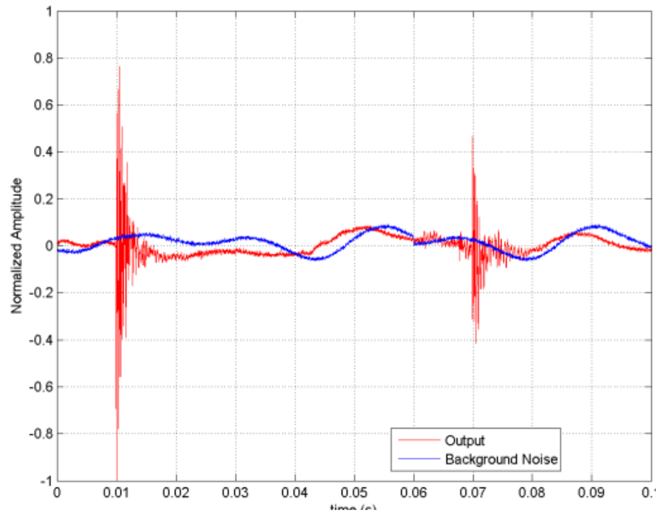



Figure 159. Audio Time Spectrum – Pack 2 Switch 3

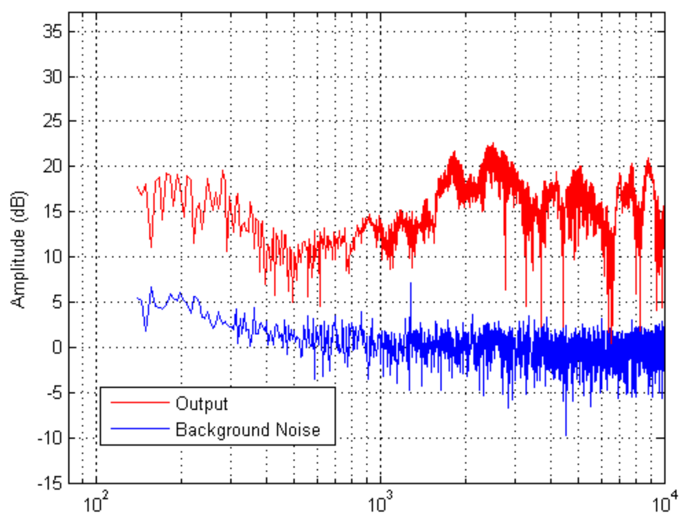



Figure 160. Audio Frequency Spectrum – Pack 2 Switch 3

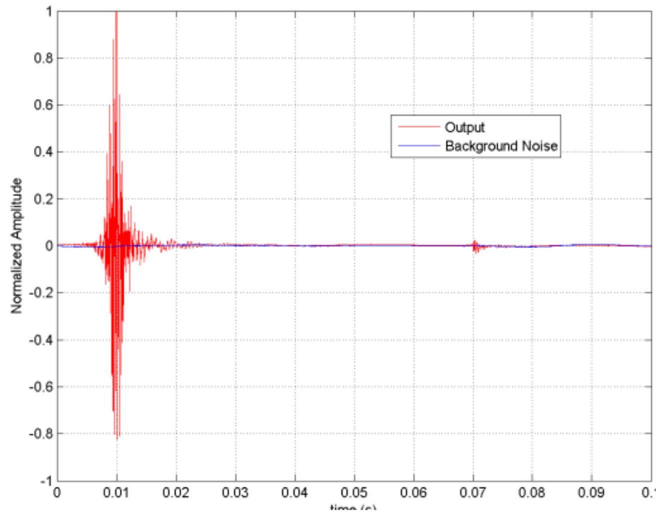



Figure 161. Audio Time Spectrum – Pack 2 Switch 4

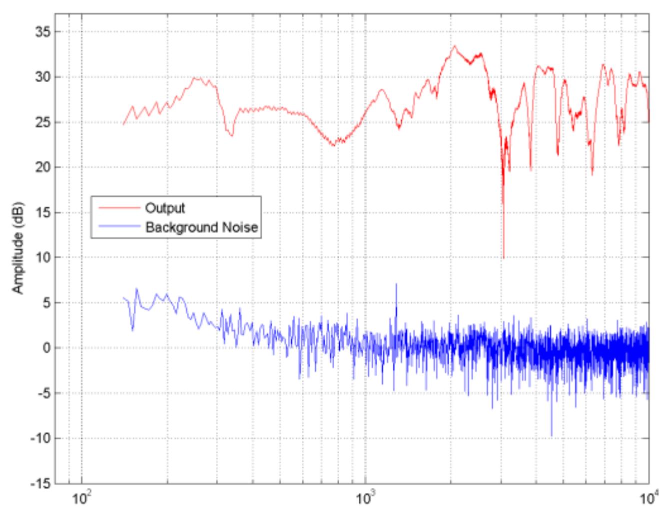



Figure 162. Audio Frequency Spectrum – Pack 2 Switch 4

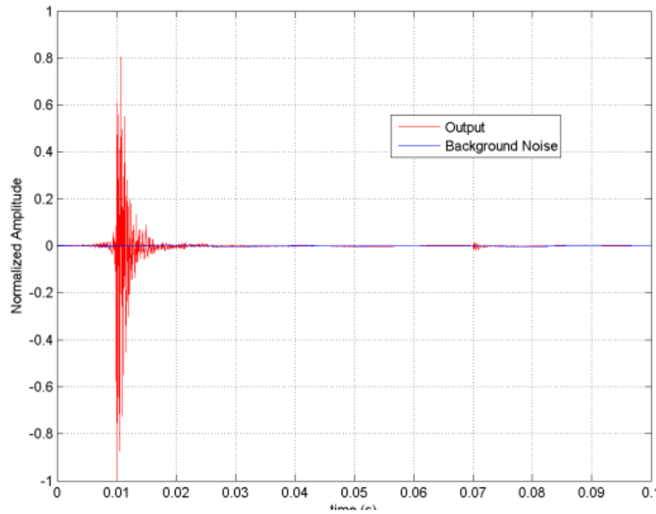



Figure 163. Audio Time Spectrum – Pack 2

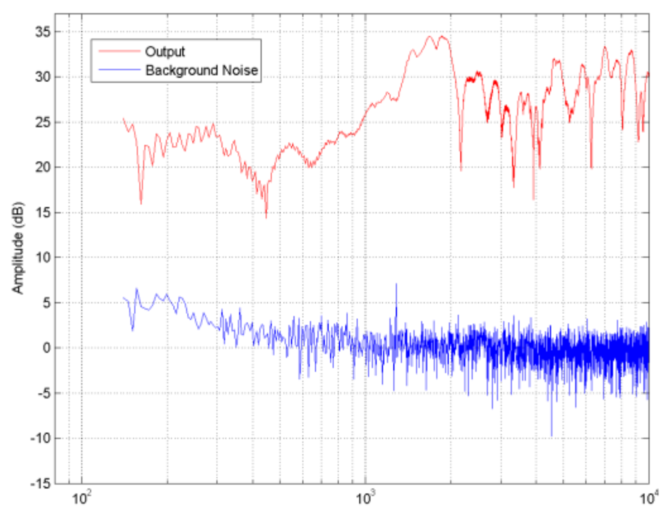



Figure 164. Audio Frequency Spectrum – Pack 2 Switch 5

## Appendix K.3. Pack 3

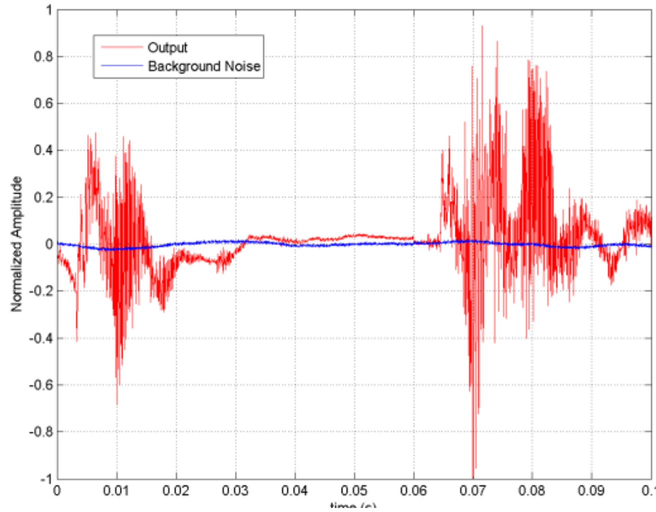



Figure 165. Audio Time Spectrum – Pack 3 Switch 1

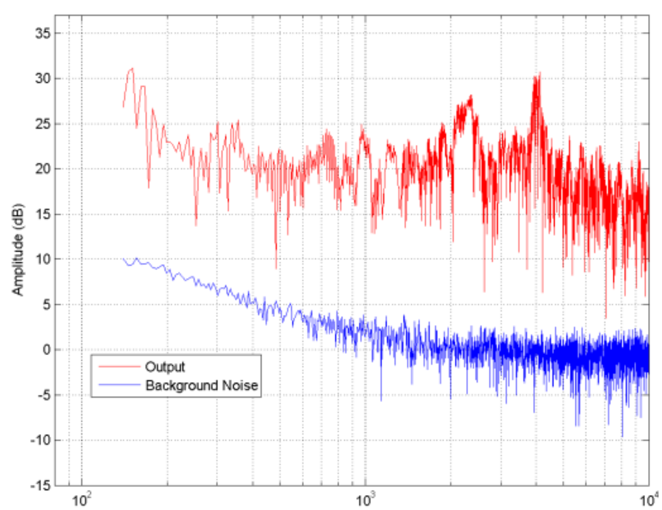



Figure 166. Audio Frequency Spectrum – Pack 3 Switch 1

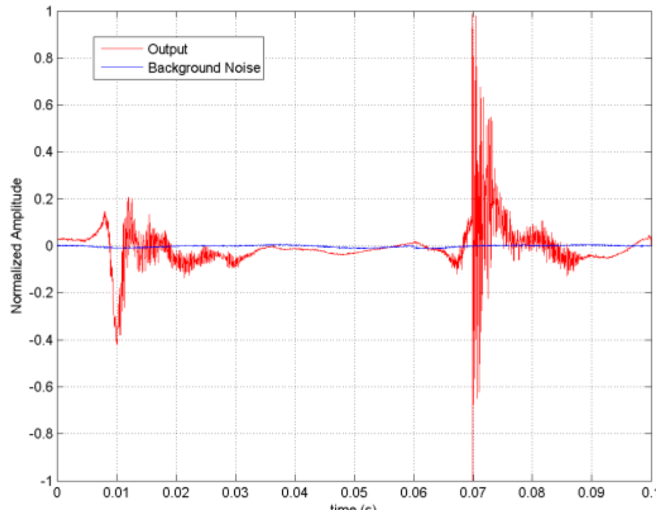



Figure 167. Audio Time Spectrum – Pack 3 Switch 2

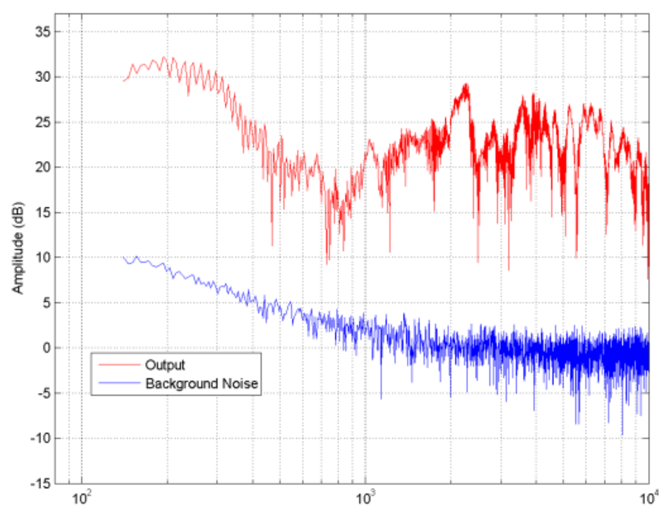



Figure 168. Audio Frequency Spectrum – Pack 3 Switch 2

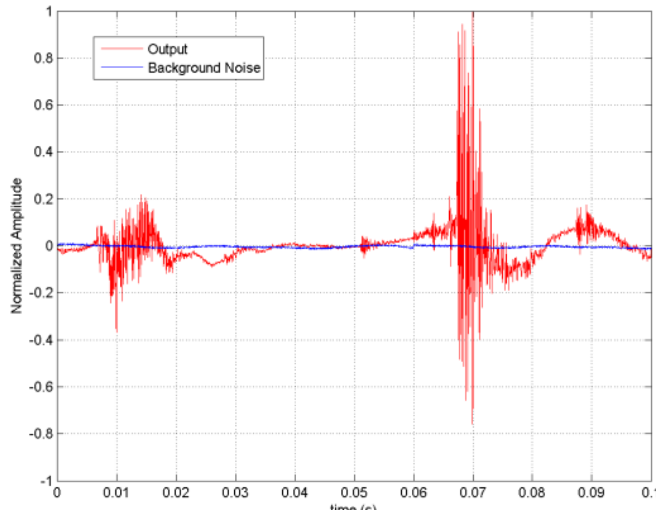



Figure 169. Audio Time Spectrum – Pack 3 Switch 3

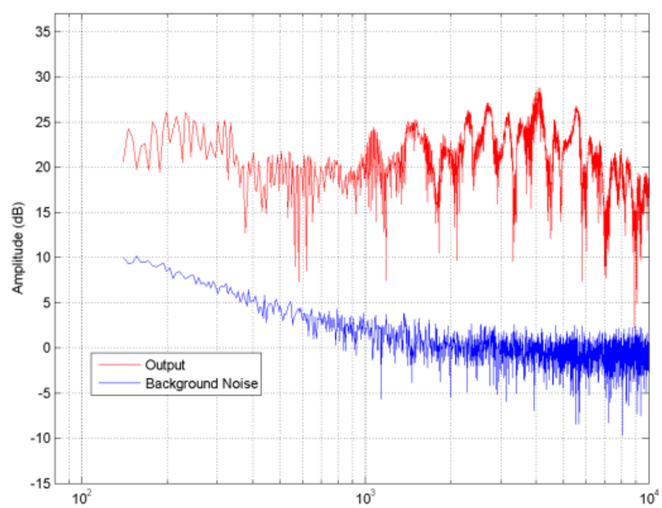



Figure 170. Audio Frequency Spectrum – Pack 3 Switch 3

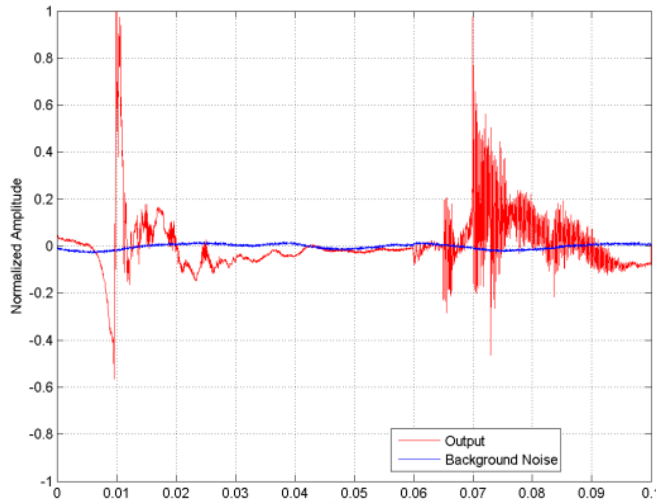



Figure 171. Audio Time Spectrum – Pack 3 Switch 4

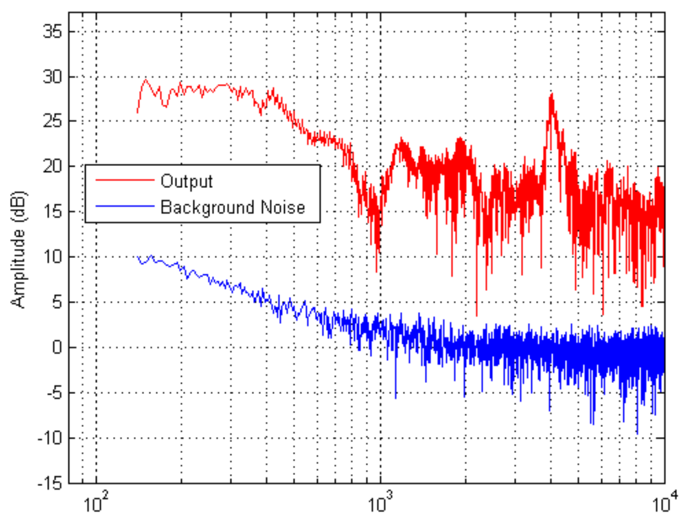



Figure 172. Audio Frequency Spectrum – Pack 3 Switch 4

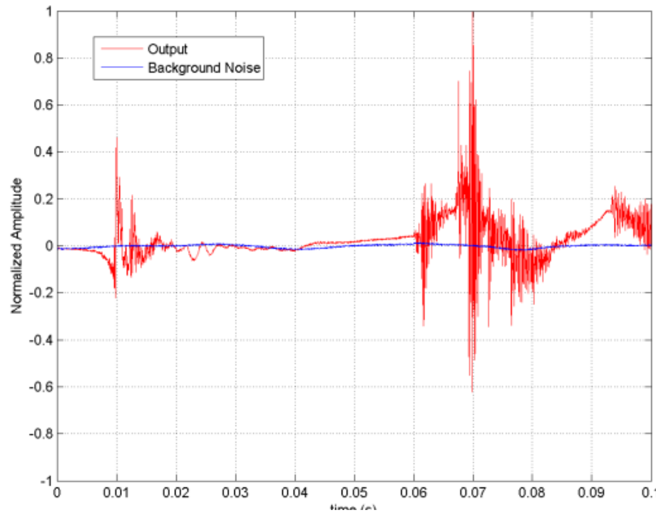



Figure 173. Audio Time Spectrum – Pack 3 Switch 5

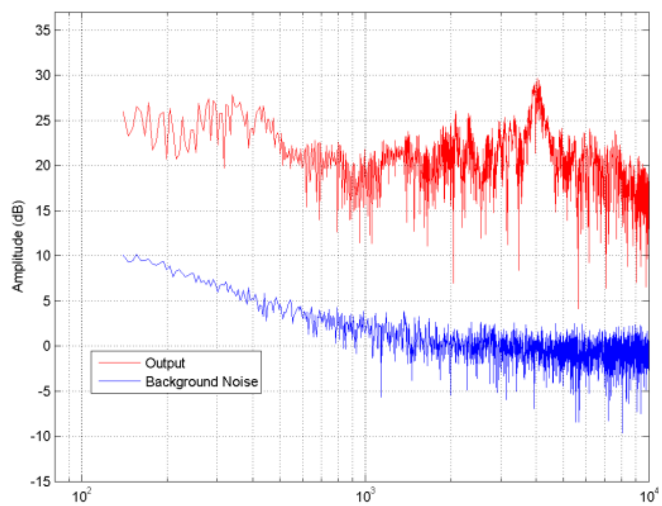



Figure 174. Audio Frequency Spectrum – Pack 3 Switch 5

## Appendix K.4. Pack 4

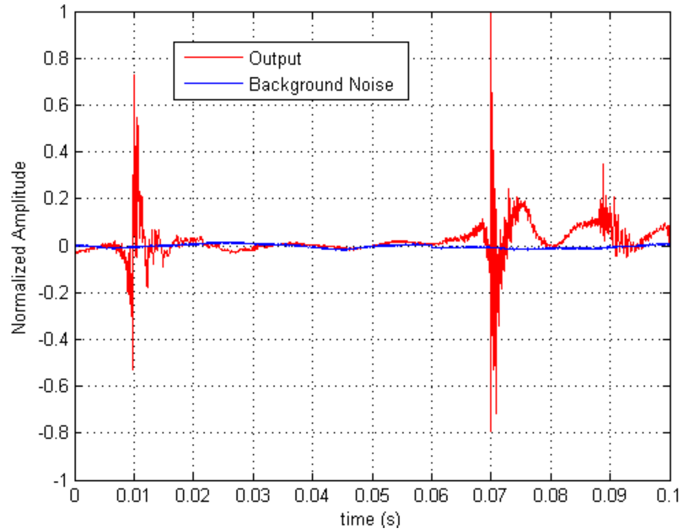



Figure 175. Audio Time Spectrum – Pack 4 Switch 1

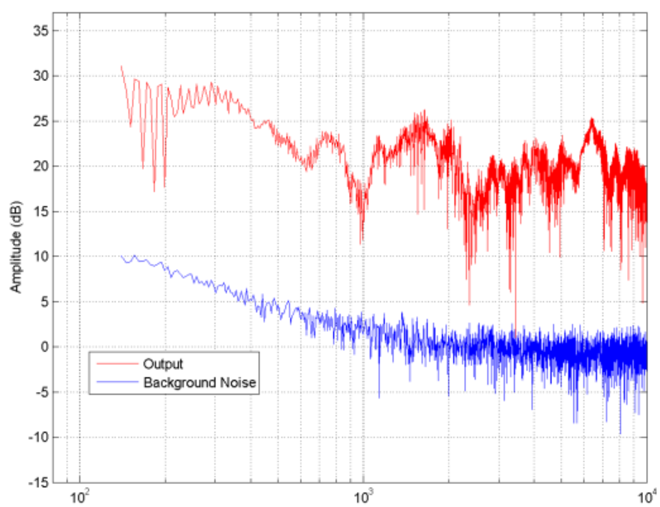



Figure 176. Audio Frequency Spectrum – Pack 4 Switch 1

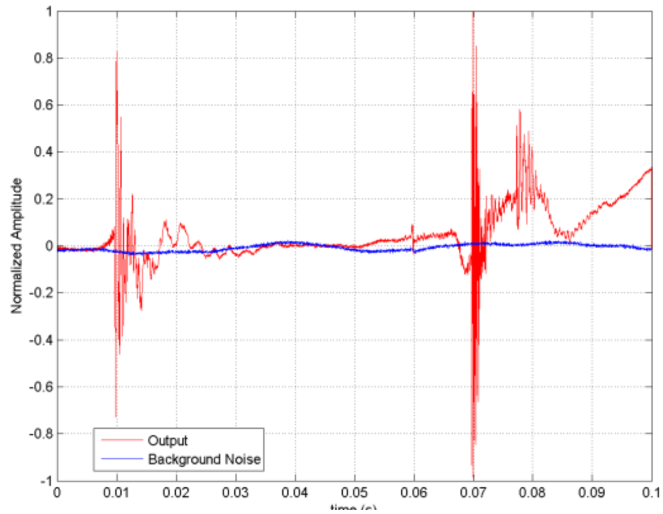



Figure 177. Audio Time Spectrum – Pack 4 Switch 2

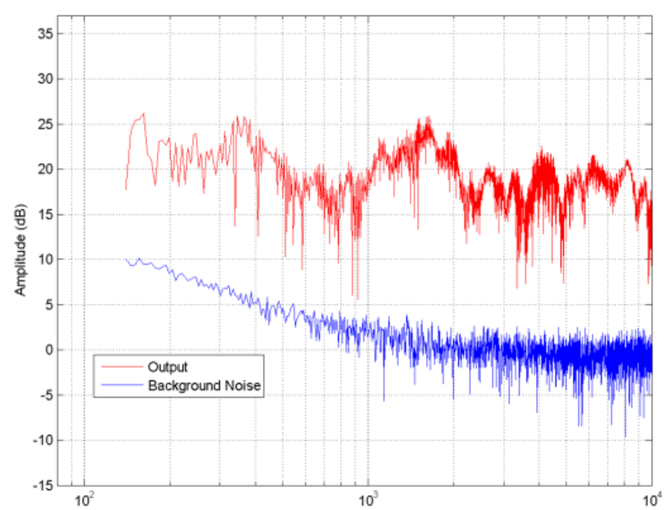



Figure 178. Audio Frequency Spectrum – Pack 4 Switch 2

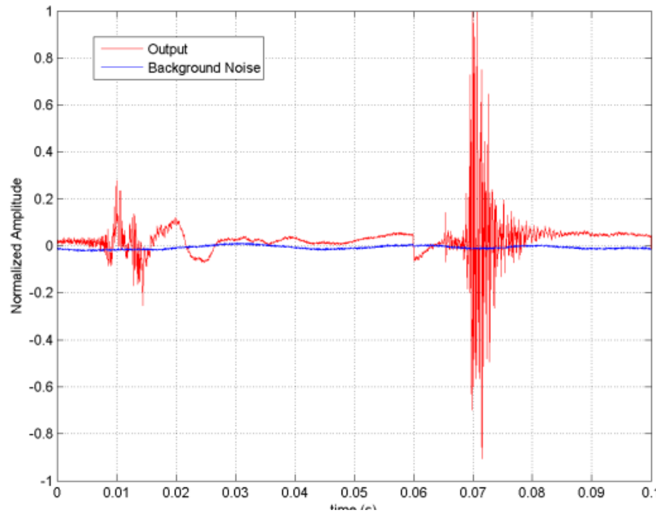



Figure 179. Audio Time Spectrum – Pack 4 Switch 3

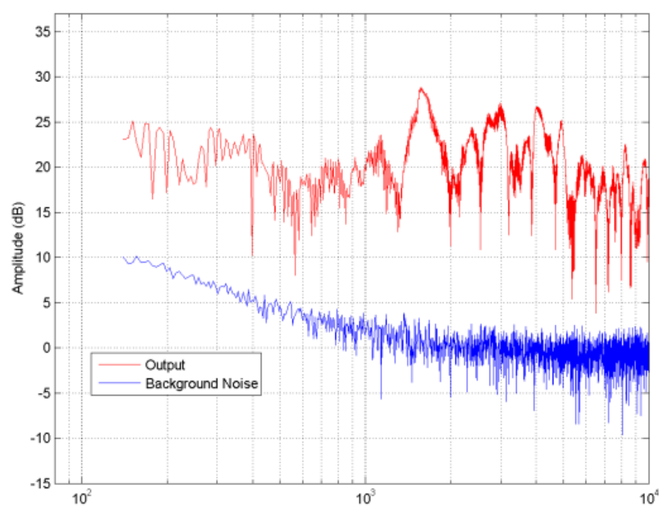



Figure 180. Audio Frequency Spectrum – Pack 4 Switch 3

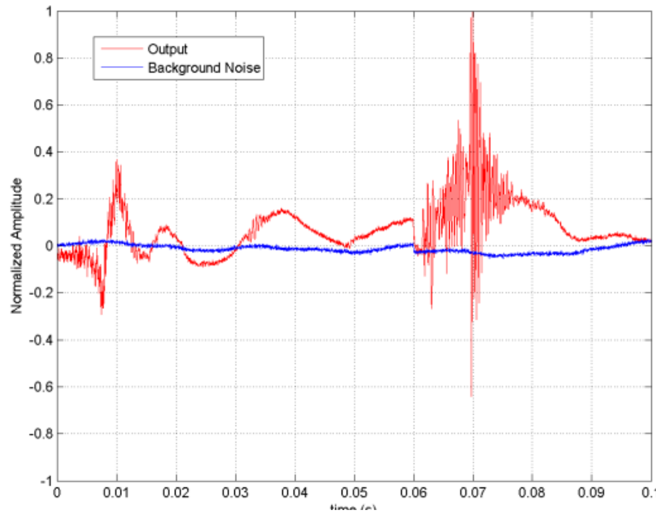



Figure 181. Audio Time Spectrum – Pack 4 Switch 4

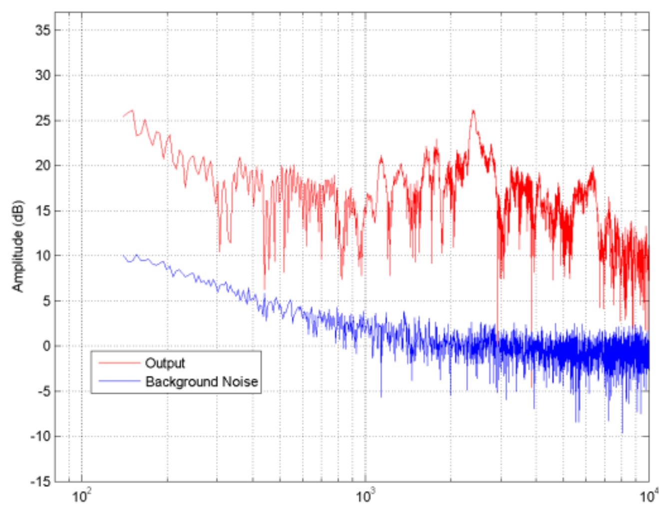



Figure 182. Audio Frequency Spectrum – Pack 4 Switch 4

## Appendix K.5. Pack 5

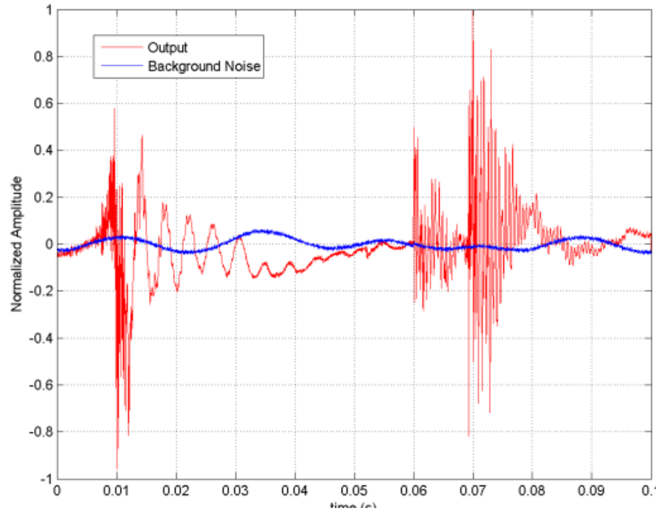



Figure 183. Audio Time Spectrum – Pack 5 Switch 1

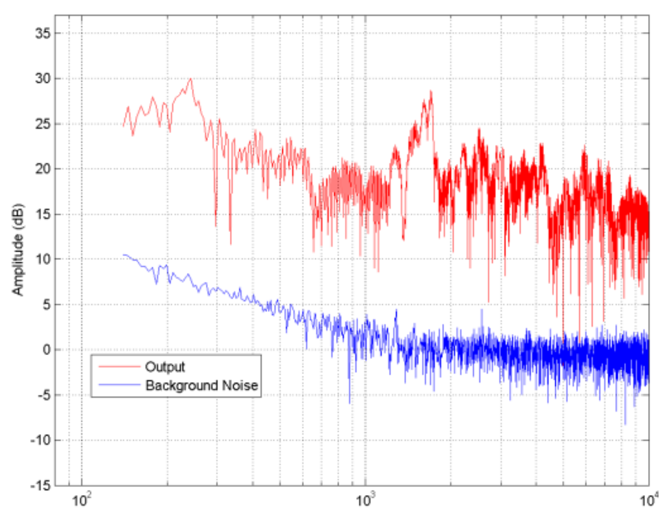



Figure 184. Audio Frequency Spectrum – Pack 5 Switch 1

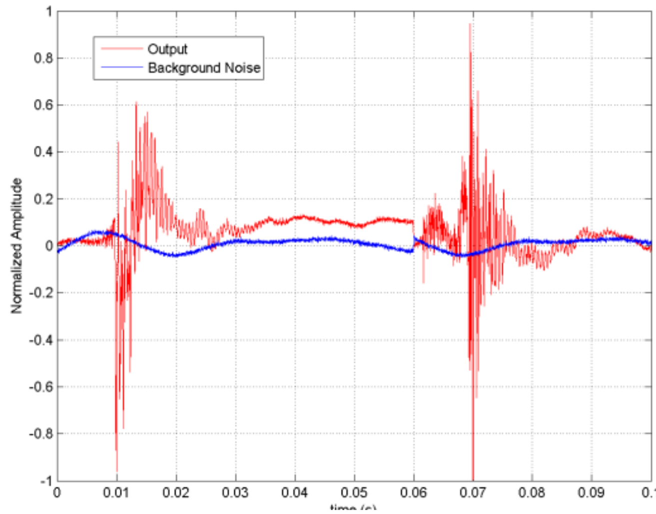



Figure 185. Audio Time Spectrum – Pack 5 Switch 2

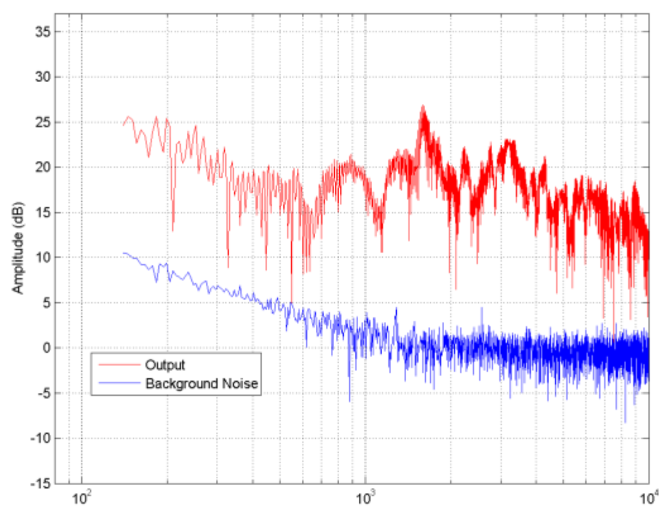



Figure 186. Audio Frequency Spectrum – Pack 5 Switch 2

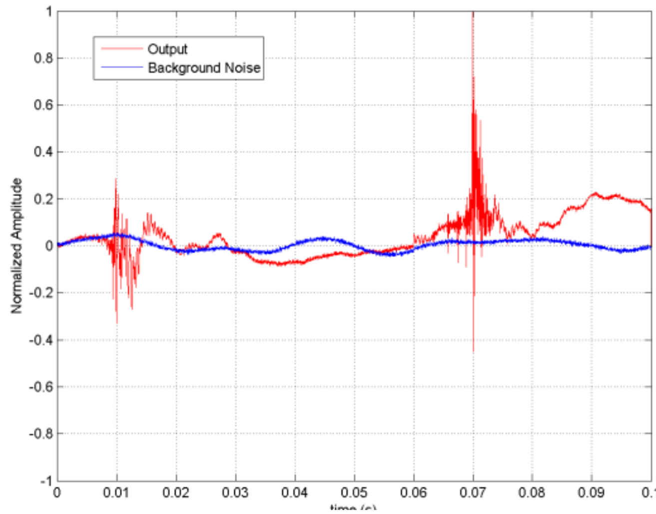



Figure 187. Audio Time Spectrum – Pack 5 Switch 3

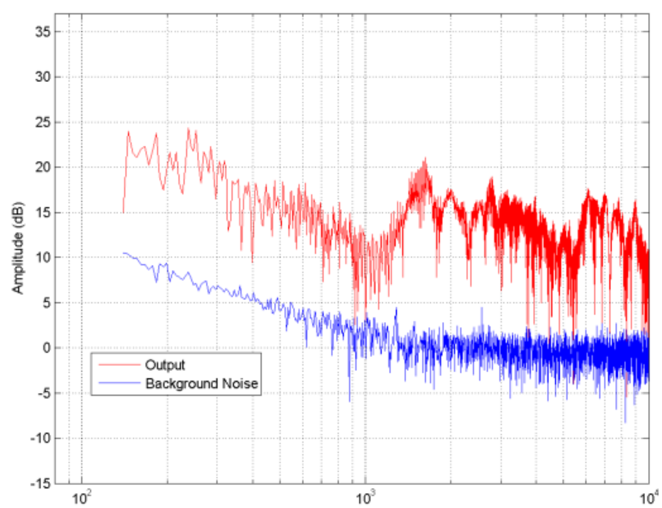



Figure 188. Audio Frequency Spectrum – Pack 5 Switch 3

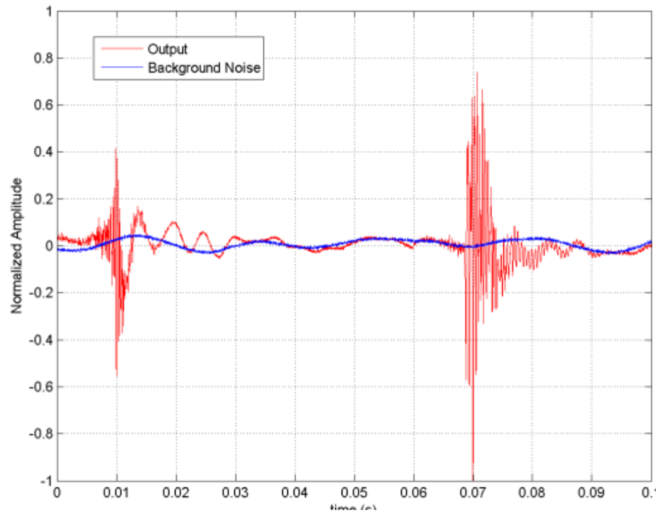



Figure 189. Audio Time Spectrum – Pack 5 Switch 4

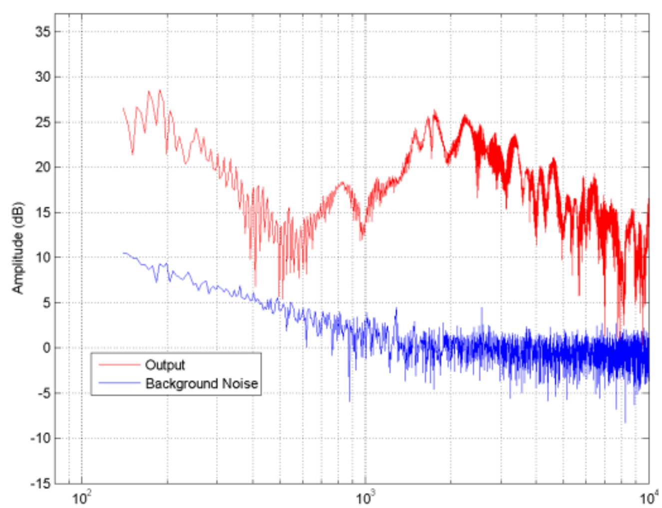



Figure 190. Audio Frequency Spectrum – Pack 5 Switch 4

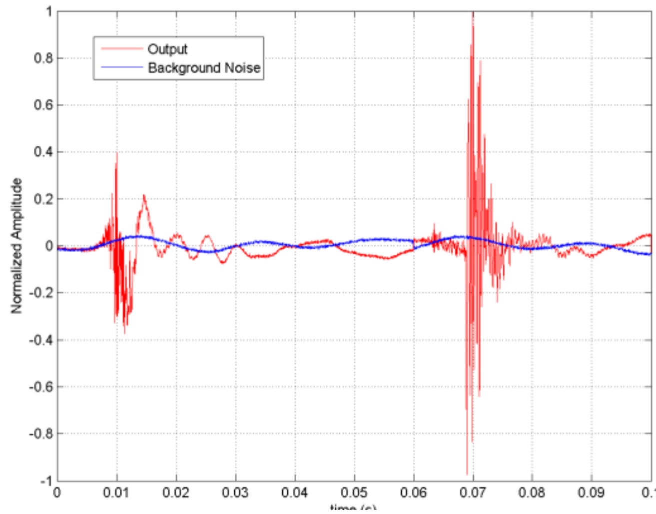



Figure 191. Audio Time Spectrum – Pack 5 Switch 5

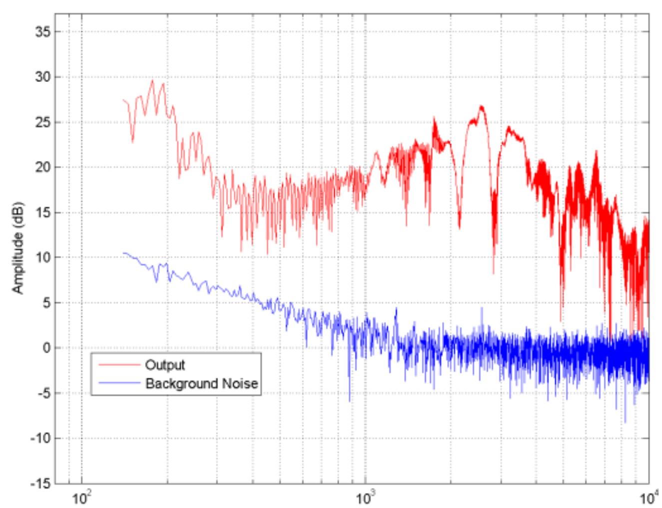



Figure 192. Audio Frequency Spectrum – Pack 5 Switch 5

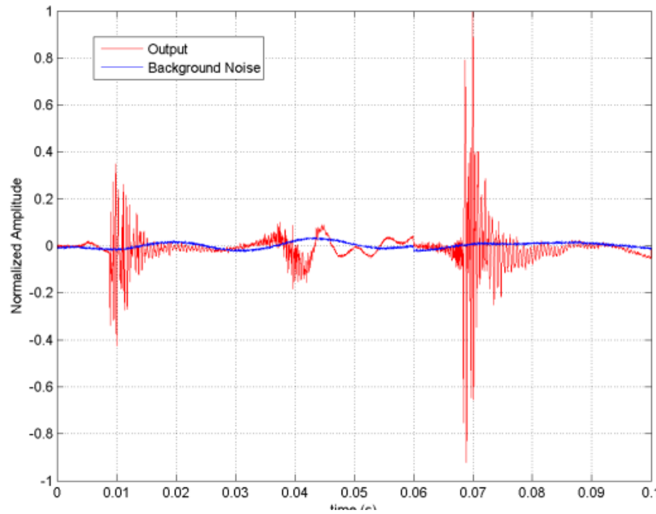



Figure 193. Audio Time Spectrum – Pack 5 Switch 6

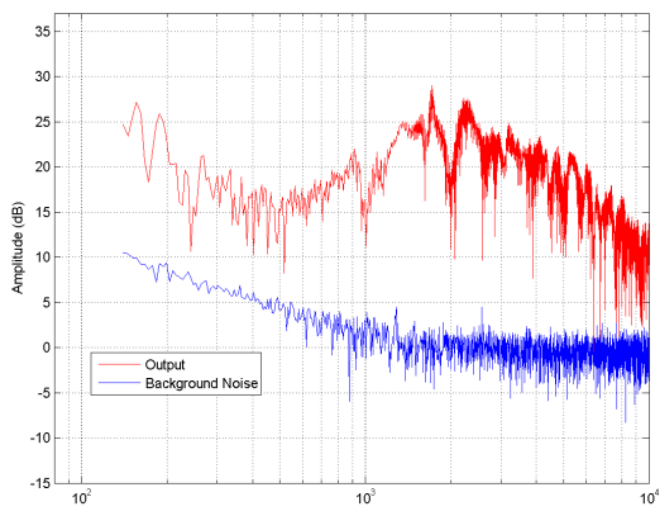



Figure 194. Audio Frequency Spectrum – Pack 5 Switch 6

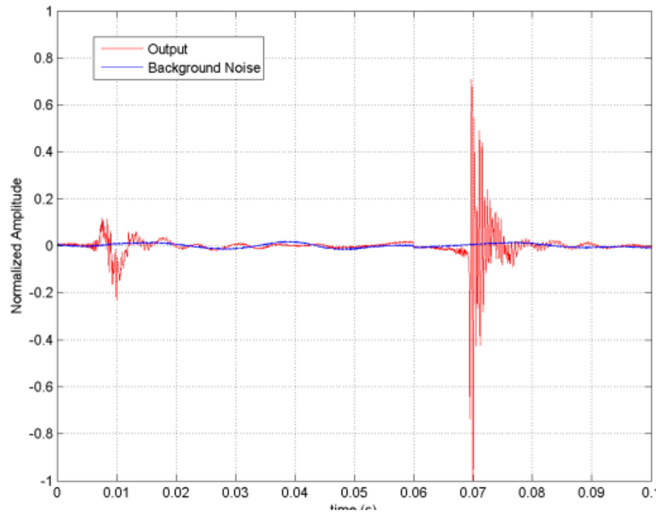



Figure 195. Audio Time Spectrum – Pack 5 Switch 7

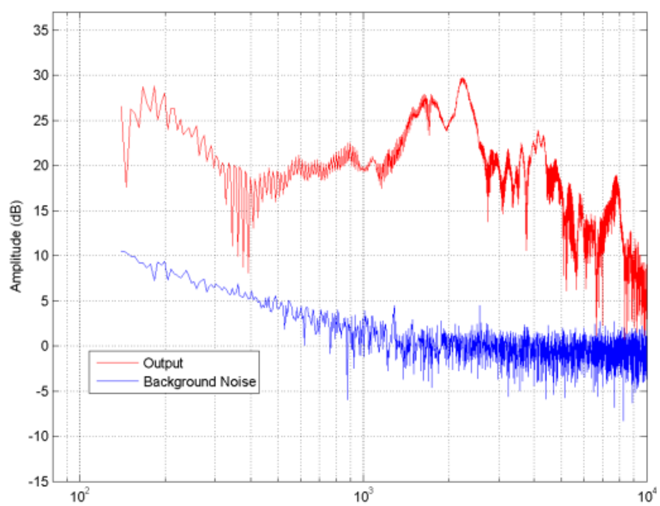



Figure 196. Audio Frequency Spectrum – Pack 5 Switch 7

## Appendix K.6. Pack 6

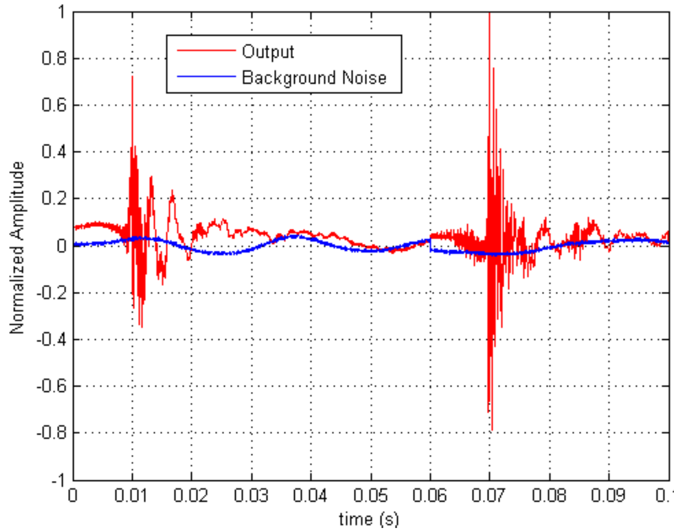



Figure 197. Audio Time Spectrum – Pack 6 Switch 1

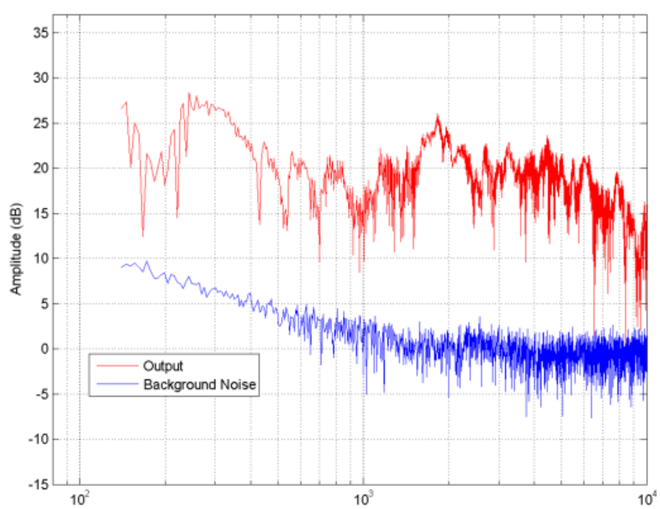



Figure 198. Audio Frequency Spectrum – Pack 6 Switch 1

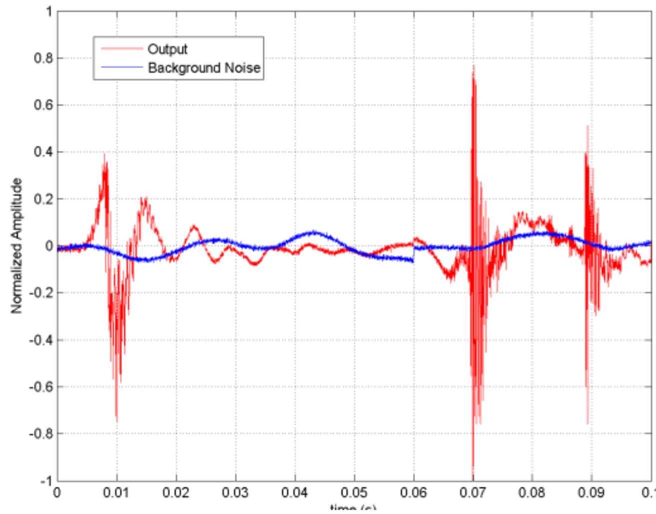



Figure 199. Audio Time Spectrum – Pack 6 Switch 2

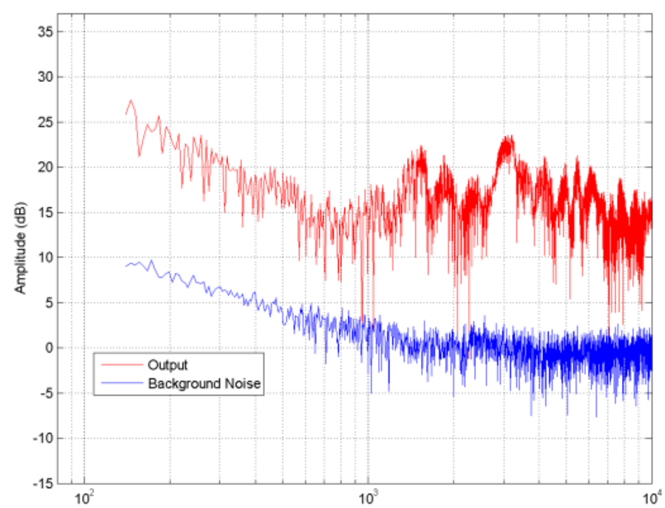



Figure 200. Audio Frequency Spectrum – Pack 6 Switch 2

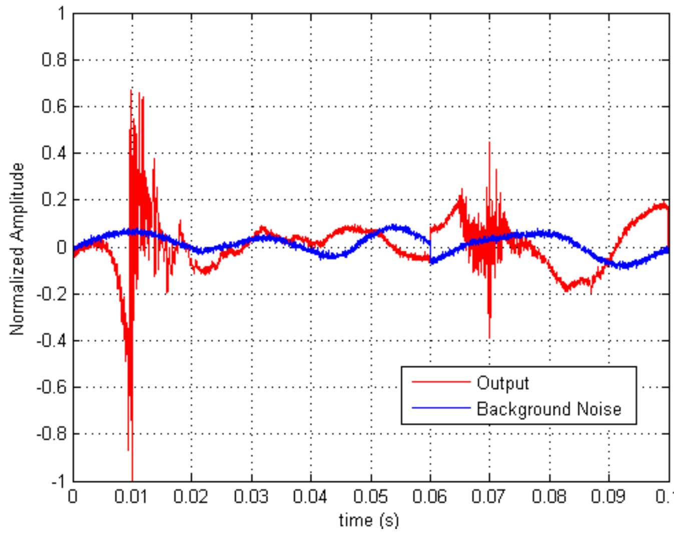



Figure 201. Audio Time Spectrum – Pack 6 Switch 3

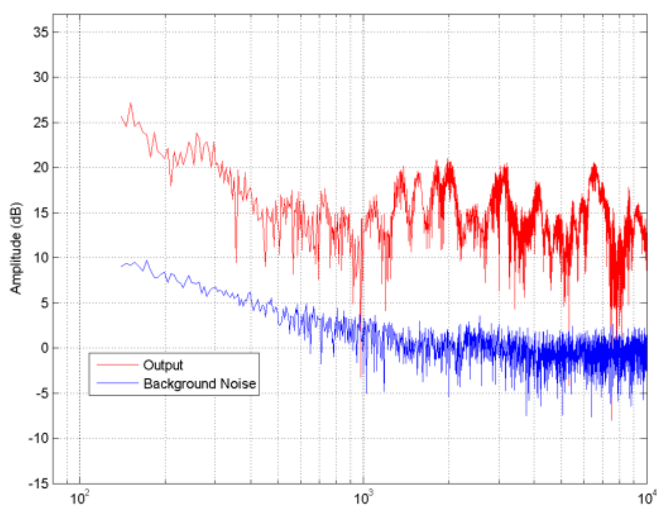



Figure 202. Audio Frequency Spectrum – Pack 6 Switch 3

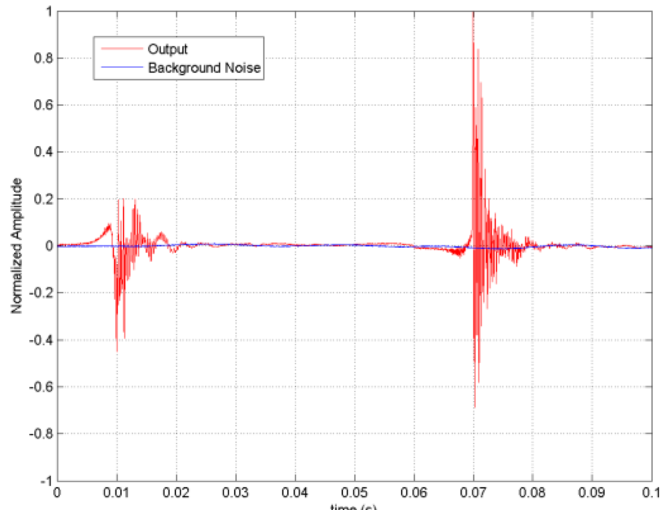



Figure 203. Audio Time Spectrum – Pack 6 Switch 4

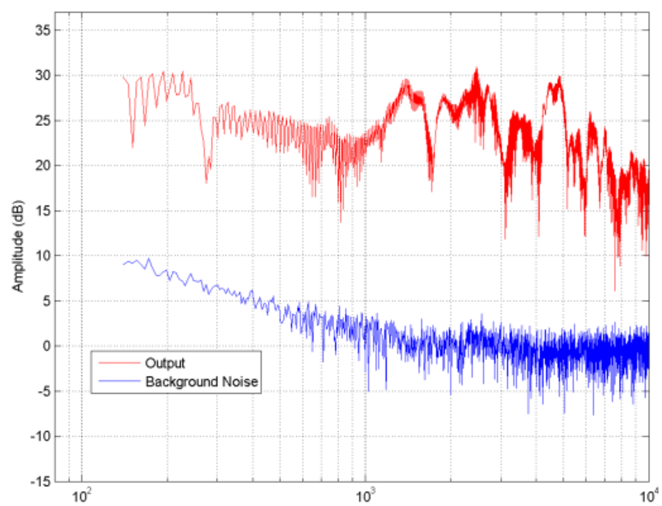



Figure 204. Audio Frequency Spectrum – Pack 6 Switch 4

**Appendix K.7. Pack 7**

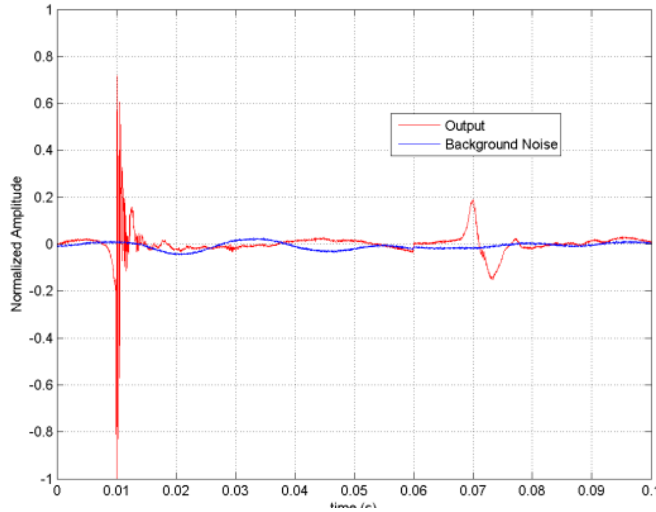



Figure 205. Audio Time Spectrum – Pack 7 Switch 1

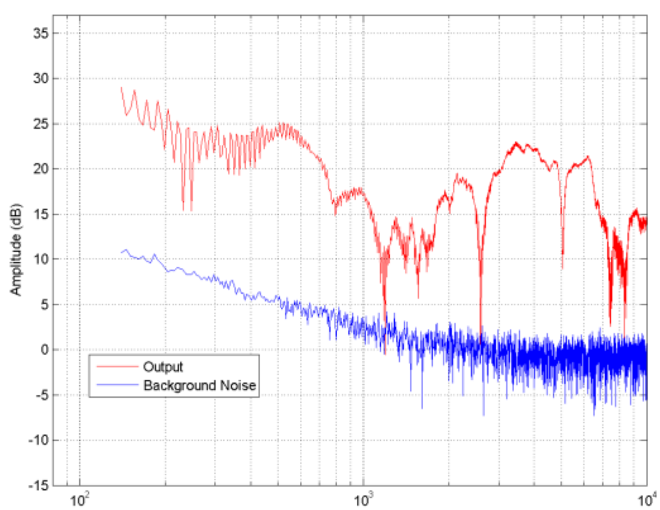



Figure 206. Audio Frequency Spectrum – Pack 7 Switch 1

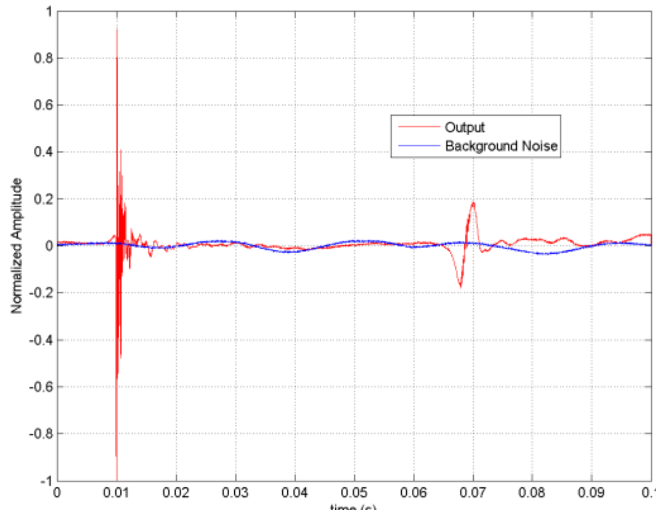



Figure 207. Audio Time Spectrum – Pack 7 Switch 2

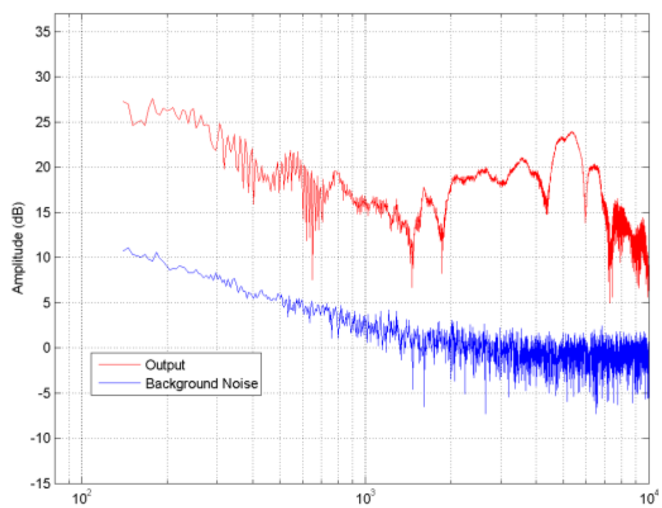



Figure 208. Audio Frequency Spectrum – Pack 7 Switch 2

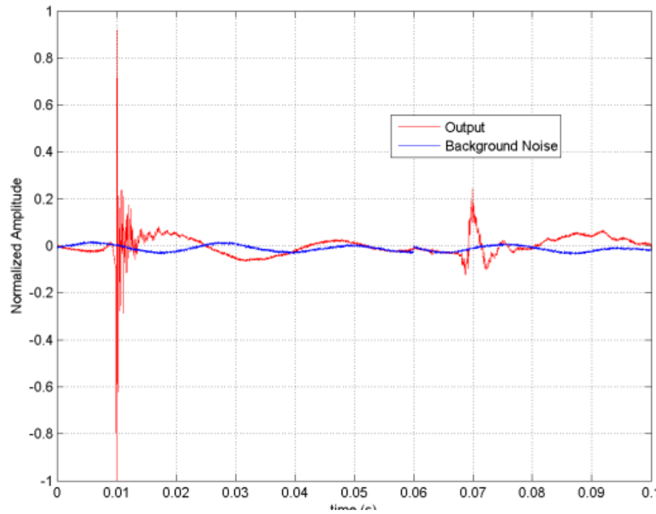



Figure 209. Audio Time Spectrum – Pack 7 Switch 3

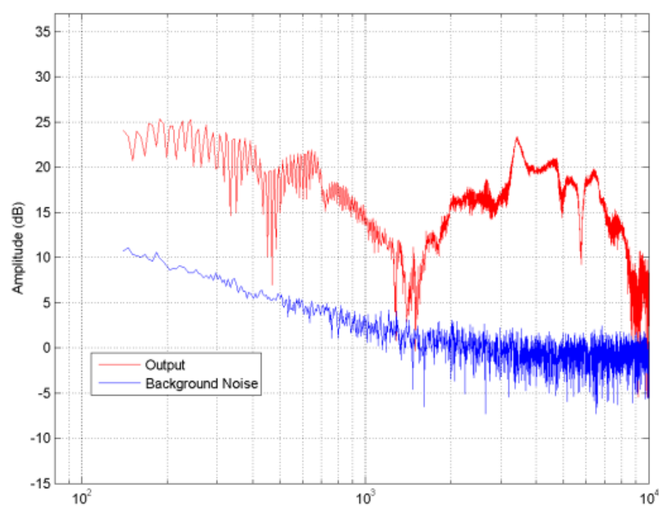



Figure 210. Audio Frequency Spectrum – Pack 7 Switch 3

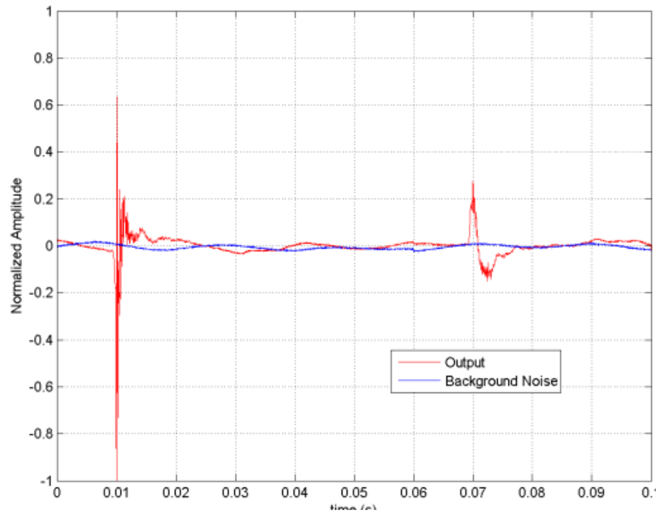



Figure 211. Audio Time Spectrum – Pack 7 Switch 4

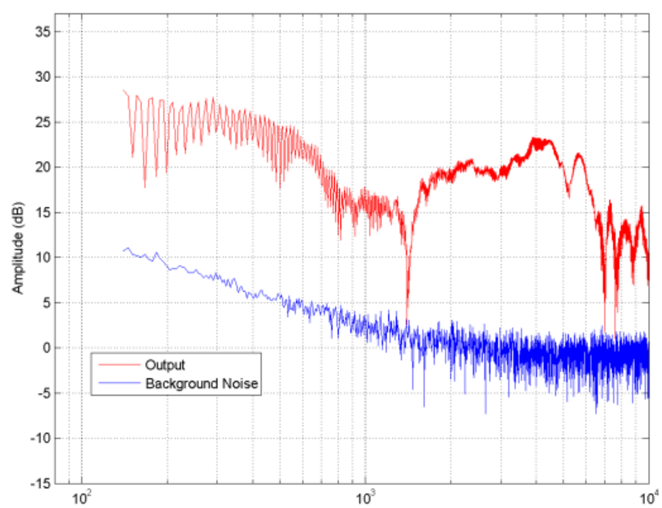



Figure 212. Audio Frequency Spectrum – Pack 7 Switch 4

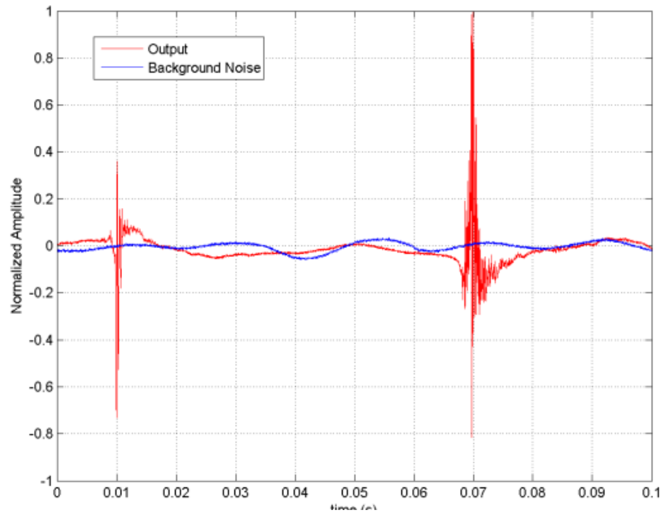



Figure 213. Audio Time Spectrum – Pack 7 Switch 5

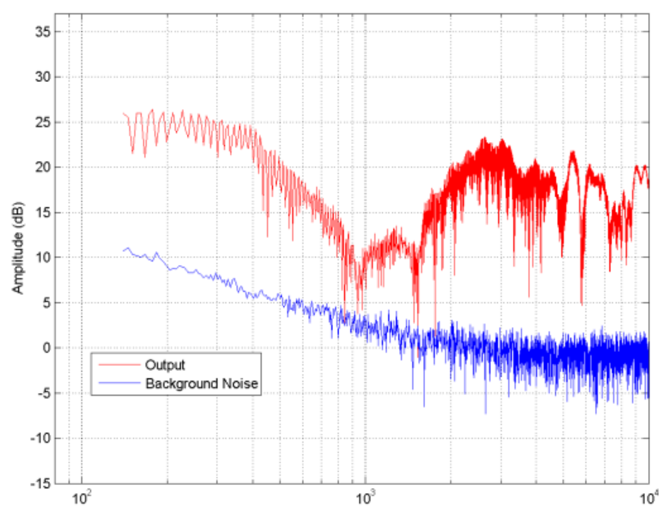



Figure 214. Audio Frequency Spectrum – Pack 7 Switch 5

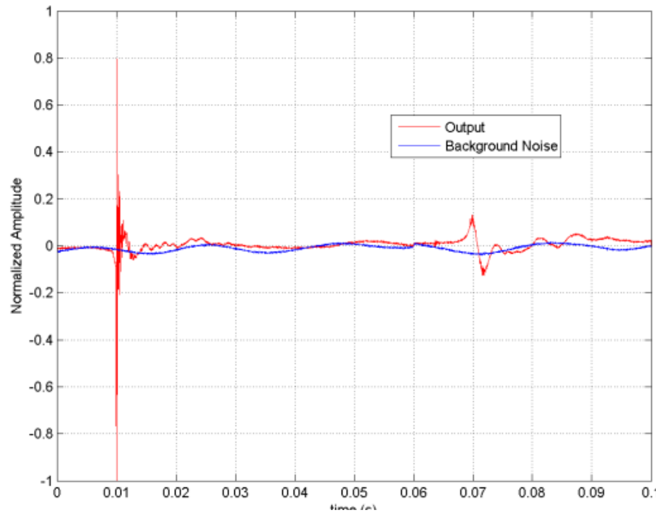



Figure 215. Audio Time Spectrum – Pack 7 Switch 6

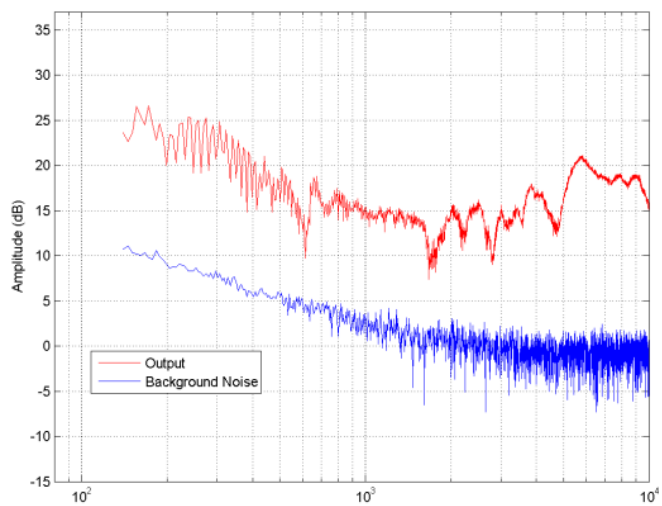



Figure 216. Audio Frequency Spectrum – Pack 7 Switch 6

**Appendix K.8. Pack 8**

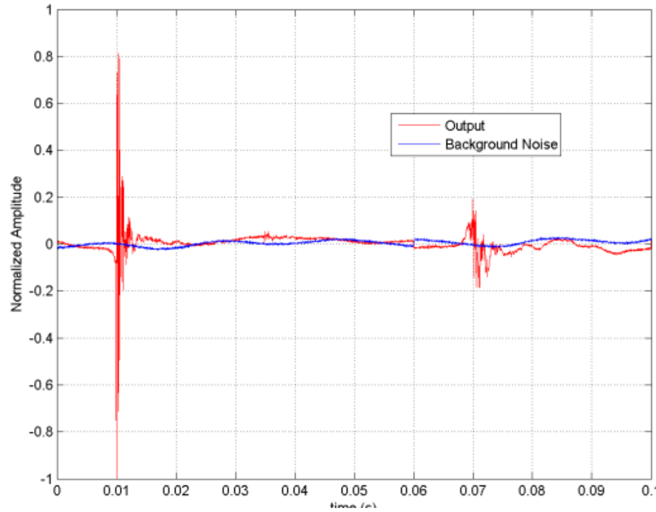



Figure 217. Audio Time Spectrum – Pack 8 Switch 1

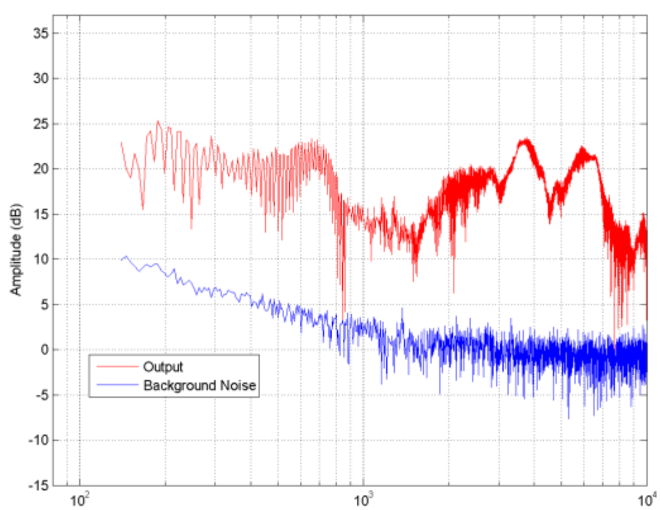



Figure 218. Audio Frequency Spectrum – Pack 8 Switch 1

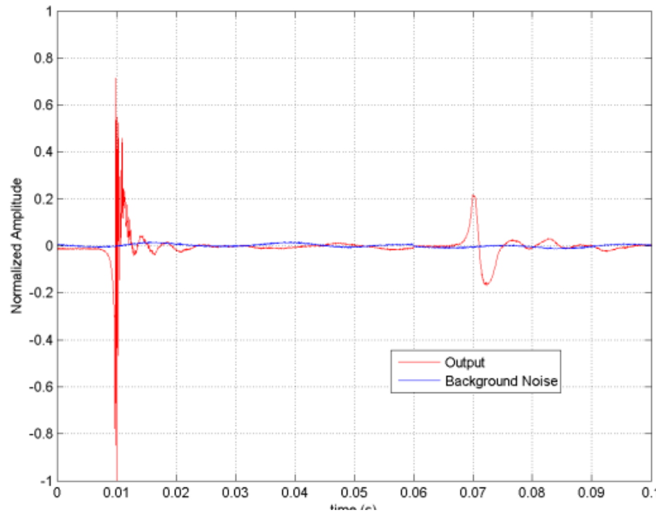



Figure 219. Audio Time Spectrum – Pack 8 Switch 2

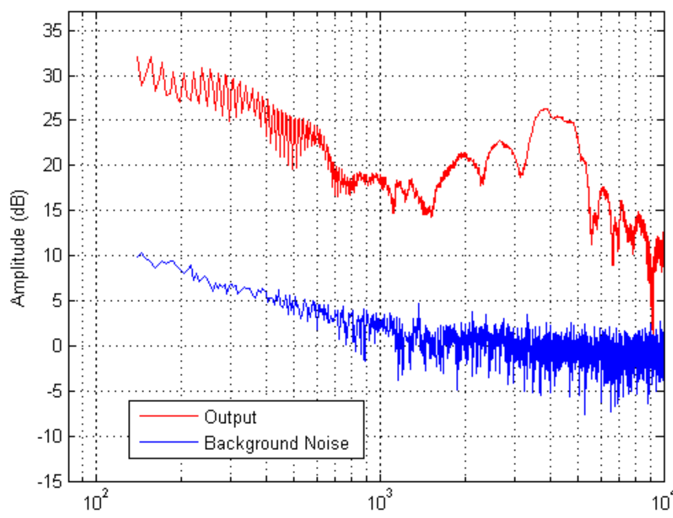



Figure 220. Audio Frequency Spectrum – Pack 8 Switch 2

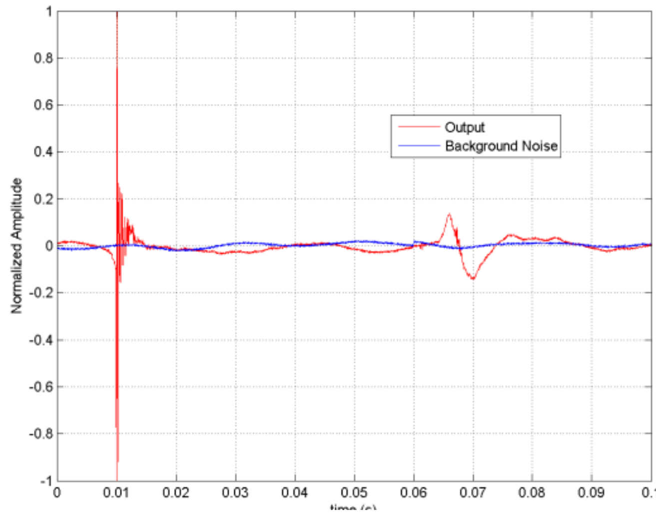



Figure 221. Audio Time Spectrum – Pack 8 Switch 3

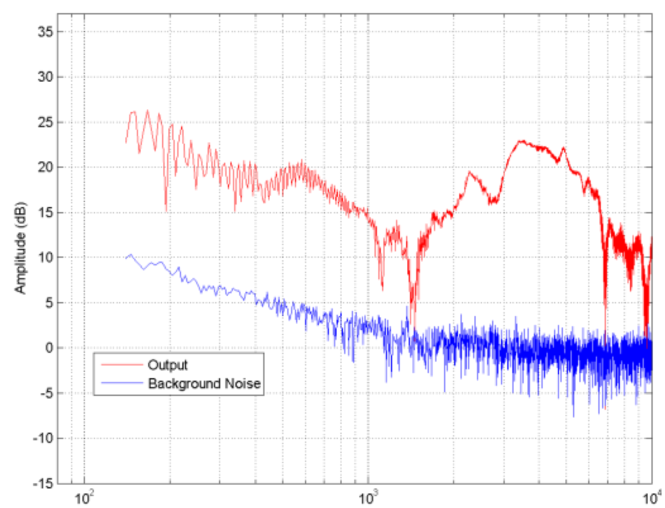



Figure 222. Audio Frequency Spectrum – Pack 8 Switch 3

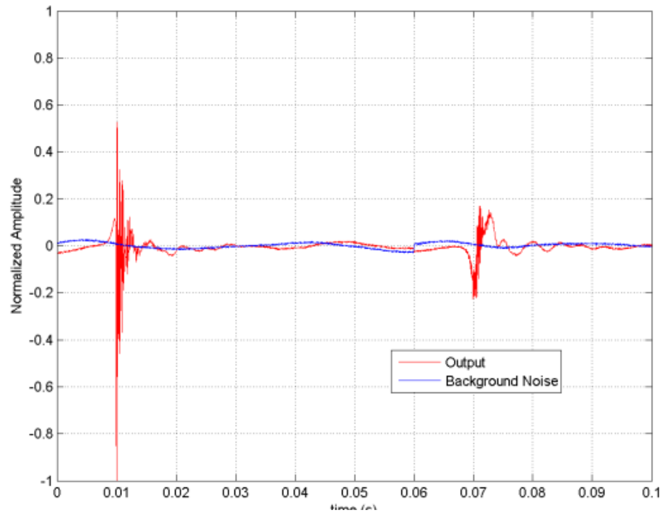



Figure 223. Audio Time Spectrum – Pack 8 Switch 4

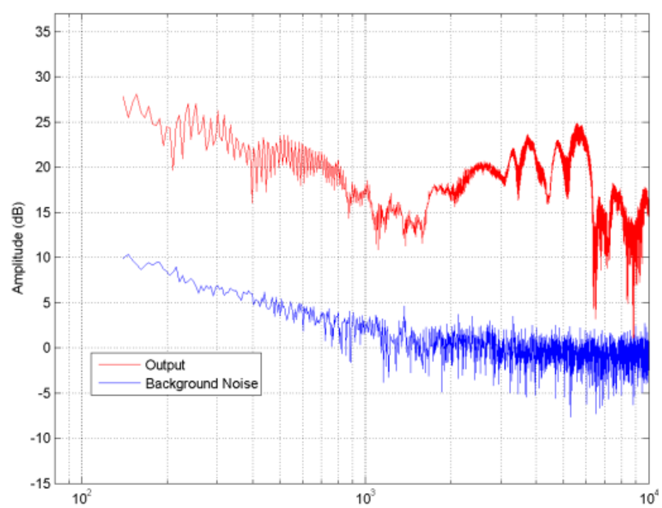



Figure 224. Audio Frequency Spectrum – Pack 8 Switch 4

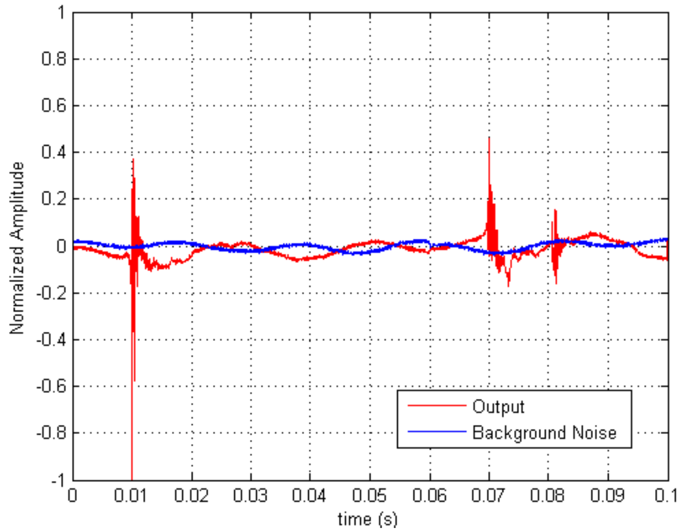



Figure 225. Audio Time Spectrum – Pack 8 Switch 5

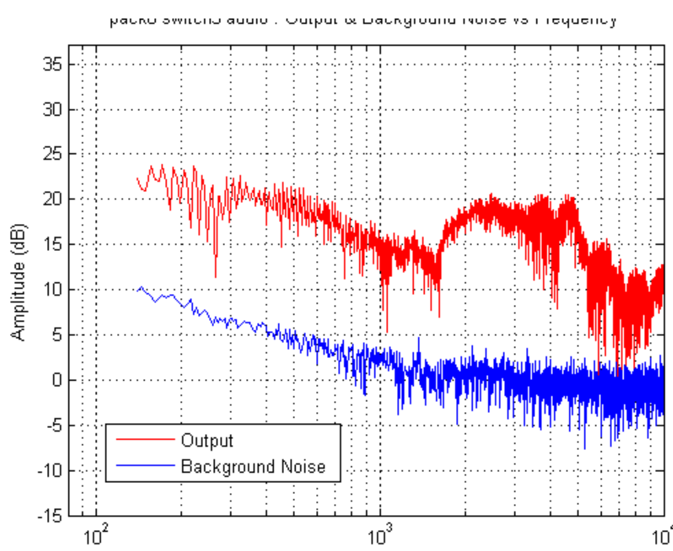



Figure 226. Audio Frequency Spectrum – Pack 8 Switch 5

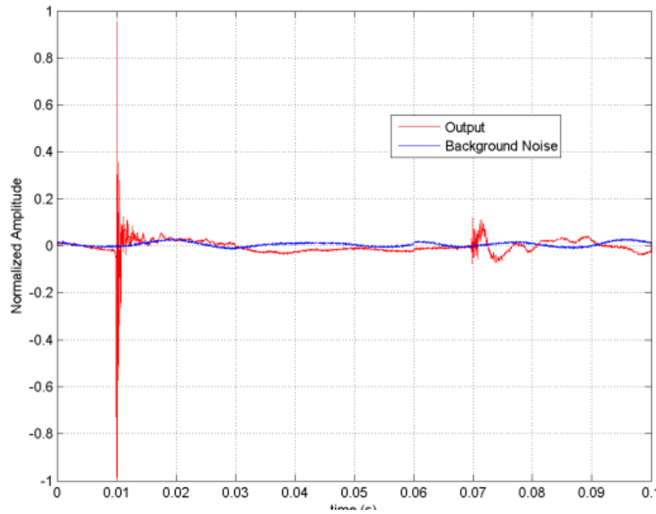



Figure 227. Audio Time Spectrum – Pack 8 Switch 6

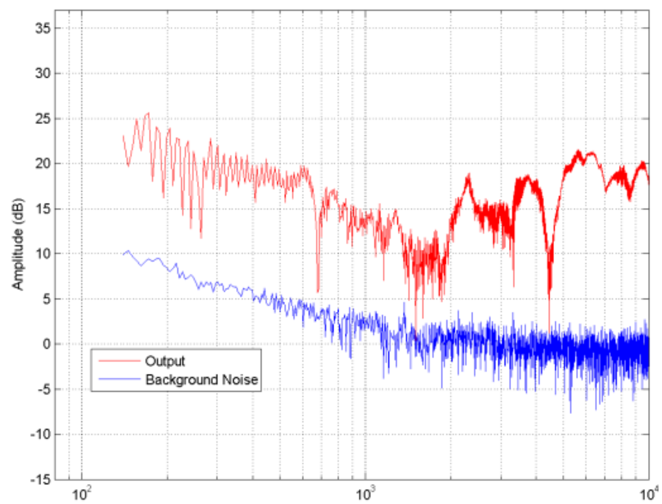



Figure 228. Audio Frequency Spectrum – Pack 8 Switch 6

## Appendix K.9. Pack 9

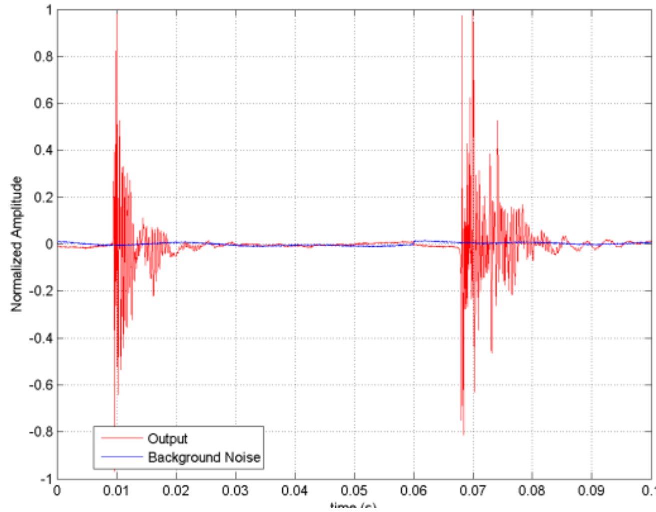



Figure 229. Audio Time Spectrum – Pack 9 Switch 1

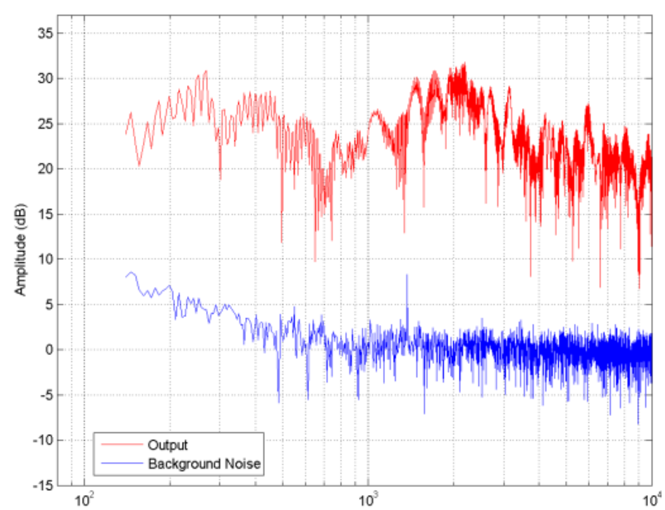



Figure 230. Audio Frequency Spectrum – Pack 9 Switch 1

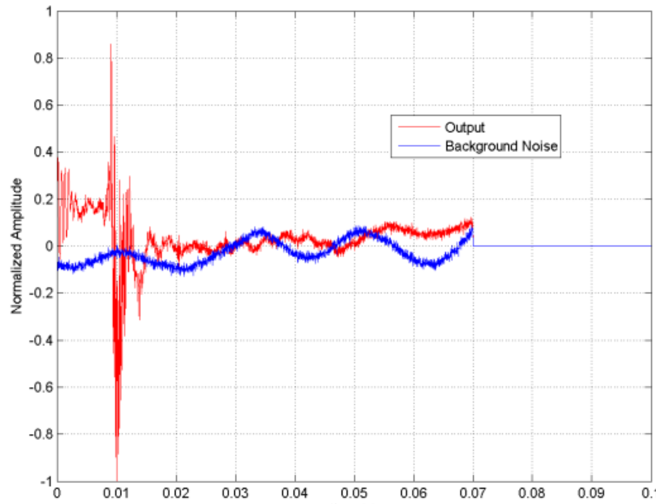



Figure 231. Audio Time Spectrum – Pack 9 Switch 2

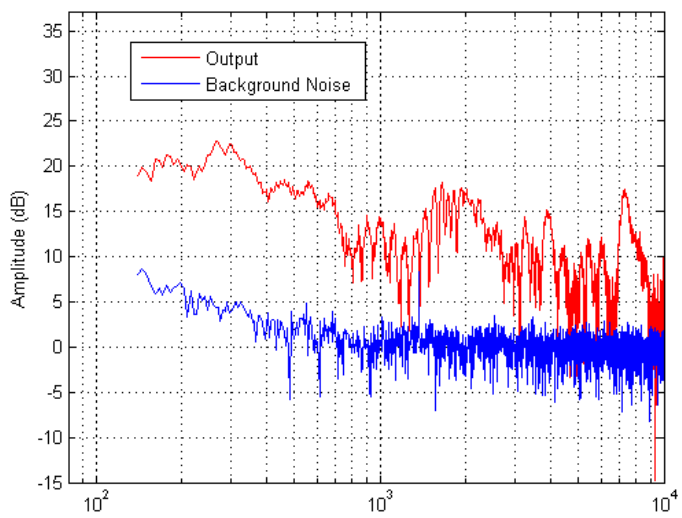



Figure 232. Audio Frequency Spectrum – Pack 9 Switch 2

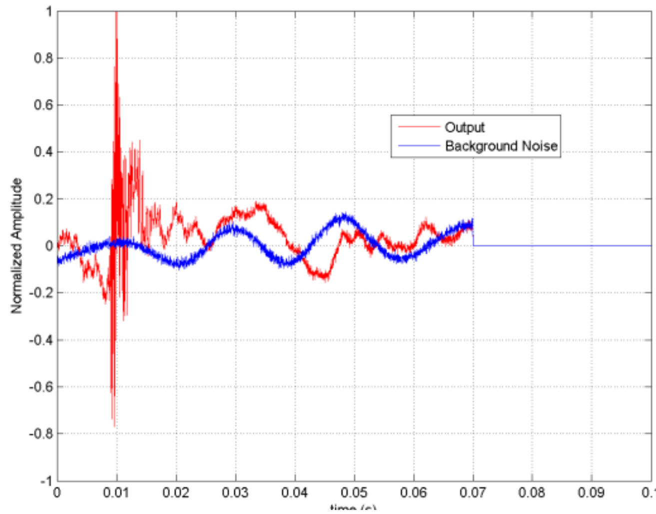



Figure 233. Audio Time Spectrum – Pack 9 Switch 3

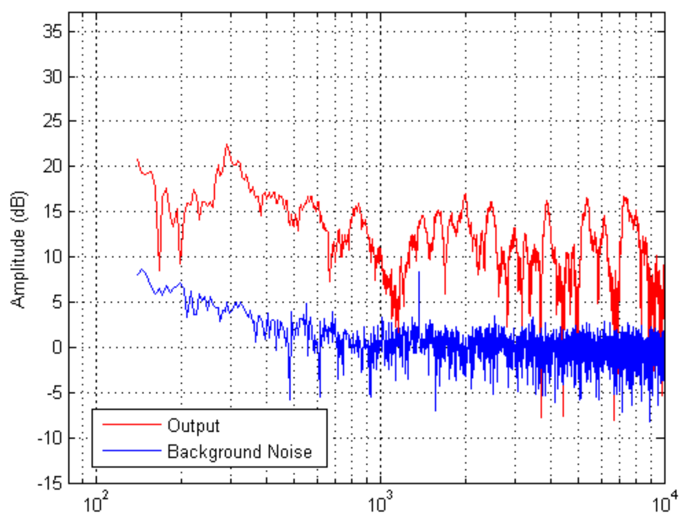



Figure 234. Audio Frequency Spectrum – Pack 9 Switch 3

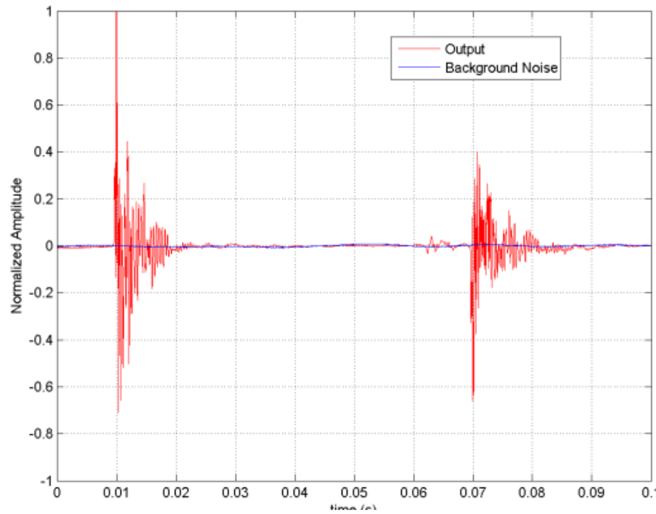



Figure 235. Audio Time Spectrum – Pack 9 Switch 4

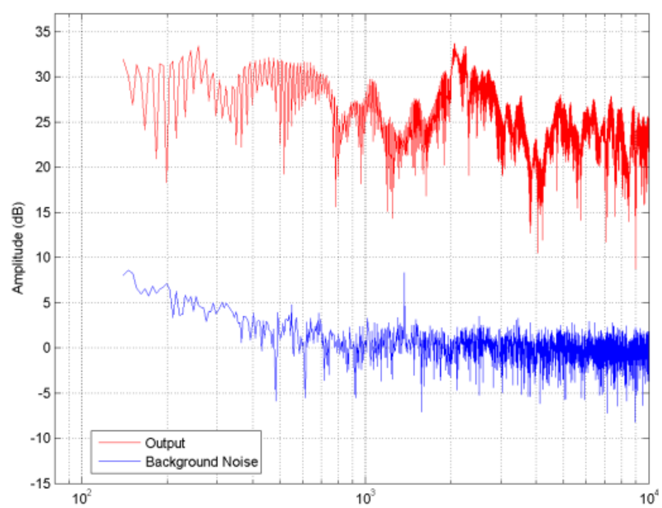



Figure 236. Audio Frequency Spectrum – Pack 9 Switch 4

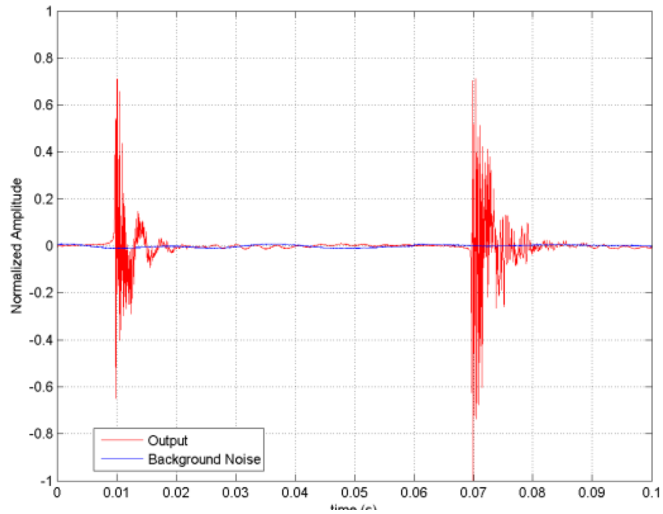



Figure 237. Audio Time Spectrum – Pack 9 Switch 5

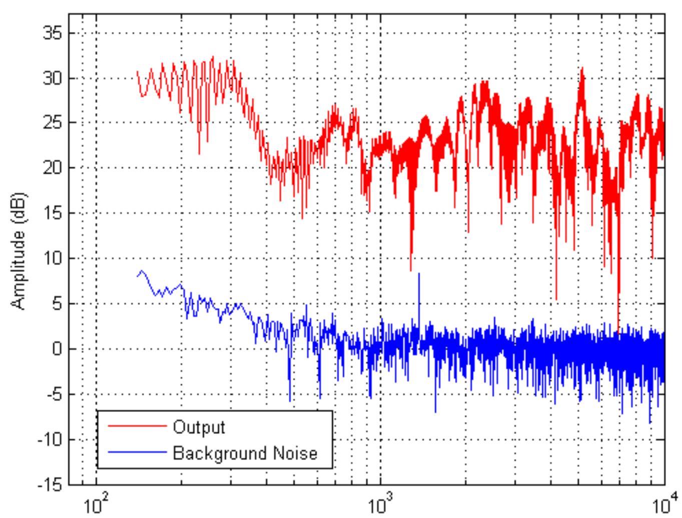



Figure 238. Audio Frequency Spectrum – Pack 9 Switch 5

## Appendix K.10. Pack 10

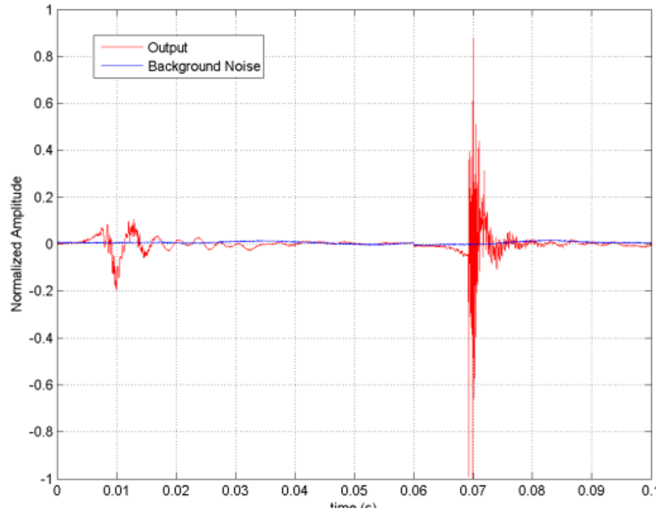



Figure 239. Audio Time Spectrum – Pack 10 Switch 1

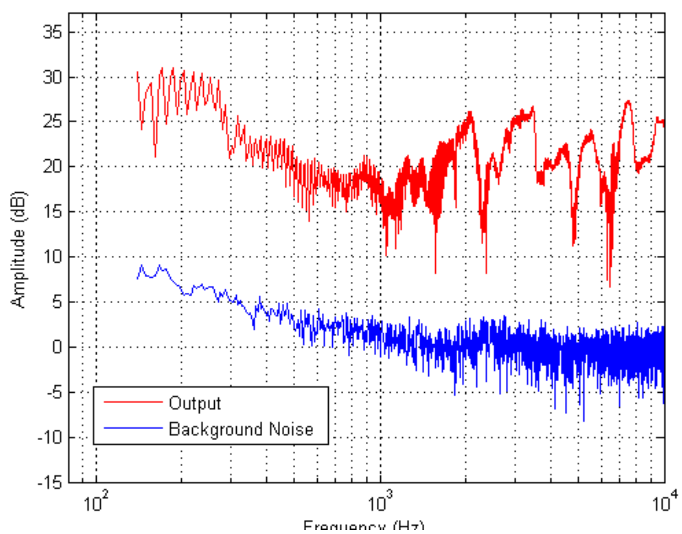



Figure 240. Audio Frequency Spectrum – Pack 10 Switch 1

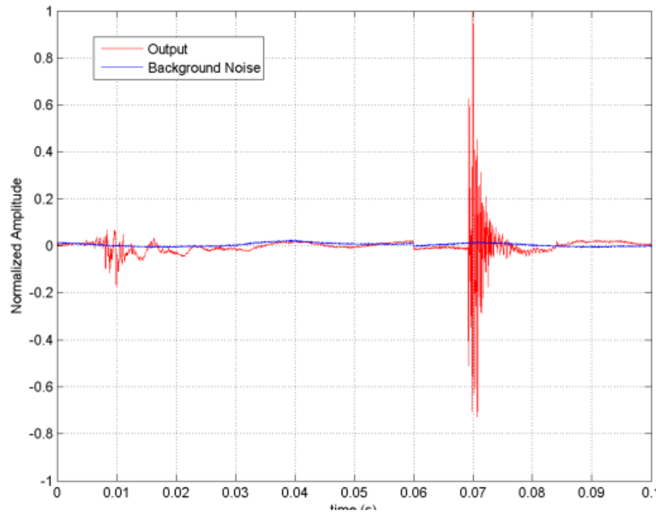



Figure 241. Audio Time Spectrum – Pack 10 Switch 2

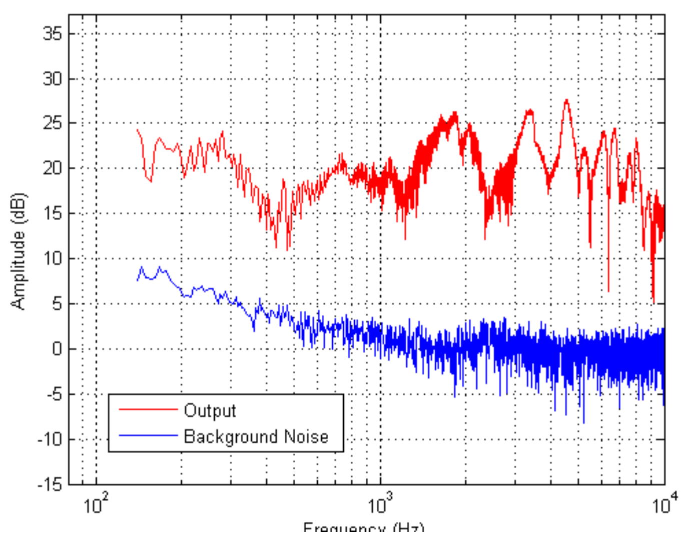



Figure 242. Audio Frequency Spectrum – Pack 10 Switch 2

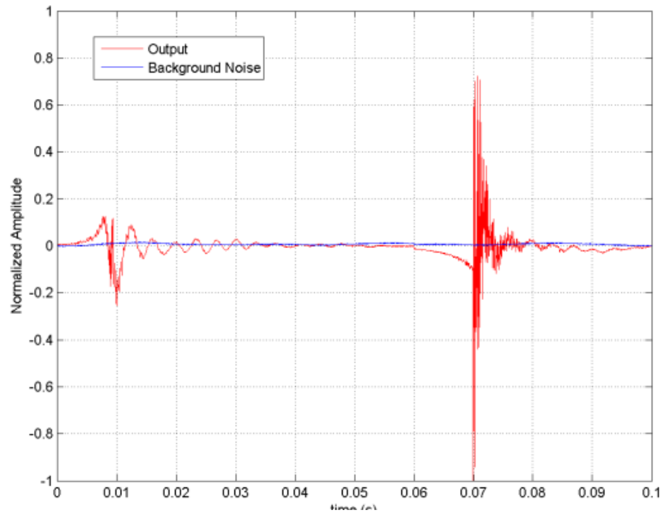



Figure 243. Audio Time Spectrum – Pack 10 Switch 3

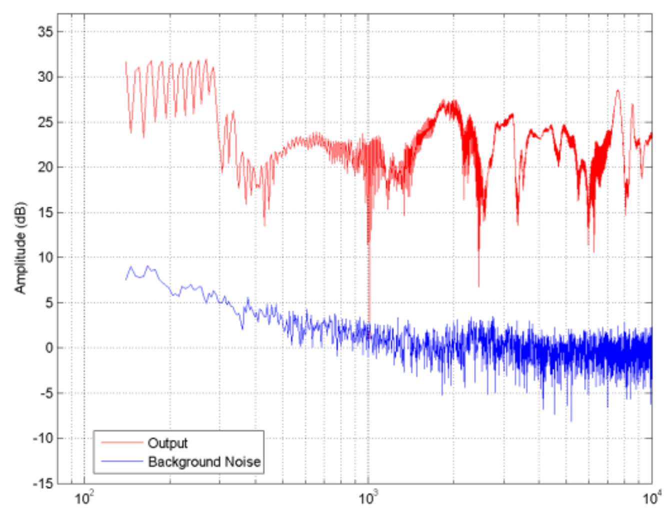



Figure 244. Audio Frequency Spectrum – Pack 10 Switch 3

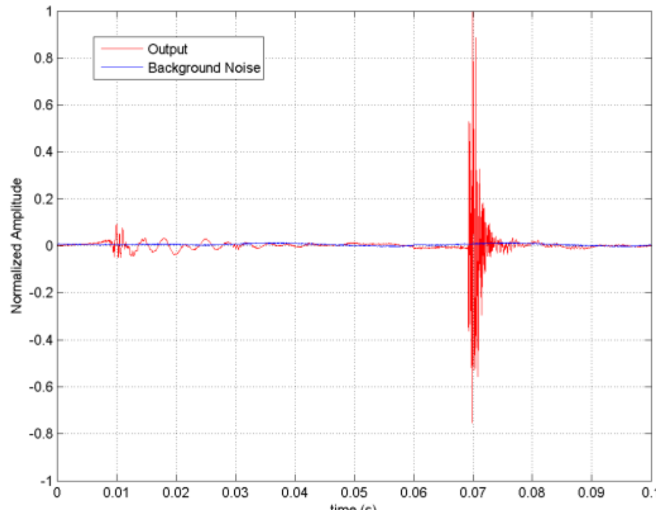



Figure 245. Audio Time Spectrum – Pack 10 Switch 4

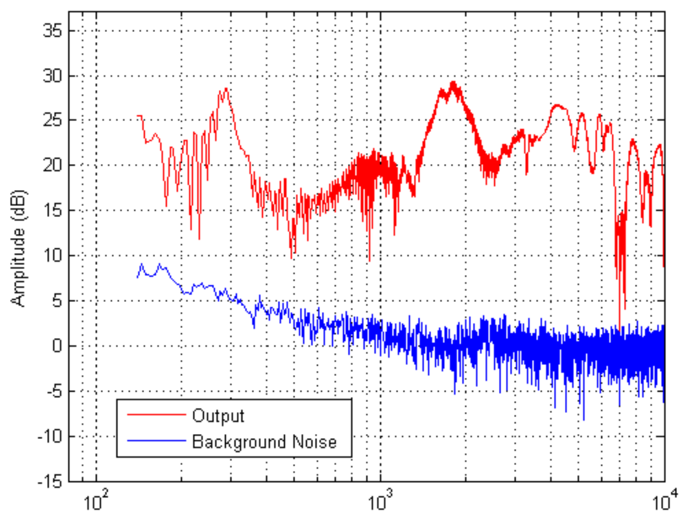



Figure 246. Audio Frequency Spectrum – Pack 10 Switch 4

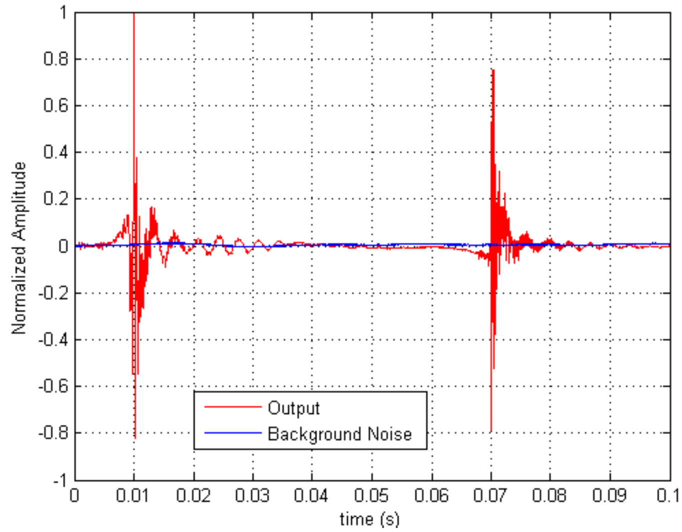



Figure 247. Audio Time Spectrum – Pack 10 Switch 5

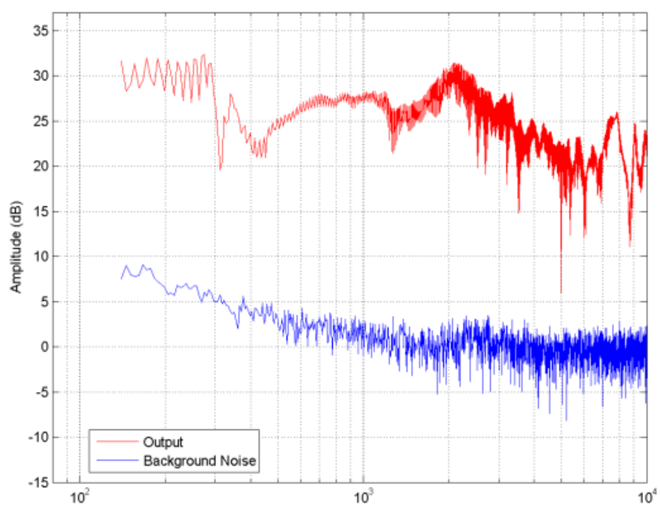



Figure 248. Audio Frequency Spectrum – Pack 10 Switch 5

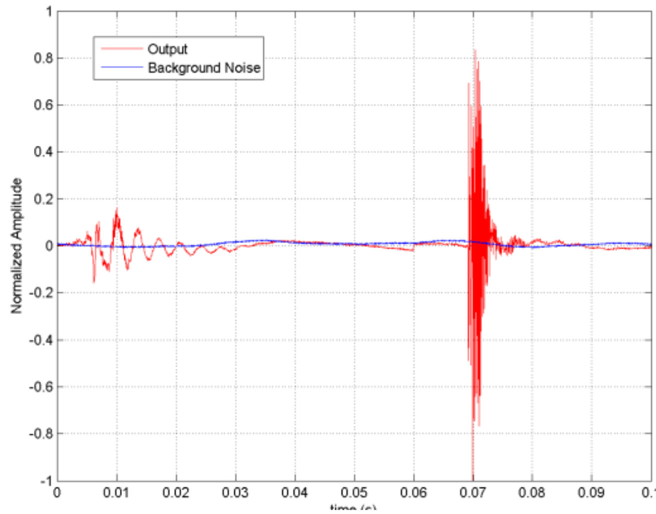



Figure 249. Audio Time Spectrum – Pack 10 Switch 6

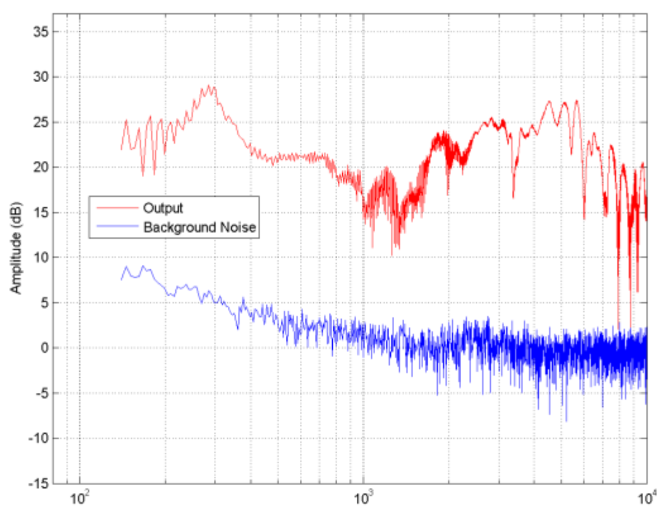



Figure 250. Audio Frequency Spectrum – Pack 10 Switch 6

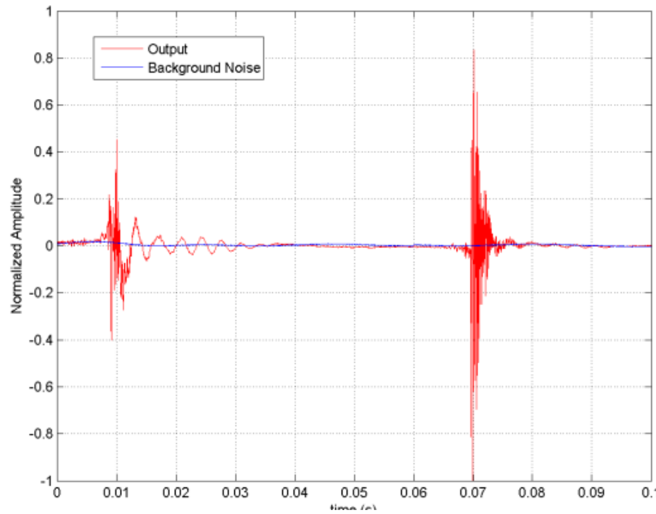



Figure 251. Audio Time Spectrum – Pack 10 Switch 7

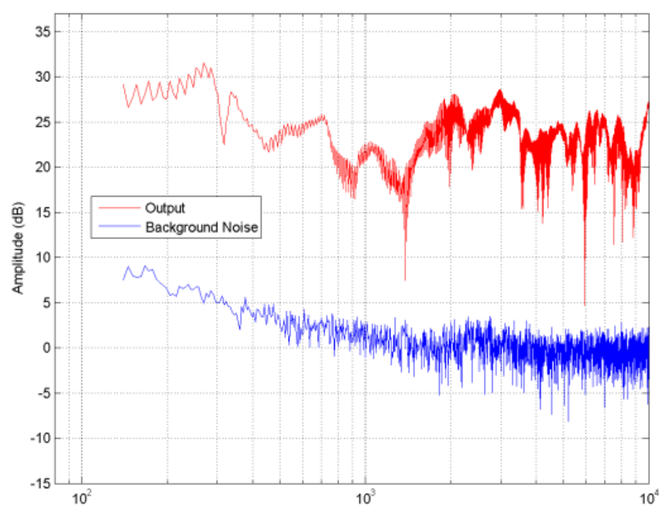



Figure 252. Audio Frequency Spectrum – Pack 10 Switch 7

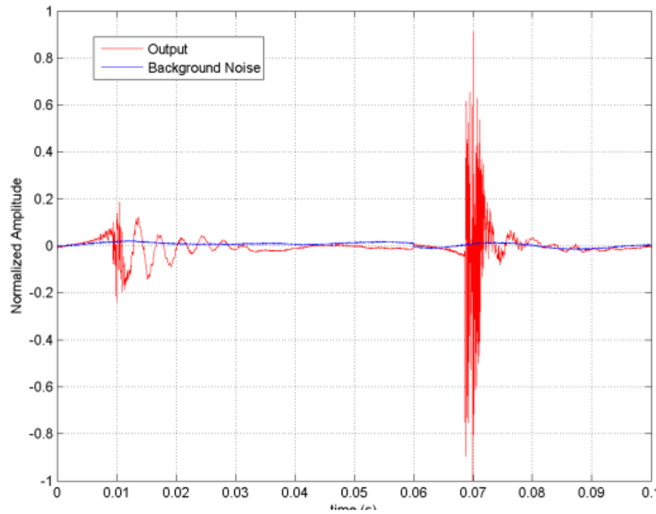



Figure 253. Audio Time Spectrum – Pack 10 Switch 8

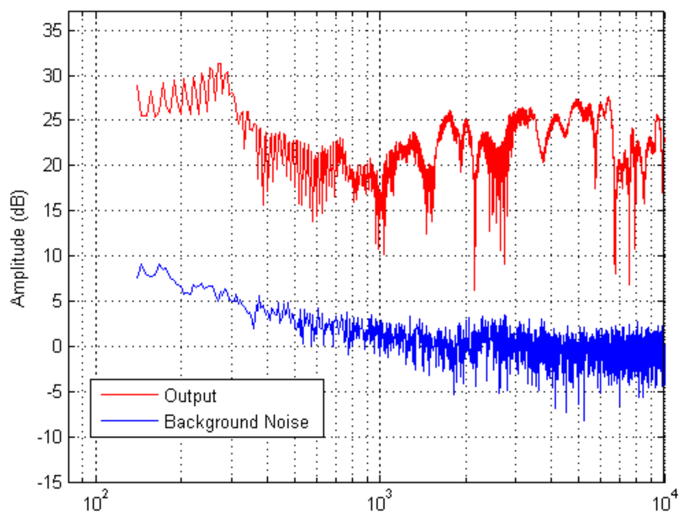



Figure 254. Audio Frequency Spectrum – Pack 10 Switch 8

## Appendix K.11. Pack 11

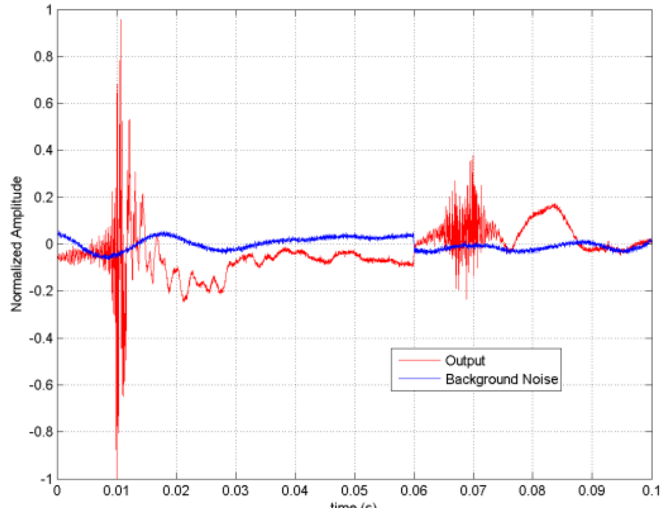



Figure 255. Audio Time Spectrum – Pack 11 Switch 1

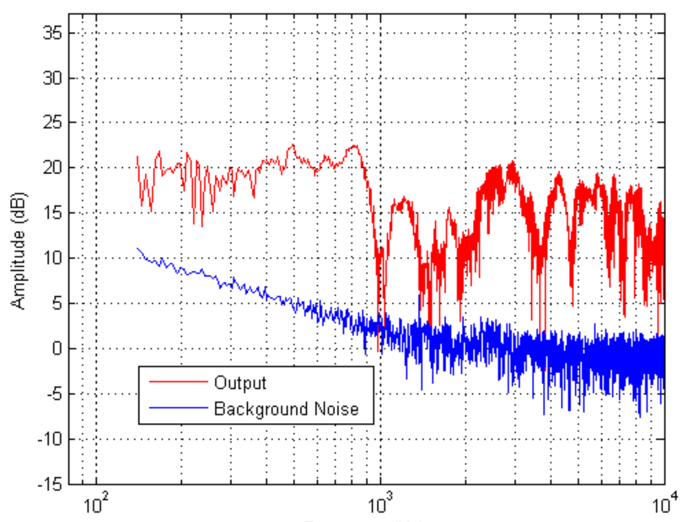



Figure 256. Audio Frequency Spectrum – Pack 11 Switch 1

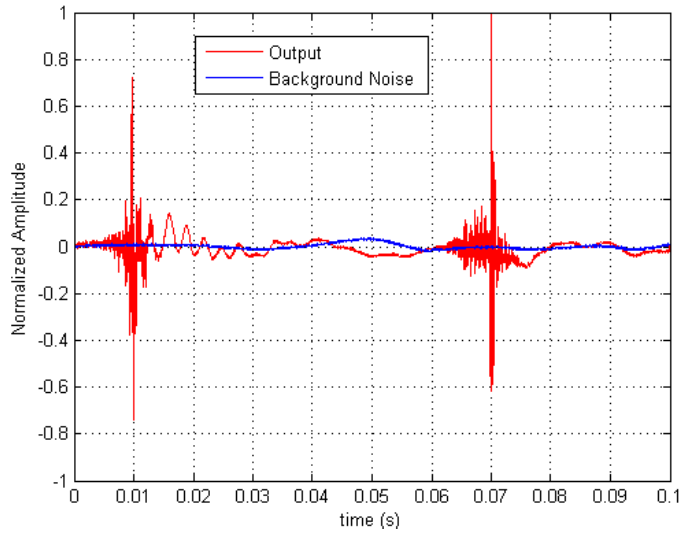



Figure 257. Audio Time Spectrum – Pack 11 Switch 2

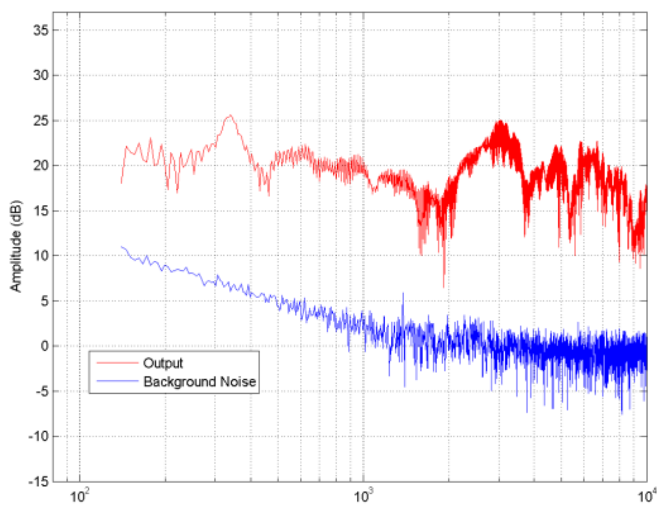



Figure 258. Audio Frequency Spectrum – Pack 11 Switch 2

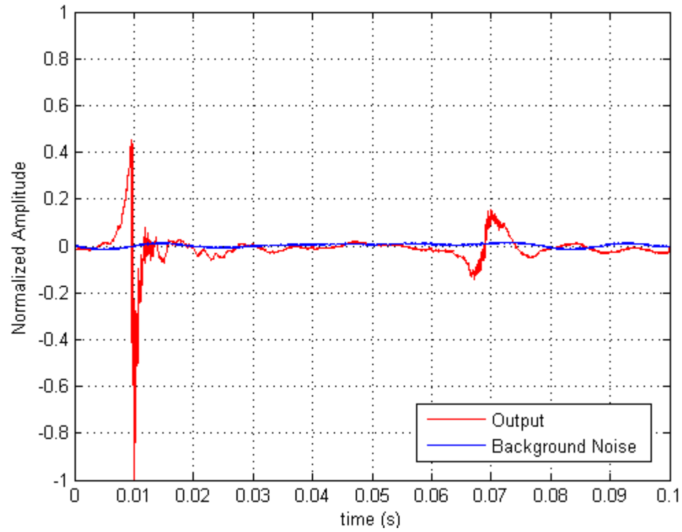



Figure 259. Audio Time Spectrum – Pack 11 Switch 3

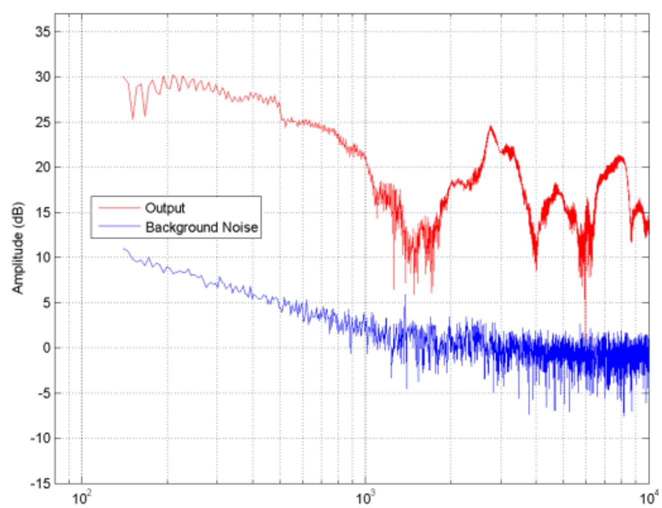



Figure 260. Audio Frequency Spectrum – Pack 11 Switch 3

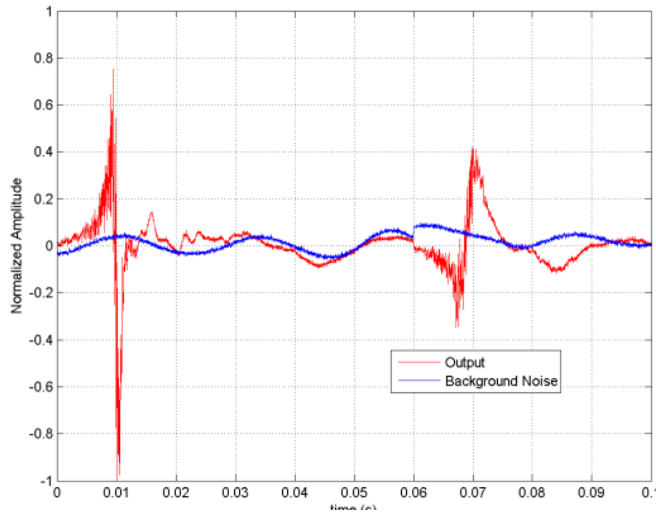



Figure 261. Audio Time Spectrum – Pack 11 Switch 4

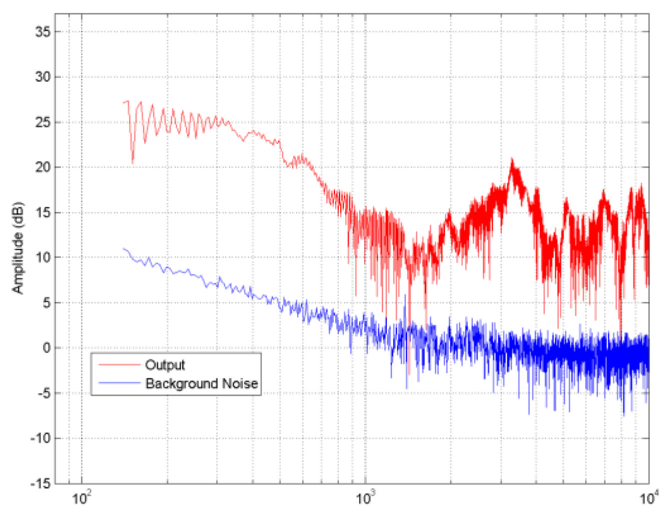



Figure 262. Audio Frequency Spectrum – Pack 11 Switch 4

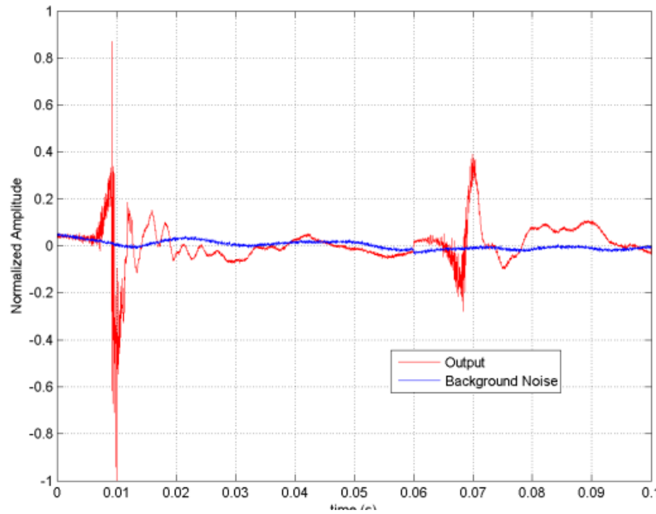



Figure 263. Time Spectrum – Pack 11 Switch 5

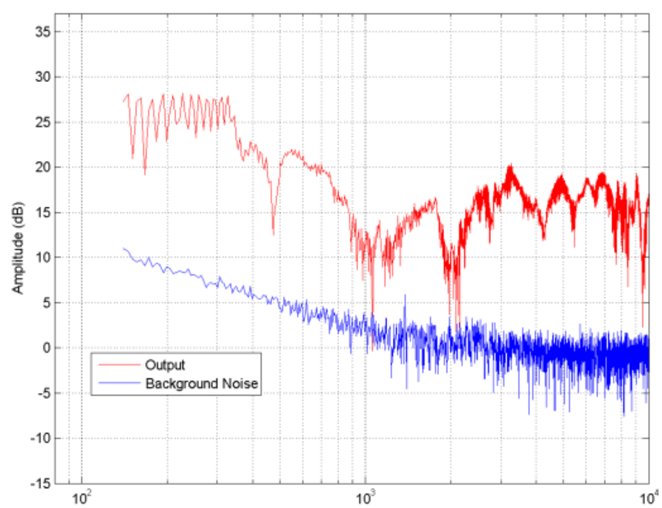



Figure 264. Frequency Spectrum – Pack 11 Switch 5

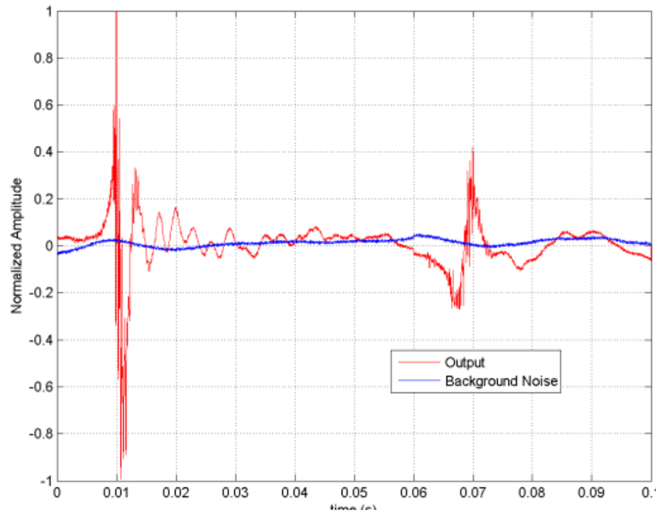



Figure 265. Audio Time Spectrum – Pack 11 Switch 6

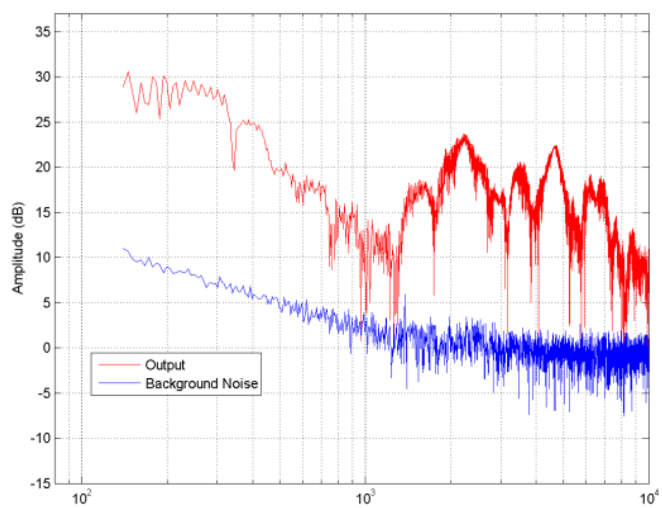



Figure 266. Audio Frequency Spectrum – Pack 11 Switch 6

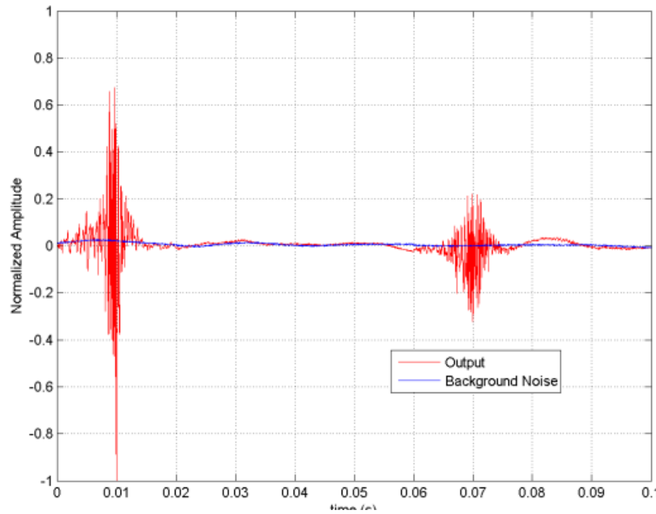



Figure 267. Audio Time Spectrum – Pack 11 Switch 7

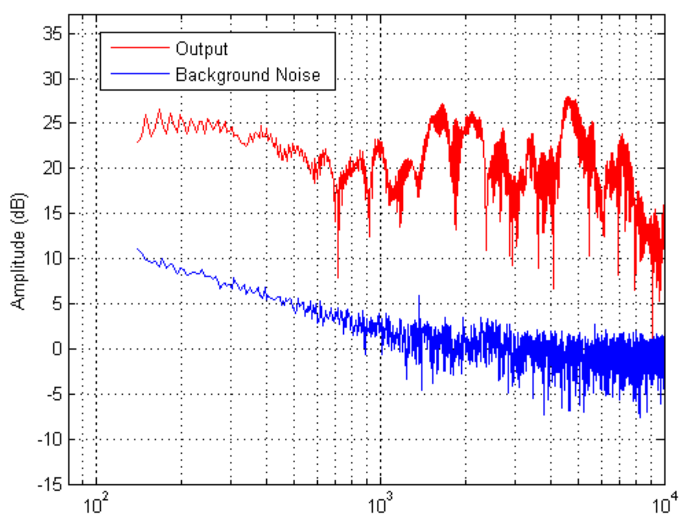



Figure 268. Audio Frequency Spectrum – Pack 11 Switch 7

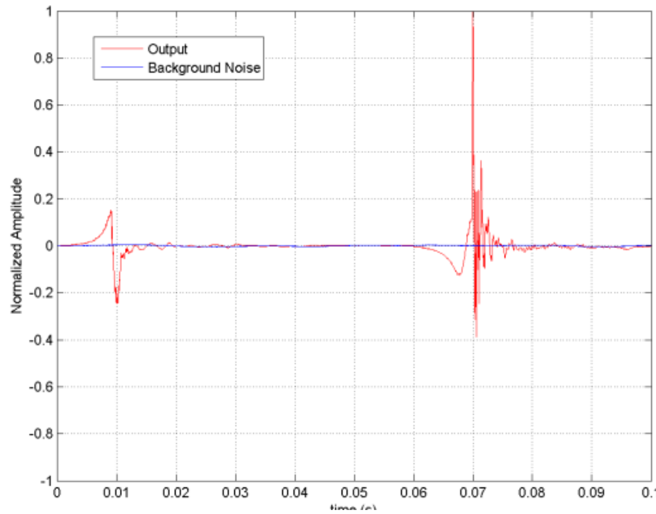



Figure 269. Audio Time Spectrum – Pack 11 Switch 8

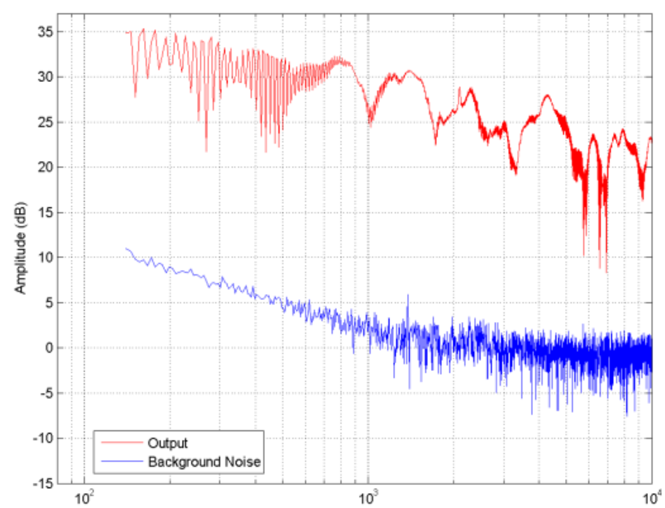



Figure 270. Audio Frequency Spectrum – Pack 11 Switch 8

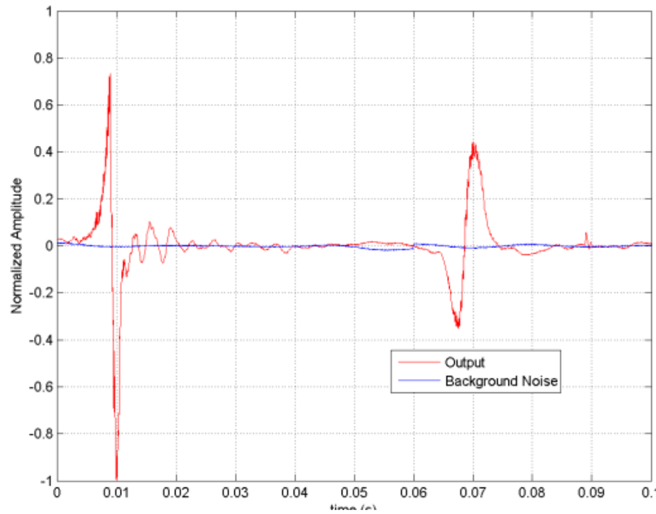



Figure 271. Audio Time Spectrum – Pack 11 Switch 9

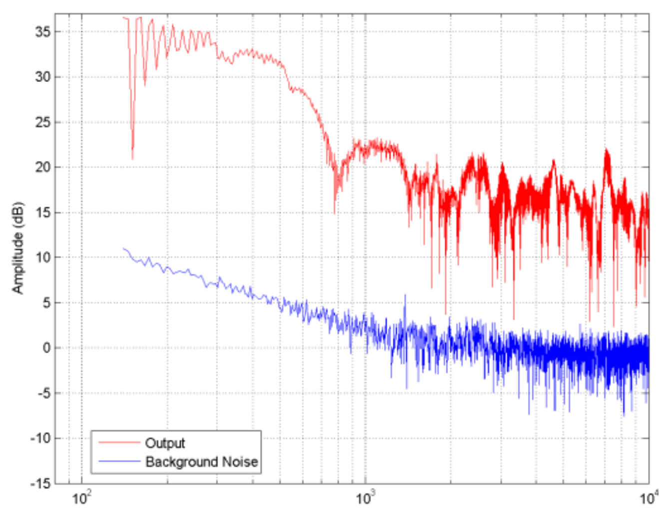



Figure 272. Audio Frequency Spectrum – Pack 11 Switch 9

## Appendix L. Google Survey Screenshots

# Sound Quality of Automotive Button Sounds - Jury Evaluation

Subject Questionnaire

* Required

**Research Subject Number ***

Given to you by the investigator

**Gender ***

- ○ Male
- ○ Female

**Are you aware of an existing hearing impairment? ***

- ○ Yes
- ○ No

**Have you had a cold or other sinus infection/illness in the past month? ***

- ○ Yes
- ○ No

**Do you currently own or regularly operate a motor vehicle? ***

- ○ Yes
- ○ No

When you are ready to continue, please press "Begin Test" on the dialogue box and proceed to the next page

Continue »

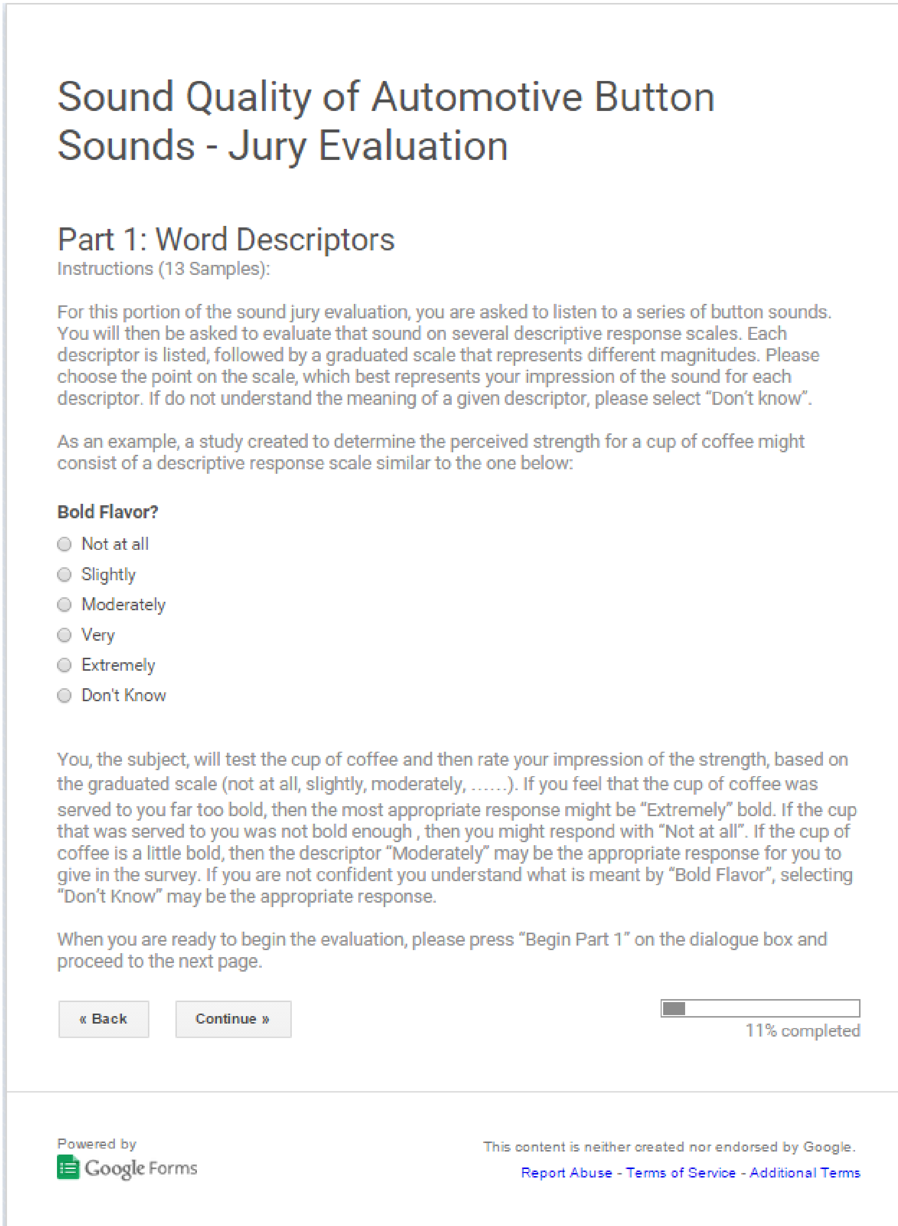
5% completed

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure 273. Google Survey Page 1



# Sound Quality of Automotive Button Sounds - Jury Evaluation

## Part 1: Word Descriptors

Instructions (13 Samples):

For this portion of the sound jury evaluation, you are asked to listen to a series of button sounds. You will then be asked to evaluate that sound on several descriptive response scales. Each descriptor is listed, followed by a graduated scale that represents different magnitudes. Please choose the point on the scale, which best represents your impression of the sound for each descriptor. If do not understand the meaning of a given descriptor, please select "Don't know".

As an example, a study created to determine the perceived strength for a cup of coffee might consist of a descriptive response scale similar to the one below:

**Bold Flavor?**

- ○ Not at all
- ○ Slightly
- ○ Moderately
- ○ Very
- ○ Extremely
- ○ Don't Know

You, the subject, will test the cup of coffee and then rate your impression of the strength, based on the graduated scale (not at all, slightly, moderately, ……). If you feel that the cup of coffee was served to you far too bold, then the most appropriate response might be "Extremely" bold. If the cup that was served to you was not bold enough , then you might respond with "Not at all". If the cup of coffee is a little bold, then the descriptor "Moderately" may be the appropriate response for you to give in the survey. If you are not confident you understand what is meant by "Bold Flavor", selecting "Don't Know" may be the appropriate response.

When you are ready to begin the evaluation, please press "Begin Part 1" on the dialogue box and proceed to the next page.

« Back | Continue »

11% completed

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure 274. Google Survey Page 2

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 1

Part 1: Word Descriptors

**Sharp? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back   Continue »

16% completed

Powered by Google Forms



Figure 275. Google Survey Page 3

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 2

Part 1: Word Descriptors

**Sharp? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back    Continue »

16% completed



Figure 276. Google Survey Page 4

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 3

Part 1: Word Descriptors

**Sharp? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back | Continue »

16% completed



Figure 277. Google Survey Page 5

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 4

Part 1: Word Descriptors

**Sharp? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back | Continue »

16% completed

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure 278. Google Survey Page 6

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 5

Part 1: Word Descriptors

**Sharp? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back Continue »

16% completed

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure 279. Google Survey Page 7

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 6

Part 1: Word Descriptors

**Sharp? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back | Continue »

16% completed

Powered by Google Forms

This content is neither created nor endorsed by Google.
Report Abuse - Terms of Service - Additional Terms

Figure 280. Google Survey Page 8

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 7

Part 1: Word Descriptors

**Sharp?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back    Continue »

16% completed

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure 281. Google Survey Page 9

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 8

Part 1: Word Descriptors

**Sharp?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back    Continue »

16% completed

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure 282. Google Survey Page 10

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 9

Part 1: Word Descriptors

**Sharp?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back    Continue »

16% completed



Figure 283. Google Survey Page 11

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 10

Part 1: Word Descriptors

**Sharp? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful? ***
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back Continue »

16% completed

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure 284. Google Survey Page 12

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 11

Part 1: Word Descriptors

| Sharp? * | Dynamic? * | Tonal? * | Expressive? * |
|---|---|---|---|
| Not at all | Not at all | Not at all | Not at all |
| Slightly | Slightly | Slightly | Slightly |
| Moderately | Moderately | Moderately | Moderately |
| Very | Very | Very | Very |
| Extremely | Extremely | Extremely | Extremely |
| Don't Know | Don't Know | Don't Know | Don't Know |

| Dark? * | Intense? * | Loud? * | Solid? * |
|---|---|---|---|
| Not at all | Not at all | Not at all | Not at all |
| Slightly | Slightly | Slightly | Slightly |
| Moderately | Moderately | Moderately | Moderately |
| Very | Very | Very | Very |
| Extremely | Extremely | Extremely | Extremely |
| Don't Know | Don't Know | Don't Know | Don't Know |

| High (frequency)? * | Low (frequency)? * | Smooth? * | Reliable? * |
|---|---|---|---|
| Not at all | Not at all | Not at all | Not at all |
| Slightly | Slightly | Slightly | Slightly |
| Moderately | Moderately | Moderately | Moderately |
| Very | Very | Very | Very |
| Extremely | Extremely | Extremely | Extremely |
| Don't Know | Don't Know | Don't Know | Don't Know |

| Long? * | Short? * | Strong? * | Powerful? * |
|---|---|---|---|
| Not at all | Not at all | Not at all | Not at all |
| Slightly | Slightly | Slightly | Slightly |
| Moderately | Moderately | Moderately | Moderately |
| Very | Very | Very | Very |
| Extremely | Extremely | Extremely | Extremely |
| Don't Know | Don't Know | Don't Know | Don't Know |

« Back   Continue »

16% completed



Figure 285. Google Survey Page 13

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 12

Part 1: Word Descriptors

**Sharp?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dynamic?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Tonal?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Expressive?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Dark?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Intense?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Loud?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Solid?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**High (frequency)?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Low (frequency)?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Smooth?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Reliable?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Long?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Short?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Strong?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

**Powerful?** *
- Not at all
- Slightly
- Moderately
- Very
- Extremely
- Don't Know

« Back  Continue »

16% completed

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure 286. Google Survey Page 14

# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Sample 13

Part 1: Word Descriptors

| Sharp? * | Dynamic? * | Tonal? * | Expressive? * |
| --- | --- | --- | --- |
| Not at all | Not at all | Not at all | Not at all |
| Slightly | Slightly | Slightly | Slightly |
| Moderately | Moderately | Moderately | Moderately |
| Very | Very | Very | Very |
| Extremely | Extremely | Extremely | Extremely |
| Don't Know | Don't Know | Don't Know | Don't Know |

| Dark? * | Intense? * | Loud? * | Solid? * |
| --- | --- | --- | --- |
| Not at all | Not at all | Not at all | Not at all |
| Slightly | Slightly | Slightly | Slightly |
| Moderately | Moderately | Moderately | Moderately |
| Very | Very | Very | Very |
| Extremely | Extremely | Extremely | Extremely |
| Don't Know | Don't Know | Don't Know | Don't Know |

| High (frequency)? * | Low (frequency)? * | Smooth? * | Reliable? * |
| --- | --- | --- | --- |
| Not at all | Not at all | Not at all | Not at all |
| Slightly | Slightly | Slightly | Slightly |
| Moderately | Moderately | Moderately | Moderately |
| Very | Very | Very | Very |
| Extremely | Extremely | Extremely | Extremely |
| Don't Know | Don't Know | Don't Know | Don't Know |

| Long? * | Short? * | Strong? * | Powerful? * |
| --- | --- | --- | --- |
| Not at all | Not at all | Not at all | Not at all |
| Slightly | Slightly | Slightly | Slightly |
| Moderately | Moderately | Moderately | Moderately |
| Very | Very | Very | Very |
| Extremely | Extremely | Extremely | Extremely |
| Don't Know | Don't Know | Don't Know | Don't Know |

« Back   Continue »

16% completed

Powered by Google Forms

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Figure 287. Google Survey Page 15

# Sound Quality of Automotive Button Sounds - Jury Evaluation

## Part 2: Command Function Descriptors

To complete the sound jury survey, some training is required, so that you, the subject, understand what the questionnaire is asking for. Please read the definitions below to ensure full understanding of the meaning of each command function.

When responding to the questionnaire, you are asked to match each sample to an appropriate command function.

Cruise
1. Set +
   When cruise control is on and active, increases the set speed by 1 mile per hour
2. Set -
   When cruise control is on and active, decreases the set speed by 1 mile per hour
3. On
   Turns on cruise control.
4. Off
   Turns off cruise control.

Phone
5. Answer
   Answer incoming phone call
6. End
   End current phone call

Media
7. Media Mode
   Change the media source (CD, FM/AM, Aux, etc.)
8. Next Track
   Advance to the next media track
9. Previous Track
   Return to the previous media track
10. Play / Pause
    Play / pause the current media track
11. Mute - On
    Mute the current media track
12. Mute - Off
    Unmute the current media track
13. Volume Up
    Increase the media volume by one unit
14. Volume Down
    Decrease the media volume by one unit

Voice
15. Activate
    Begin listening for voice-issued commands

Now that you have read the definitions for each command function, please continue to the next page to read the instructions for the study.

« Back    Continue »

88% completed

Powered by Google Forms



Figure 288. Google Survey Page 16

# Sound Quality of Automotive Button Sounds - Jury Evaluation

## Part 2: Command Function Descriptors

Instructions (17 Samples):

For this sound jury evaluation, you are asked to listen to a series of automotive button sounds. You will then be asked to match each sound to the command function that you believe is best associated with it. A command function is the action that takes place once the button is pressed. For example, the command function for the power button on a computer would be "power on/off". Each command function must be assigned a sound, and you may assign a single sound to more than one command function if you so choose.

PLEASE WRITE THE APPROPRIATE NUMBER TO INDICATE YOUR RESPONSE.

As an example, a study created to determine the seasons in which customers prefer different flavors of coffee:

1. Sample 1
2. Sample 2
3. Sample 3
4. Sample 4

**Spring**

**Summer**

**Autumn**

**Winter**

You, the subject, will test each cup of coffee and then match it with the season you believe it would be most desirable to drink it in. Each season must be assigned only one sample, though you may assign a single sample to more than one season if you so choose. If you feel the second cup of coffee has flavors of pumpkin, then you may choose to pair it with the autumn season, and thus would write the number 2 in the blank below "Autumn". If you feel the third cup of coffee would be ideal to drink in autumn and summer, then you may write the number 3 in the blanks below both "Autumn" and "Summer".

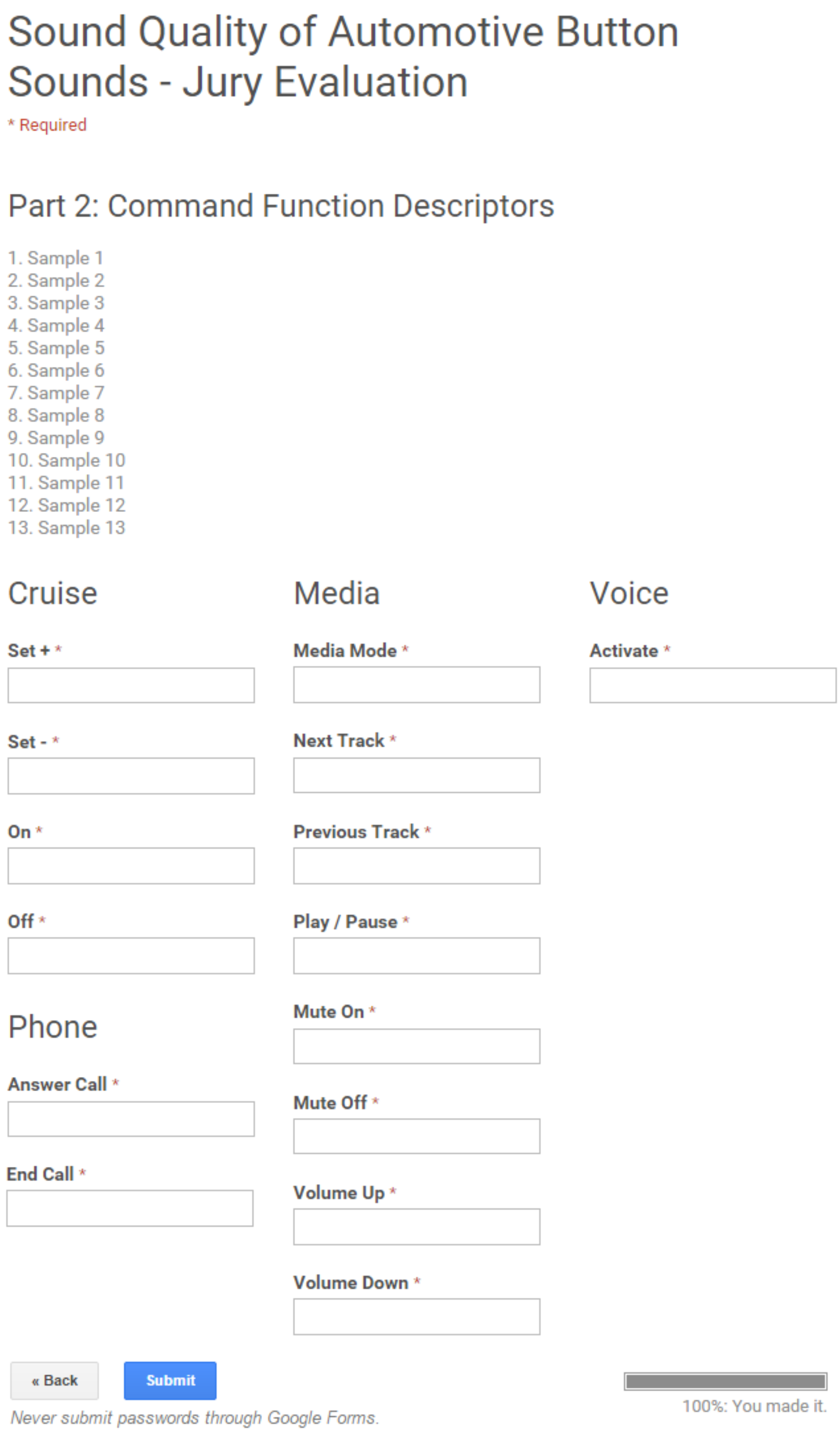
When you are ready to begin the evaluation, please press "Begin Part 2" on the dialogue box and proceed to the next page.

« Back    Continue »

94% completed



Figure 289. Google Survey Page 17



# Sound Quality of Automotive Button Sounds - Jury Evaluation

* Required

## Part 2: Command Function Descriptors

1. Sample 1
2. Sample 2
3. Sample 3
4. Sample 4
5. Sample 5
6. Sample 6
7. Sample 7
8. Sample 8
9. Sample 9
10. Sample 10
11. Sample 11
12. Sample 12
13. Sample 13

### Cruise

**Set +** *

**Set -** *

**On** *

**Off** *

### Phone

**Answer Call** *

**End Call** *

### Media

**Media Mode** *

**Next Track** *

**Previous Track** *

**Play / Pause** *

**Mute On** *

**Mute Off** *

**Volume Up** *

**Volume Down** *

### Voice

**Activate** *

« Back | Submit

100%: You made it.

*Never submit passwords through Google Forms.*

Figure 290. Google Survey Page 18

## Appendix M. MATLAB Figure: GUI Screenshots

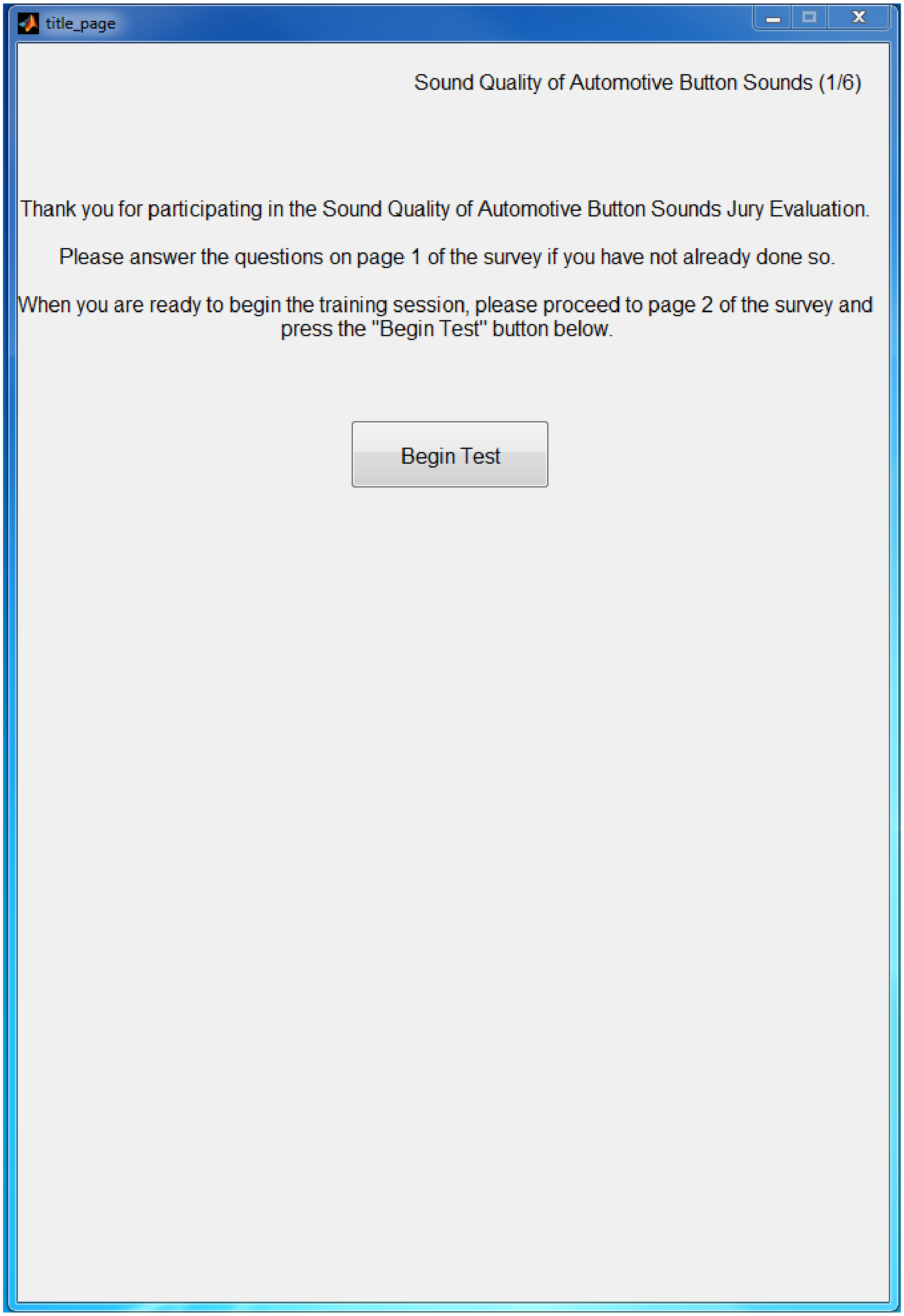



Figure 291. MATLAB GUI Title Page

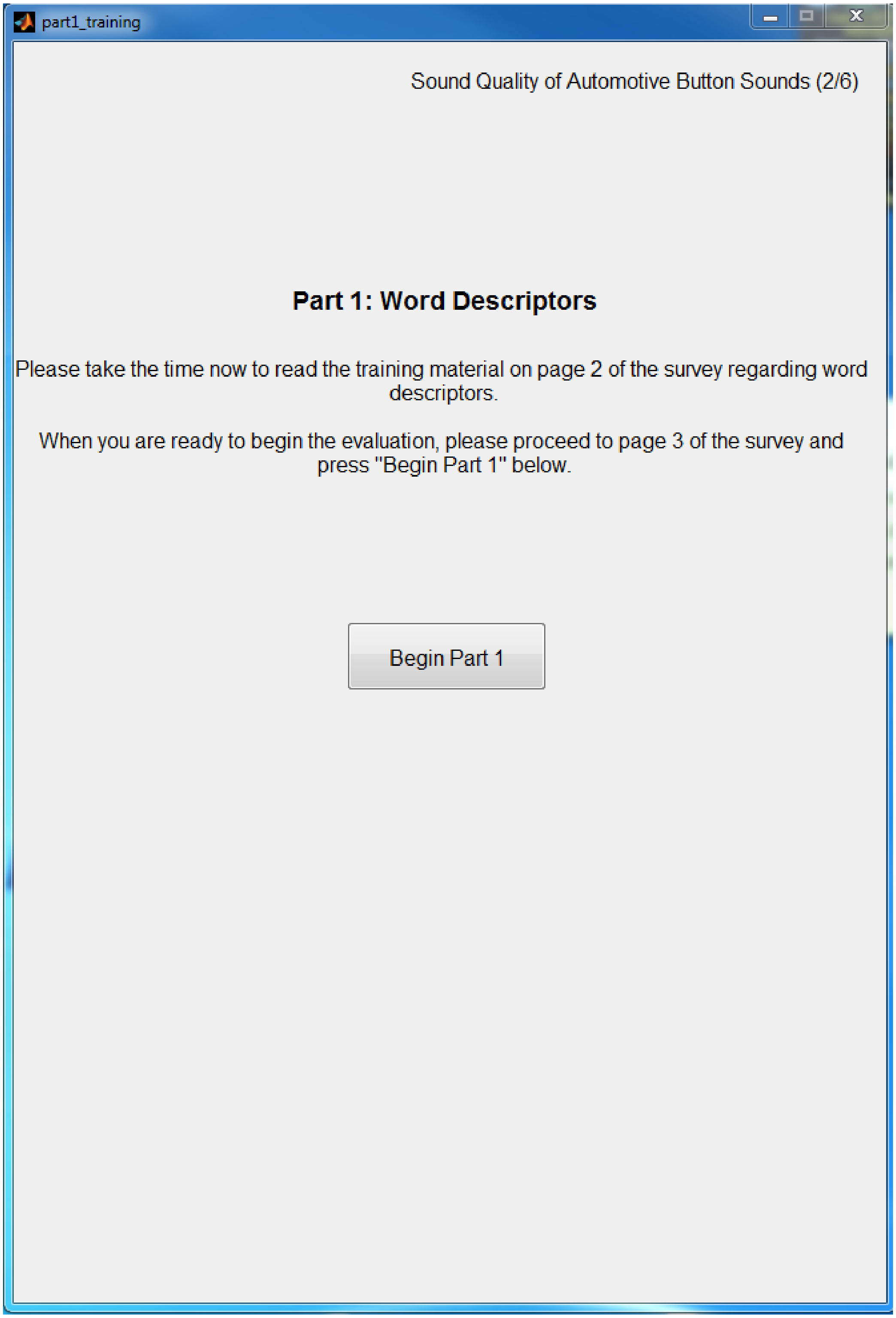



Figure 292. MATLAB GUI Part 1 Training

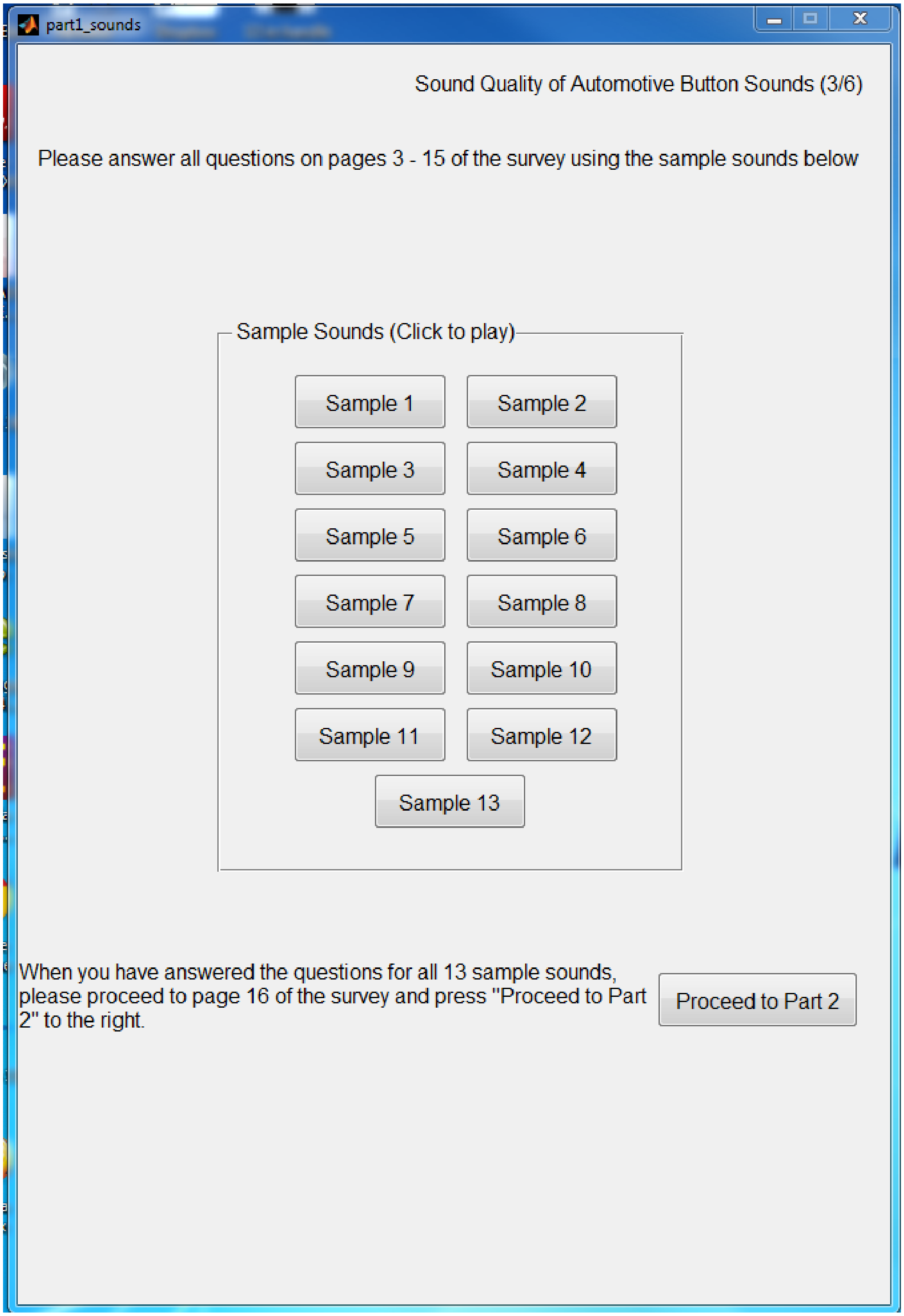



Figure 293. MATLAB GUI Part 1 Sounds

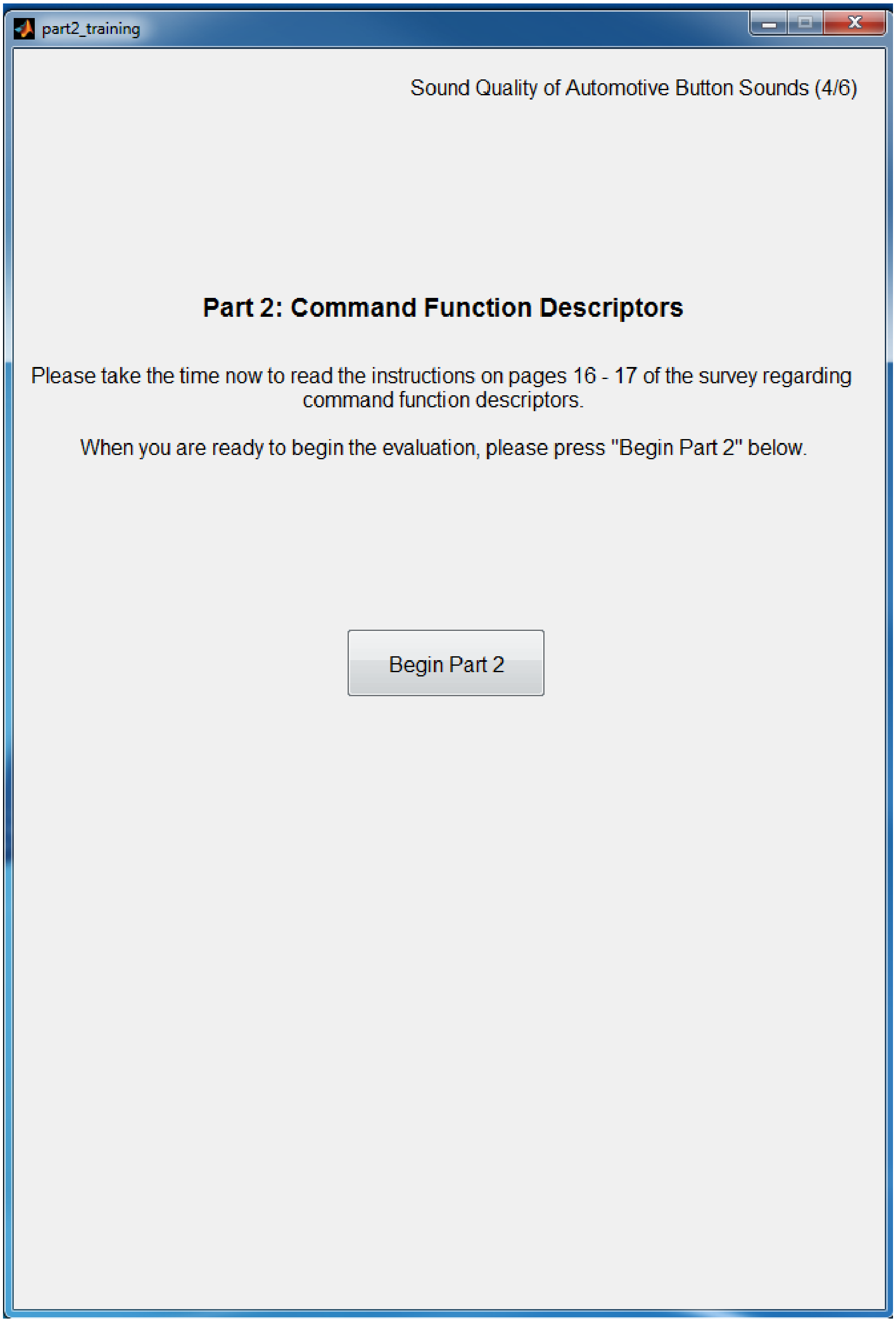



Figure 294. MATLAB GUI Part 2 Training

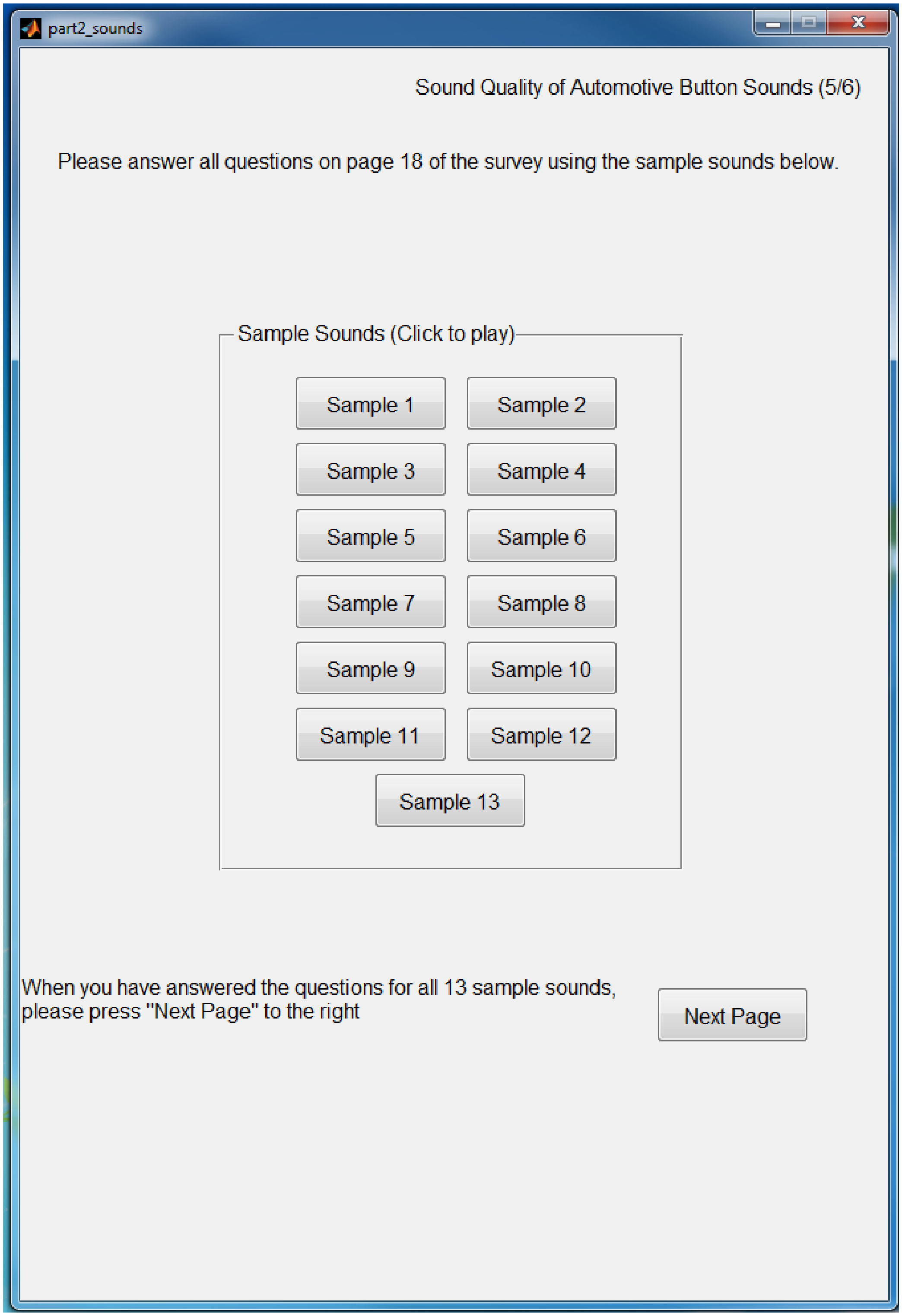



Figure 295. MATLAB GUI Part 2 Sounds

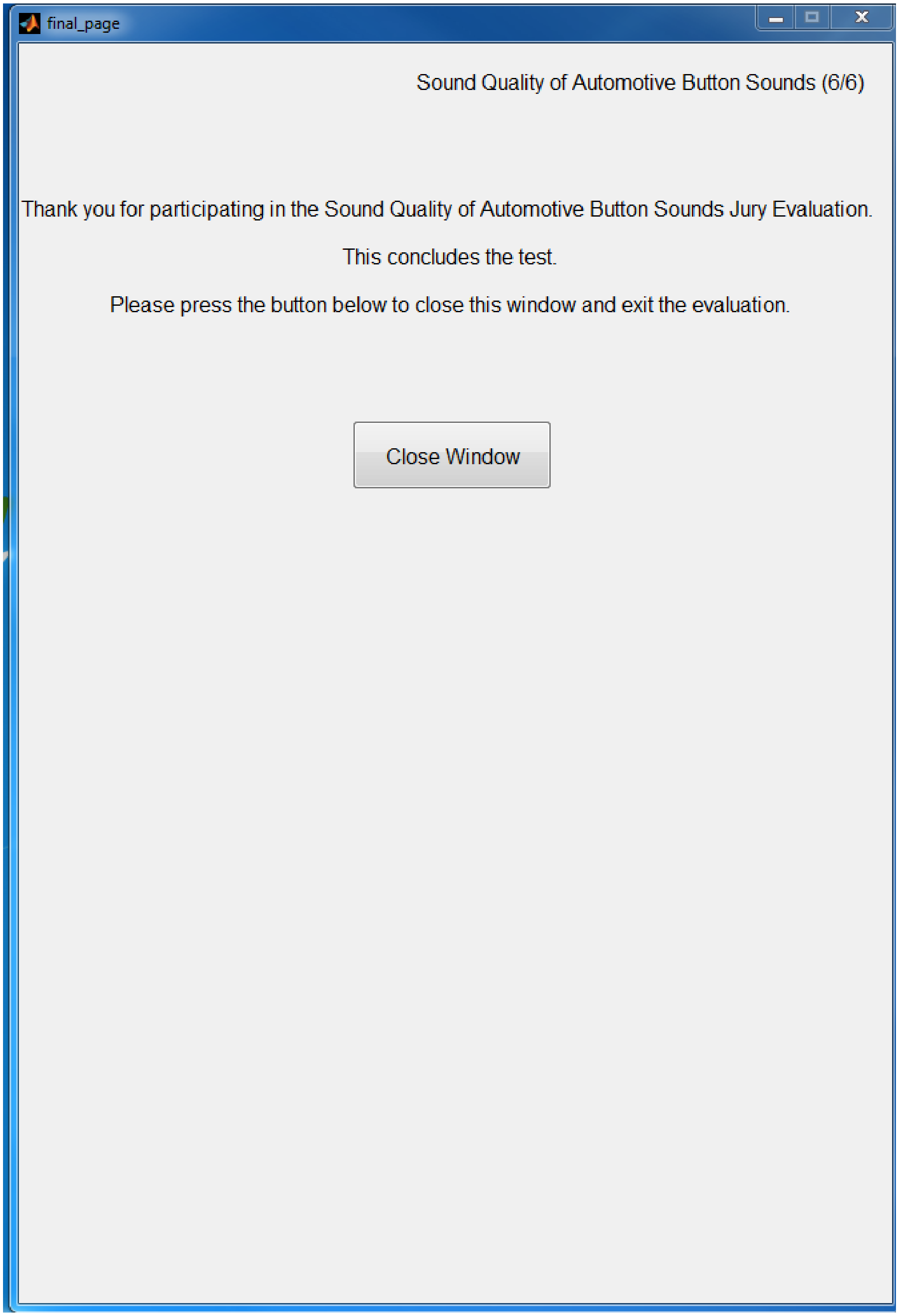



Figure 296. MATLAB GUI Final Page

# Appendix N. IRB Documents

## Appendix N.1. Grant

**Proposal for Steering-Wheel Mounted User Interface Audio Development**

Prof. Kenneth A. Cunefare
The Georgia Institute of Technology
August 22, 2013

**Objective:**
Develop system to generate required input waveform in order to generate a target output audio signal for steering-wheel mounted touch-controlled automotive user interface.

**Scope:**
The sponsor company has developed a steering-wheel mounted/integrated technology for touch-control of vehicle functions. The system has the capability of generating haptic feedback, vibratory and acoustic, to user interactions. The acoustic feedback component of the technology will be the focus of the work proposed here.

The work proposed here will consider what should be the digital waveform input into the audio reproduction element of the technology such that a desired audible signal is generated. Aspects of the work include consideration of just what the desired audible signal should be (e.g., a button "click") appropriate to the user-commanded function.

The basic outline of the work will take a transfer-function approach to represent the signal generation path from the waveform input to the audio output. Inversion of this transfer function representation will permit determination of the input waveform necessary for generation of a target output signal.

Limited subjective testing will be performed to evaluate the sound quality or suitability of generated output signals with respect to the desired audio cueing of a commanded function. Note that this will not be a completely open-ended sound quality assessment. That is, this work will not seek to find the "best" waveform matched to a given command function. Rather, among a set of limited audio cues, such as "clicks", "thumps" or similar such qualitative descriptors of sounds that reflect a command function analogous to a button-push, the work will seek to find a suitable waveform that clearly cues the commanded function.

Switch packs represent a current technology used to implement vehicle command functions; the sound that these switches produce represent a basis of comparison for the audio queues generated from the haptic feedback device. Measurement of the sound produced by exemplar switch packs will be included in the scope of the project.

**The sponsor company Obligations:**
The sponsor company will:

1) provide us functional steering-wheel mounted interface devices.
2) provide technical detail on the audio output capabilities of the interface device.
3) provide a matrix of commanded-functions to be implemented through the interface device, as well as an initial set of qualitative command-cue descriptors for each function (e.g., “button click”)
4) provide details or technical means to enable us to download waveforms into the interface device.
5) provide exemplar switch packs for comparative testing

**Georgia Tech Tasks:**
Task 1) Perform a search for ergonomic and psychoacoustic literature that address audio cueing of interface command functions.

Task 2) Measure the frequency-response of the interface device *(Question: does the technology have adaptive audio output capability? That is, does it automatically adjust the amplitude of the output based on the ambient noise environment, what some call “speed sensitive” volume control?)*.

Task 3) Implement in MATLAB or Simulink an inverse block-transfer-matrix representation for the frequency response of the system, such that desired output waveforms are mapped to required input waveforms.

Task 4) Measure generated output signal vs. desired output signal for a given input waveform. Refine model and inversion method as necessary to minimize error between actual and target waveforms.

Task 5) Perform subjective sound-quality tests of output sounds with respect to desired command cues. Refine inversion model and waveform input generator as appropriate to sound quality assessment.

Task 6) Measure output sounds of exemplar switch packs, compare to measured audio signature output obtained from the interface device, and classify switch sounds by device category.

**Deliverables**
We shall provide a technical report documenting the literature, frequency response measurements, transfer matrix modeling, and sound quality tests. We will provide the MATLAB or other source code necessary to map from desired output to required input waveforms. Reporting and documentation to be complete by 12 months after project initiation.

**Budget Assumptions:**
Because of the need to recruit and retain a graduate student for this effort, a minimum commitment of 12 months to that student is required.

The budget includes costs for incidental supplies related to the testing of the device.

Graduate student health insurance at 1.8% is charged against the graduate student stipend.
Graduate student tuition remission is charged at $1264/month.

Direct costs are burdened with 59.8% Indirect Costs (overhead).

**Budget:**

| A | Salaries and Wages | Unit Price | Basis | Units | |
|---|---|---|---|---|---|
| | Dr. Cunefare | $12,664 | Month | 0.3 | $3,799 |
| | Graduate Research Assistants | $2,070 | Month | 12 | $24,840 |
| | Graduate Student Health Benefits | 1.9% | | | $472 |
| C | Total Fringe Benefits | 28.5% | | | $1,555 |
| | | | | | |
| E | Materials and Supplies | $1,500 | | | $1,500 |
| | | | | | |
| H | Tuition Remission | $1,253 | $/Month/grad student | | $15,036 |
| | | | | | |
| K | Total Direct Cost (TDC) | | | | $46,730 |
| | | | | | |
| M | Indirect Costs (IDC) | 59.8% | | | $18,953 |
| | | | | | |
| N | Total Budget for Georgia Tech | | | | $65,683 |

## Appendix N.2. Abstract

Protocol Title: Sound Quality of Automotive Button Sounds

**Investigators:** Matt Edwards (medwards32@gatech.edu), Professor Ken Cunefare (ken.cunefare@me.gatech.edu).

**Abstract:**
Switch packs represent a current technology used to implement vehicle command functions. Research is conducted to evaluate the sound quality or suitability of generated output signals (i.e. "clicks", "thumps", etc.) with respect to the desired audio cueing of a commanded function (i.e. volume up, cruise off, etc.). In order for objective lab measurements to be correlated to human-perceived sound quality, a sound jury study must be performed. The study has jury subjects listen to recorded button sounds from various switches. The jury then matches these sounds with what they believe to be the ideal associated command functions. A sample size of 30 individuals will be used for the experiment.

## Appendix N.3. Confidentiality Statement

**Protocol Title:** Sound Quality of Automotive Button Sounds

**Investigators:** Matt Edwards (medwards32@gatech.edu), Professor Ken Cunefare (ken.cunefare@me.gatech.edu).

**Confidentiality Statement:**
Each set of data and questionnaire will be assigned a number with which no name is associated. The consent form is the only document containing the names of the subjects which is stored separately in a locked drawer in the office.

## Appendix N.4. Inclusion/Exclusion Criteria:

**Protocol Title:** Sound Quality of Automotive Button Sounds

**Investigators:** Matt Edwards (medwards32@gatech.edu), Professor Ken Cunefare (ken.cunefare@me.gatech.edu).

**Inclusion/Exclusion Criteria:**
Subjects will be recruited from Georgia Institute of Technology student population. One group of people will be explicitly excluded in the study: subjects with "non-normal" hearing. The screening audiogram quantifies a subjects hearing loss in terms of dB hearing level, or dBHL across the frequency range of human hearing. Normal hearing is classified as being between -10dBHL and 15dBHL. A subject who falls outside of this envelope will not qualify for the study. They will still receive a $5 gift card for participation up to this point. The audiogram is an automated system, and the determination of normal hearing is 100% objective. The audiogram is administered by co-investigator Matt Edwards.

Several groups are potentially vulnerable. Economically disadvantaged group may be at a disadvantage since this research would require the use of a computer which may not be available to them. However, the disadvantage is insignificant since the tasks to be performed are based more on eye-hand coordination and tactile sensation than on computer knowledge. Given that the subjects will be recruited from Georgia Institute of Technology, it is safe to assume those who do not own a computer can easily access one on-campus. Non-native English speakers may be at a disadvantage since the instructions and paperwork used during the study will be in English. However, it is safe to assume that students admitted into Georgia Institute of Technology at least understands English as a second language, in which case the investigator will instruct the subjects thoughtfully. Certain illness that affects an individual's hearing may put him or her at a disadvantage, in which case he or she is not eligible for this study. We do not expect to have pregnant women participating in this study. However they are certainly eligible as our laboratory provides ample seating. Since students will be used for this study, there is a chance that he or she may have studied under the investigator. This may produce a psychological effect that may affect performance.

## Appendix N.5. Lay Summary

**Protocol Title:** Sound Quality of Automotive Button Sounds

**Investigators:** Matt Edwards (medwards32@gatech.edu), Professor Ken Cunefare (ken.cunefare@me.gatech.edu).

**Lay Summary**

The automotive industry uses switch packs, such as those located on modern car steering wheels, to initiate vehicle command functions (e.g. volume control and cruise control). However, little research has been conducted to identify which switch sounds should be associated with which command functions to illicit high customer satisfaction. A lower perception of quality can result if the ideal audio cue is not matched with the proper command function. Therefore, if the desired audio cue/command function pairs can be identified and implemented, the vehicle will offer additional value to the customer.

## Appendix N.6. Protocol Description

**Protocol Title:** Sound Quality of Automotive Button Sounds

**Investigators:** Matt Edwards (medwards32@gatech.edu), Professor Ken Cunefare (ken.cunefare@me.gatech.edu).

**Protocol Description:**
First the subject will be asked to fill out a short questionnaire. The questionnaire will ask about the subject's age, gender, impairments relevant to their hearing, and their experience with this family of products. Then he or she will be introduced to the listening hardware (headphones, amplifier, etc.) involved in the study. These devices are similar to hardware found in home audio equipment, both in form and function. The user will wear the headphones, while listening to recordings played over the equipment.

The subject will first undergo an audiogram, which is simply a standard hearing test. The audiogram shows whether the subject has "normal" hearing or not. The screening audiogram is an automated system. The audiogram quantifies a subjects hearing loss in terms of dB hearing level, or dBHL across the frequency range of human hearing. Normal hearing is classified as being between -10dBHL and 15dBHL. A subject who falls outside of this envelope will not qualify for the study. This determination is 100% objective. After the qualifying audiogram, the experiment begins.

The subject will listen to a series of recorded sounds, and rate their relative characteristics. For example, a subject will be asked to rate the pitch of each sound on a 5-point subjective scale ranging from "Not at all" to "Extremely". Subjects will also have the option of selecting "Don't Know" if they feel they do not have sufficient understanding of the metric being asked of them to provide a confident answer. The procedure will be repeated for other sound descriptors, such as tonality, loudness, strength, etc.

The second task involves listening to the same sounds and matching each with the command function that is best associated with it. For example, a subject will be asked whether a certain sound is appropriate for the "volume up" function, the "cruise on" function, etc. The pairs will not be mutually exclusive (i.e. a subject will be free to assign one sound to multiple command functions if he/she so chooses).

## Appendix N.7. Recruitment

**Protocol Title:** Sound Quality of Automotive Button Sounds

**Investigators:** Matt Edwards (medwards32@gatech.edu), Professor Ken Cunefare (ken.cunefare@me.gatech.edu).

**Recruitment:**
The recruitment plan involves both word of mouth and ads. Flyers will be distributed in, before, and after classes in Georgia Institute of Technology. A copy of the flyer is attached. From there the study will spread by the word of mouth. Sign up is on first come first serve basis via e-mail.

Each subject will be offered a US$10 gift card for Barnes and Nobles bookstore for completing the study. Should they choose to leave in the middle of the study for any reason or if they are screened by the audiogram, they will receive a US$5 gift card for Barnes and Nobles bookstore.

**Verbal Recruitment Script:**
We are conducting a sound jury to evaluate the sound quality automotive button sounds. As a result, we are recruiting students to participate in this sound jury. To participate in the study, you will first take a qualifying audiogram, which is standard hearing test. The test is automatically administered and the results are pass/fail. Afterwards, you will listen and subjectively rate the sound quality of button sounds. All of the audiogram and sound jury testing will be performed by a computer over headphones. A $10 gift card to Barnes and Noble bookstore will be provided for your participation. If you participate, but are unable to complete the survey for any reason, you will receive a $5 gift card to Barnes and Noble bookstore. The testing will take approximately one hour of your time.

**Appendix N.8. Advertisement**

# Students Needed!
# Psychoacoustics Study

# Participate in a Sound Jury

Who: Any GT Student
What: Psychoacoustics Sound Jury
When: By Appointment (approx. 1 hr)

$10 Gift Card for Completing Study

Contact:
medwards32@gatech.edu

## Appendix N.9. Risk/Benefit Statement

**Protocol Title:** Sound Quality of Automotive Button Sounds

**Investigators:** Matt Edwards (medwards32@gatech.edu), Professor Ken Cunefare (ken.cunefare@me.gatech.edu).

**Risk/Benefit Statement:**
The risks involved are no greater than those involved in daily activities such as listening to music over headphones at a comfortable level. Safety is insured in this experiment.
The audio hardware is a commercially available off the shelf device used extensively in industry. All recordings will be played at a level safe for listening.

The subjects are not likely to benefit in any way from joining this study.

**Appendix N.10. Scientific Methodology**

**Protocol Title:** Sound Quality of Automotive Button Sounds

**Investigators:** Matt Edwards (medwards32@gatech.edu), Professor Ken Cunefare (ken.cunefare@me.gatech.edu).

**Scientific Methodology:**
Attempts have been made this to make this study as scientifically objective as possible. In all tasks, there will be a training period consisting of a dry run, so that the subject becomes comfortable with the protocol and ask questions. The subject will learn primarily by doing it; input from the investigator will be kept at a minimum except in cases where device operation itself and safety is concerned. Each run of the task will be similar but not identical as noted below.

The audiogram ensures that subjects are within the “normal” limits for human hearing. This screening will ensure that results are not skewed by erroneous data stemming from hearing loss. In the first task, subjects will listen to a series of sounds and rate them. The sounds will be rated based on how the sound fits adjective descriptors, such as loud, tonal, dynamic, pitch, etc. In order to quantify the result, a 5-point rating system ranging from “Not at all” to “Extremely” allows subjects to assign a degree to each one. Subjects will also be given the option of answering “Don’t Know” if they are not confident they understand the meaning of the metric being asked of them enough to provide a proper answer.

The second task involves listening to the same sounds and matching each with the command function that is best associated with it. For example, a subject will be asked whether a certain sound is appropriate for the “volume up” function, the “cruise on” function, etc. The pairs will not be mutually exclusive (i.e. a subject will be free to assign one sound to multiple command functions if he/she so chooses).

After the human testing is completed, the responses will be compared with each other. Statistics will be used to correlate the sound/command function pairs with the subjective perceptions obtained in the human study.

# Appendix O. MATLAB Code

## Appendix O.1. MATLAB Code – GUI

### Appendix O.1.1. Title Page

```
function varargout = title_page(varargin)
% TITLE_PAGE MATLAB code for title_page.fig
%      TITLE_PAGE, by itself, creates a new TITLE_PAGE or raises the
existing
%      singleton*.
%
%      H = TITLE_PAGE returns the handle to a new TITLE_PAGE or the
handle to
%      the existing singleton*.
%
%      TITLE_PAGE('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in TITLE_PAGE.M with the given input
arguments.
%
%      TITLE_PAGE('Property','Value',...) creates a new TITLE_PAGE or
raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before title_page_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to title_page_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help title_page

% Last Modified by GUIDE v2.5 02-Sep-2014 17:29:50

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @title_page_OpeningFcn, ...
                   'gui_OutputFcn',  @title_page_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before title_page is made visible.
function title_page_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to title_page (see VARARGIN)

% Choose default command line output for title_page
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes title_page wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = title_page_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in begin_test1.
function begin_test1_Callback(hObject, eventdata, handles)
% hObject    handle to begin_test1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
part1_training
```

Appendix O.1.2. Part 1 Training

```
function varargout = part1_training(varargin)
% PART1_TRAINING MATLAB code for part1_training.fig
%      PART1_TRAINING, by itself, creates a new PART1_TRAINING or
raises the existing
%      singleton*.
%
%      H = PART1_TRAINING returns the handle to a new PART1_TRAINING or
the handle to
%      the existing singleton*.
%
%      PART1_TRAINING('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in PART1_TRAINING.M with the given input
arguments.
%
%      PART1_TRAINING('Property','Value',...) creates a new
PART1_TRAINING or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before part1_training_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to part1_training_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help part1_training

% Last Modified by GUIDE v2.5 02-Sep-2014 17:11:05

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @part1_training_OpeningFcn, ...
                   'gui_OutputFcn',  @part1_training_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
% End initialization code - DO NOT EDIT


% --- Executes just before part1_training is made visible.
function part1_training_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to part1_training (see VARARGIN)

close title_page

% Choose default command line output for part1_training
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes part1_training wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = part1_training_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in begin_test_2.
function begin_test_2_Callback(hObject, eventdata, handles)
% hObject    handle to begin_test_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
part1_sounds

% --- Executes on button press in pitch_2.
function pitch_2_Callback(hObject, eventdata, handles)
% hObject    handle to pitch_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in naturalness_2.
function naturalness_2_Callback(hObject, eventdata, handles)
% hObject    handle to naturalness_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% --- Executes on button press in duration_2.
function duration_2_Callback(hObject, eventdata, handles)
% hObject    handle to duration_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in loudness_2.
function loudness_2_Callback(hObject, eventdata, handles)
% hObject    handle to loudness_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in quaility_2.
function quaility_2_Callback(hObject, eventdata, handles)
% hObject    handle to quaility_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Appendix O.1.3. Part 1 Sounds

```
function varargout = part1_sounds(varargin)
% PART1_SOUNDS MATLAB code for part1_sounds.fig
%      PART1_SOUNDS, by itself, creates a new PART1_SOUNDS or raises
the existing
%      singleton*.
%
%      H = PART1_SOUNDS returns the handle to a new PART1_SOUNDS or the
handle to
%      the existing singleton*.
%
%      PART1_SOUNDS('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in PART1_SOUNDS.M with the given input
arguments.
%
%      PART1_SOUNDS('Property','Value',...) creates a new PART1_SOUNDS
or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before part1_sounds_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to part1_sounds_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
%      Instead of changing the "master_folder" variable, you need to
change
%      the "handles.filepath" variable, located on line 58

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help part1_sounds

% Last Modified by GUIDE v2.5 07-Oct-2014 16:28:13

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @part1_sounds_OpeningFcn, ...
                   'gui_OutputFcn',  @part1_sounds_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before part1_sounds is made visible.
function part1_sounds_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to part1_sounds (see VARARGIN)

close part1_training
handles.filepath = 'C:\Users\Kenneth A. Cunefare\Documents\Test
GUI\sample sounds\';

% Choose default command line output for part1_sounds
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes part1_sounds wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = part1_sounds_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in next_page3.
function next_page3_Callback(hObject, eventdata, handles)
% hObject    handle to next_page3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
part2_training


% --- Executes on button press in sample1_3.
function sample1_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample1_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample1.wav']);
sound(y,fs);
```

```
% --- Executes on button press in sample2_3.
function sample2_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample2_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample2.wav']);
sound(y,fs);

% --- Executes on button press in sample13_3.
function sample13_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample13_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample13.wav']);
sound(y,fs);

% --- Executes on button press in sample12_3.
function sample12_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample12_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample12.wav']);
sound(y,fs);

% --- Executes on button press in sample11_3.
function sample11_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample11_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample11.wav']);
sound(y,fs);

% --- Executes on button press in sample10_3.
function sample10_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample10_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample10.wav']);
sound(y,fs);

% --- Executes on button press in sample9_3.
function sample9_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample9_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample9.wav']);
sound(y,fs);

% --- Executes on button press in sample8_3.
function sample8_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample8_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample8.wav']);
```

```
sound(y,fs);

% --- Executes on button press in sample7_3.
function sample7_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample7_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample7.wav']);
sound(y,fs);

% --- Executes on button press in sample6_3.
function sample6_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample6_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample6.wav']);
sound(y,fs);

% --- Executes on button press in sample5_3.
function sample5_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample5_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample5.wav']);
sound(y,fs);

% --- Executes on button press in sample4_3.
function sample4_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample4_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample4.wav']);
sound(y,fs);

% --- Executes on button press in sample3_3.
function sample3_3_Callback(hObject, eventdata, handles)
% hObject    handle to sample3_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample3.wav']);
sound(y,fs);
```

Appendix O.1.4. Part 2 Training

```
function varargout = part2_training(varargin)
% PART2_TRAINING MATLAB code for part2_training.fig
%      PART2_TRAINING, by itself, creates a new PART2_TRAINING or
raises the existing
%      singleton*.
%
%      H = PART2_TRAINING returns the handle to a new PART2_TRAINING or
the handle to
%      the existing singleton*.
%
%      PART2_TRAINING('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in PART2_TRAINING.M with the given input
arguments.
%
%      PART2_TRAINING('Property','Value',...) creates a new
PART2_TRAINING or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before part2_training_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to part2_training_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help part2_training

% Last Modified by GUIDE v2.5 02-Sep-2014 17:18:47

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @part2_training_OpeningFcn, ...
                   'gui_OutputFcn',  @part2_training_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
% End initialization code - DO NOT EDIT


% --- Executes just before part2_training is made visible.
function part2_training_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to part2_training (see VARARGIN)

close part1_sounds

% Choose default command line output for part2_training
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes part2_training wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = part2_training_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in begin_test4.
function begin_test4_Callback(hObject, eventdata, handles)
% hObject    handle to begin_test4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
part2_sounds
```

Appendix O.1.5. Part 2 Sounds

```
function varargout = part2_sounds(varargin)
% PART2_SOUNDS MATLAB code for part2_sounds.fig
%      PART2_SOUNDS, by itself, creates a new PART2_SOUNDS or raises
the existing
%      singleton*.
%
%      H = PART2_SOUNDS returns the handle to a new PART2_SOUNDS or the
handle to
%      the existing singleton*.
%
%      PART2_SOUNDS('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in PART2_SOUNDS.M with the given input
arguments.
%
%      PART2_SOUNDS('Property','Value',...) creates a new PART2_SOUNDS
or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before part2_sounds_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to part2_sounds_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
%      Instead of changing the "master_folder" variable, you need to
change
%      the "handles.filepath" variable, located on line 58

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help part2_sounds

% Last Modified by GUIDE v2.5 07-Oct-2014 16:33:53

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @part2_sounds_OpeningFcn, ...
                   'gui_OutputFcn',  @part2_sounds_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before part2_sounds is made visible.
function part2_sounds_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to part2_sounds (see VARARGIN)

close part2_training
handles.filepath = 'C:\Users\Kenneth A. Cunefare\Documents\Test
GUI\sample sounds\';

% Choose default command line output for part2_sounds
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes part2_sounds wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = part2_sounds_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in next_page_5.
function next_page_5_Callback(hObject, eventdata, handles)
% hObject    handle to next_page_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
final_page


% --- Executes on button press in sample1_5.
function sample1_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample1_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample1.wav']);
sound(y,fs);
```

```
% --- Executes on button press in sample2_5.
function sample2_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample2_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample2.wav']);
sound(y,fs);

% --- Executes on button press in sample13_5.
function sample13_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample13_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample13.wav']);
sound(y,fs);

% --- Executes on button press in sample12_5.
function sample12_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample12_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample12.wav']);
sound(y,fs);

% --- Executes on button press in sample11_5.
function sample11_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample11_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample11.wav']);
sound(y,fs);

% --- Executes on button press in sample10_5.
function sample10_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample10_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample10.wav']);
sound(y,fs);

% --- Executes on button press in sample9_5.
function sample9_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample9_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample9.wav']);
sound(y,fs);

% --- Executes on button press in sample8_5.
function sample8_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample8_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample8.wav']);
```

```
sound(y,fs);

% --- Executes on button press in sample7_5.
function sample7_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample7_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample7.wav']);
sound(y,fs);

% --- Executes on button press in sample6_5.
function sample6_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample6_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample6.wav']);
sound(y,fs);

% --- Executes on button press in sample5_5.
function sample5_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample5_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample5.wav']);
sound(y,fs);

% --- Executes on button press in sample4_5.
function sample4_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample4_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample4.wav']);
sound(y,fs);

% --- Executes on button press in sample3_5.
function sample3_5_Callback(hObject, eventdata, handles)
% hObject    handle to sample3_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[y,fs] = audioread([handles.filepath 'sample3.wav']);
sound(y,fs);
```

Appendix O.1.6. Final Page

```
function varargout = final_page(varargin)
% FINAL_PAGE MATLAB code for final_page.fig
%      FINAL_PAGE, by itself, creates a new FINAL_PAGE or raises the
existing
%      singleton*.
%
%      H = FINAL_PAGE returns the handle to a new FINAL_PAGE or the
handle to
%      the existing singleton*.
%
%      FINAL_PAGE('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in FINAL_PAGE.M with the given input
arguments.
%
%      FINAL_PAGE('Property','Value',...) creates a new FINAL_PAGE or
raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before final_page_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to final_page_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help final_page

% Last Modified by GUIDE v2.5 02-Sep-2014 17:27:28

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @final_page_OpeningFcn, ...
                   'gui_OutputFcn',  @final_page_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before final_page is made visible.
function final_page_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to final_page (see VARARGIN)

close part2_sounds

% Choose default command line output for final_page
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes final_page wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = final_page_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in close_window_6.
function close_window_6_Callback(hObject, eventdata, handles)
% hObject    handle to close_window_6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close final_page
```

## Appendix O.2. MATLAB Code: Sine Sweep wav File Generation

```
%% Matt Edwards | Sine Sweep Wav File Generator | Last Update: 02/06/15

%*********************************************************************
****
% - The following code creates a sine sweep that is t_end seconds long,
at
%   sampling frequency fs, with N bits per sample.
% - The sine wave will sweep from f_min to f_max (in Hz). Keep in mind
that
%   f_max should not exceed half the sampling frequency fs.
% - If you wish to be able to write this wav file to a blob file in
order
%   to load it to the haptic board, then t_end cannot exceed
%   0.34. This is because the blob file must be under 8192 bits.
%*********************************************************************
****

t_end = 0.34;
fs = 11025;
f_min = 20;
f_max = fs/2;
N = 16;

t = [0:1/fs:t_end];
t = t(1:end-1);
t=t';

y = chirp(t,f_min,t_end,f_max);
y = y./max(abs(y));

sound(y,fs,N)
audiowrite(['sine_sweep_' num2str(fs) '.wav'],y,fs,'BitsPerSample',N)
```

## Appendix O.3. MATLAB Function: Figure Cascade

```
function cascade
%CASCADE Cascade existing figures so that they don't directly overlap.
%   CASCADE takes and returns no arguments.  This function will cascade
as
%   many figures as will fit the height/width of the screen.  If there
are
%   more figures than can cascade in a screen, those additional figures
are
%   left in their original position.
%
%   Author: Isaac Noh
%   Copyright The MathWorks, Inc.
%   November 2007

% Find Existing Figures
figs = findobj(0,'Type','figure');
figs = sort(figs);

% Size of Entire Screen
ss = get(0,'ScreenSize');

units = get(figs,'Units');
set(figs,'Units','pixels')

for n = 1:length(figs)
    pos = get(figs(n),'Position');
    if n == 1
        bot = ss(4) - pos(4) - 70;
        set(figs(n),'Position',[0 bot pos(3:4)]);
    else
        pPos = get(figs(n-1),'Position');
        left = pPos(1) + 20;
        bot = pPos(2) - 20;
        if ((left + pos(3)) > ss(3)) || (bot < 0)
            break
        end
        set(figs(n),'Position',[left bot  pos(3:4)]);
    end
    figure(figs(n));
end
set(figs,{'Units'},units);
```

## Appendix O.4. MATLAB Code: CSV-to-Mat Raw Data Converter

```
%% Matt Edwards | CSV-to-Mat Raw Data Converter | Last Update: 7/24/14

%**********************************************************************
****
%   - The following script is designed to take the raw data from an
%     experiment in .csv text format and convert it to .mat binary
format.
%   - When prompted, load the .csv file corresponding to the experiment
you
%     wish to analyze
%   - Each .csv file may only correspond to ONE type of signal form ONE
%     pack. (i.e. One .csv file would be for "Pack 3 Audio", or "Pack 9
%     Tactile", but not "Packs 5 & 6 Audio" or "Pack 1 Audio &
Tactile").
%   - The imported file must be in a VERY specific format:
%     - The data must have been copy/pasted directly from NI
SignalExpress
%       into Excel.
%     - The raw data and "delta x" entries must NOT have been edited.
Other
%       header entries may have been edited (e.g. the title of each
header
%       could have been changed to designate which pack & switch number
%       that data belongs to).
%     - Each data set must consist of 3 columns: "time", "amplitude",
and
%       "comment". There must be NO empty columns between data sets
(Note
%       that the "comment" column may appear as an empty column,
because
%       the word "comment" is the only entry in the entire column. This
%       must stay like that).
%     - The first data set MUST be the noise data.
%     - The following data sets (from left to right) must be of the
%       following pattern: Switch 1 push, Switch 1 pull, Switch 2 push,
%       Switch 2 pull, etc.
%     - An example of the correct format can be found in the Switch
Pack
%       MATLAB Workspace folder.
%   - When importing the data, be sure to import as a MATRIX, and be
sure
%     to rename the array "raw_data".
%**********************************************************************
****

clear all
close all
clc

disp(' ')
disp('Load the excel file containing the desired experimental raw data:
')
disp(' ')
```

```
disp('Be sure to load the data as a MATRIX')
disp('Be sure to name the matrix "raw_data"')
uiimport('-file');

% identify which pack is being imported
name_pack = input('Please enter the Pack # you are analyzing: ');
if name_pack > 11 || name_pack < 1
    error('Please enter a Pack # between 1 and 11')
end

% build the name array
name_type = input('Is this audio or tactile data? [a/t] ','s');
if name_type == 'a'
    name_type = 'Audio';
    if name_pack == 1
        name_pack = 'Pack1_';
        c = {'P1S1pushaudio'; 'P1S1pullaudio'; 'P1S2pushaudio';
'P1S2pullaudio'; 'P1S3pushaudio'; 'P1S3pullaudio'; 'P1S4pushaudio';
'P1S4pullaudio'; 'P1S5pushaudio'; 'P1S5pullaudio'; 'P1S6pushaudio';
'P1S6pullaudio'; 'P1S7pushaudio'; 'P1S7pullaudio'};
    elseif name_pack == 2
        name_pack = 'Pack2_';
        c = {'P2S1pushaudio'; 'P2S1pullaudio'; 'P2S2pushaudio';
'P2S2pullaudio'; 'P2S3pushaudio'; 'P2S3pullaudio'; 'P2S4pushaudio';
'P2S4pullaudio'; 'P2S5pushaudio'; 'P2S5pullaudio'};
    elseif name_pack == 3
        name_pack = 'Pack3_';
        c = {'P3S1pushaudio'; 'P3S1pullaudio'; 'P3S2pushaudio';
'P3S2pullaudio'; 'P3S3pushaudio'; 'P3S3pullaudio'; 'P3S4pushaudio';
'P3S4pullaudio'; 'P3S5pushaudio'; 'P3S5pullaudio'};
    elseif name_pack == 4
        name_pack = 'Pack4_';
        c = {'P4S1pushaudio'; 'P4S1pullaudio'; 'P4S2pushaudio';
'P4S2pullaudio'; 'P4S3pushaudio'; 'P4S3pullaudio'; 'P4S4pushaudio';
'P4S4pullaudio'};
    elseif name_pack == 5
        name_pack = 'Pack5_';
        c = {'P5S1pushaudio'; 'P5S1pullaudio'; 'P5S2pushaudio';
'P5S2pullaudio'; 'P5S3pushaudio'; 'P5S3pullaudio'; 'P5S4pushaudio';
'P5S4pullaudio'; 'P5S5pushaudio'; 'P5S5pullaudio'; 'P5S6pushaudio';
'P5S6pullaudio'; 'P5S7pushaudio'; 'P5S7pullaudio'};
     elseif name_pack == 6
        name_pack = 'Pack6_';
        c = {'P6S1pushaudio'; 'P6S1pullaudio'; 'P6S2pushaudio';
'P6S2pullaudio'; 'P6S3pushaudio'; 'P6S3pullaudio'; 'P6S4pushaudio';
'P6S4pullaudio'};
    elseif name_pack == 7
        name_pack = 'Pack7_';
        c = {'P7S1pushaudio'; 'P7S1pullaudio'; 'P7S2pushaudio';
'P7S2pullaudio'; 'P7S3pushaudio'; 'P7S3pullaudio'; 'P7S4pushaudio';
'P7S4pullaudio'; 'P7S5pushaudio'; 'P7S5pullaudio'; 'P7S6pushaudio';
'P7S6pullaudio'};
    elseif name_pack == 8
        name_pack = 'Pack8_';
```

```
        c = {'P8S1pushaudio'; 'P8S1pullaudio'; 'P8S2pushaudio';
'P8S2pullaudio'; 'P8S3pushaudio'; 'P8S3pullaudio'; 'P8S4pushaudio';
'P8S4pullaudio'; 'P8S5pushaudio'; 'P8S5pullaudio'; 'P8S6pushaudio';
'P8S6pullaudio'};
    elseif name_pack == 9
        name_pack = 'Pack9_';
        c = {'P9S1pushaudio'; 'P9S1pullaudio'; 'P9S2pushaudio';
'P9S2pullaudio'; 'P9S3pushaudio'; 'P9S3pullaudio'; 'P9S4pushaudio';
'P9S4pullaudio'; 'P9S5pushaudio'; 'P9S5pullaudio'};
    elseif name_pack == 10
        name_pack = 'Pack10_';
        c = {'P10S1pushaudio'; 'P10S1pullaudio'; 'P10S2pushaudio';
'P10S2pullaudio'; 'P10S3pushaudio'; 'P10S3pullaudio'; 'P10S4pushaudio';
'P10S4pullaudio'; 'P10S5pushaudio'; 'P10S5pullaudio'; 'P10S6pushaudio';
'P10S6pullaudio'; 'P10S7pushaudio'; 'P10S7pullaudio'; 'P10S8pushaudio';
'P10S8pullaudio'};
    elseif name_pack == 11
        name_pack = 'Pack11_';
        c = {'P11S1pushaudio'; 'P11S1pullaudio'; 'P11S2pushaudio';
'P11S2pullaudio'; 'P11S3pushaudio'; 'P11S3pullaudio'; 'P11S4pushaudio';
'P11S4pullaudio'; 'P11S5pushaudio'; 'P11S5pullaudio'; 'P11S6pushaudio';
'P11S6pullaudio'; 'P11S7pushaudio'; 'P11S7pullaudio'; 'P11S8pushaudio';
'P11S8pullaudio'; 'P11S9pushaudio'; 'P11S9pullaudio'};
    end
elseif name_type == 't'
    name_type = 'Tactile';
    if name_pack == 1
        name_pack = 'Pack1_';
        c = {'P1S1pushtactile'; 'P1S1pulltactile'; 'P1S2pushtactile';
'P1S2pulltactile'; 'P1S3pushtactile'; 'P1S3pulltactile';
'P1S4pushtactile'; 'P1S4pulltactile'; 'P1S5pushtactile';
'P1S5pulltactile'; 'P1S6pushtactile'; 'P1S6pulltactile';
'P1S7pushtactile'; 'P1S7pulltactile'};
    elseif name_pack == 2
        name_pack = 'Pack2_';
        c = {'P2S1pushtactile'; 'P2S1pulltactile'; 'P2S2pushtactile';
'P2S2pulltactile'; 'P2S3pushtactile'; 'P2S3pulltactile';
'P2S4pushtactile'; 'P2S4pulltactile'; 'P2S5pushtactile';
'P2S5pulltactile'};
    elseif name_pack == 3
        name_pack = 'Pack3_';
        c = {'P3S1pushtactile'; 'P3S1pulltactile'; 'P3S2pushtactile';
'P3S2pulltactile'; 'P3S3pushtactile'; 'P3S3pulltactile';
'P3S4pushtactile'; 'P3S4pulltactile'; 'P3S5pushtactile';
'P3S5pulltactile'};
    elseif name_pack == 4
        name_pack = 'Pack4_';
        c = {'P4S1pushtactile'; 'P4S1pulltactile'; 'P4S2pushtactile';
'P4S2pulltactile'; 'P4S3pushtactile'; 'P4S3pulltactile';
'P4S4pushtactile'; 'P4S4pulltactile'};
    elseif name_pack == 5
        name_pack = 'Pack5_';
        c = {'P5S1pushtactile'; 'P5S1pulltactile'; 'P5S2pushtactile';
'P5S2pulltactile'; 'P5S3pushtactile'; 'P5S3pulltactile';
'P5S4pushtactile'; 'P5S4pulltactile'; 'P5S5pushtactile';
```

```
'P5S5pulltactile'; 'P5S6pushtactile'; 'P5S6pulltactile';
'P5S7pushtactile'; 'P5S7pulltactile'};
     elseif name_pack == 6
        name_pack = 'Pack6_';
        c = {'P6S1pushtactile'; 'P6S1pulltactile'; 'P6S2pushtactile';
'P6S2pulltactile'; 'P6S3pushtactile'; 'P6S3pulltactile';
'P6S4pushtactile'; 'P6S4pulltactile'};
    elseif name_pack == 7
        name_pack = 'Pack7_';
        c = {'P7S1pushtactile'; 'P7S1pulltactile'; 'P7S2pushtactile';
'P7S2pulltactile'; 'P7S3pushtactile'; 'P7S3pulltactile';
'P7S4pushtactile'; 'P7S4pulltactile'; 'P7S5pushtactile';
'P7S5pulltactile'; 'P7S6pushtactile'; 'P7S6pulltactile'};
    elseif name_pack == 8
        name_pack = 'Pack8_';
        c = {'P8S1pushtactile'; 'P8S1pulltactile'; 'P8S2pushtactile';
'P8S2pulltactile'; 'P8S3pushtactile'; 'P8S3pulltactile';
'P8S4pushtactile'; 'P8S4pulltactile'; 'P8S5pushtactile';
'P8S5pulltactile'; 'P8S6pushtactile'; 'P8S6pulltactile'};
    elseif name_pack == 9
        name_pack = 'Pack9_';
        c = {'P9S1pushtactile'; 'P9S1pulltactile'; 'P9S2pushtactile';
'P9S2pulltactile'; 'P9S3pushtactile'; 'P9S3pulltactile';
'P9S4pushtactile'; 'P9S4pulltactile'; 'P9S5pushtactile';
'P9S5pulltactile'};
    elseif name_pack == 10
        name_pack = 'Pack10_';
        c = {'P10S1pushtactile'; 'P10S1pulltactile';
'P10S2pushtactile'; 'P10S2pulltactile'; 'P10S3pushtactile';
'P10S3pulltactile'; 'P10S4pushtactile'; 'P10S4pulltactile';
'P10S5pushtactile'; 'P10S5pulltactile'; 'P10S6pushtactile';
'P10S6pulltactile'; 'P10S7pushtactile'; 'P10S7pulltactile';
'P10S8pushtactile'; 'P10S8pulltactile'};
    elseif name_pack == 11
        name_pack = 'Pack11_';
        c = {'P11S1pushtactile'; 'P11S1pulltactile';
'P11S2pushtactile'; 'P11S2pulltactile'; 'P11S3pushtactile';
'P11S3pulltactile'; 'P11S4pushtactile'; 'P11S4pulltactile';
'P11S5pushtactile'; 'P11S5pulltactile'; 'P11S6pushtactile';
'P11S6pulltactile'; 'P11S7pushtactile'; 'P11S7pulltactile';
'P11S8pushtactile'; 'P11S8pulltactile'; 'P11S9pushtactile';
'P11S9pulltactile'};
    end
else
    error('Please enter either "a" or "t" for audio or tactile data')
end

% make sure the data is ordered correctly
check_order = input('Is the data in your imported excel file in the
correct format? [y/n] ','s');
if check_order == 'n'
    img = imread('CSV format example.png');
    name_title = 'CSV Format Example';
    set(figure,'name',name_title,'numbertitle','off')
    image(img)
```

```
    axis off
    error('Please reorder your excel file correctly before continuing.
Use this example image as a guide.')
end

[R,C] = size(raw_data);
T = raw_data(1,2);
FS = 100*ceil(T^-1/100);
name_FS = ['_FS-' num2str(FS)];
t = raw_data(4:end,1);
noise = raw_data(4:end,2);

S = 5:3:C;
All_Data = zeros(R-3,length(S));
for i = 1:((C-3)/3)
    signal = raw_data(4:end,S(i));
    %eval(sprintf('A%d = signal', i));
    All_Data(:,i) = signal;
end

str = [name_pack name_type name_FS];
save(str,'All_Data','c','t','noise');

clear all
```

## Appendix O.5. MATLAB Code: Switch Pack Data Import

```
%% Matt Edwards | CSV-to-Mat Raw Data Converter | Last Update: 7/24/14

%***********************************************************************
****
%   - The following script is designed to take the raw data from an
%     experiment in .csv text format and convert it to .mat binary
format.
%   - When prompted, load the .csv file corresponding to the experiment
you
%     wish to analyze
%   - Each .csv file may only correspond to ONE type of signal form ONE
%     pack. (i.e. One .csv file would be for "Pack 3 Audio", or "Pack 9
%     Tactile", but not "Packs 5 & 6 Audio" or "Pack 1 Audio &
Tactile").
%   - The imported file must be in a VERY specific format:
%     - The data must have been copy/pasted directly from NI
SignalExpress
%       into Excel.
%     - The raw data and "delta x" entries must NOT have been edited.
Other
%       header entries may have been edited (e.g. the title of each
header
%       could have been changed to designate which pack & switch number
%       that data belongs to).
%     - Each data set must consist of 3 columns: "time", "amplitude",
and
%       "comment". There must be NO empty columns between data sets
(Note
%       that the "comment" column may appear as an empty column,
because
%       the word "comment" is the only entry in the entire column. This
%       must stay like that).
%     - The first data set MUST be the noise data.
%     - The following data sets (from left to right) must be of the
%       following pattern: Switch 1 push, Switch 1 pull, Switch 2 push,
%       Switch 2 pull, etc.
%     - An example of the correct format can be found in the Switch
Pack
%       MATLAB Workspace folder.
%   - When importing the data, be sure to import as a MATRIX, and be
sure
%     to rename the array "raw_data".
%***********************************************************************
****

clear all
close all
clc

disp(' ')
disp('Load the excel file containing the desired experimental raw data:
')
disp(' ')
```

```
disp('Be sure to load the data as a MATRIX')
disp('Be sure to name the matrix "raw_data"')
uiimport('-file');

% identify which pack is being imported
name_pack = input('Please enter the Pack # you are analyzing: ');
if name_pack > 11 || name_pack < 1
    error('Please enter a Pack # between 1 and 11')
end

% build the name array
name_type = input('Is this audio or tactile data? [a/t] ','s');
if name_type == 'a'
    name_type = 'Audio';
    if name_pack == 1
        name_pack = 'Pack1_';
        c = {'P1S1pushaudio'; 'P1S1pullaudio'; 'P1S2pushaudio';
'P1S2pullaudio'; 'P1S3pushaudio'; 'P1S3pullaudio'; 'P1S4pushaudio';
'P1S4pullaudio'; 'P1S5pushaudio'; 'P1S5pullaudio'; 'P1S6pushaudio';
'P1S6pullaudio'; 'P1S7pushaudio'; 'P1S7pullaudio'};
    elseif name_pack == 2
        name_pack = 'Pack2_';
        c = {'P2S1pushaudio'; 'P2S1pullaudio'; 'P2S2pushaudio';
'P2S2pullaudio'; 'P2S3pushaudio'; 'P2S3pullaudio'; 'P2S4pushaudio';
'P2S4pullaudio'; 'P2S5pushaudio'; 'P2S5pullaudio'};
    elseif name_pack == 3
        name_pack = 'Pack3_';
        c = {'P3S1pushaudio'; 'P3S1pullaudio'; 'P3S2pushaudio';
'P3S2pullaudio'; 'P3S3pushaudio'; 'P3S3pullaudio'; 'P3S4pushaudio';
'P3S4pullaudio'; 'P3S5pushaudio'; 'P3S5pullaudio'};
    elseif name_pack == 4
        name_pack = 'Pack4_';
        c = {'P4S1pushaudio'; 'P4S1pullaudio'; 'P4S2pushaudio';
'P4S2pullaudio'; 'P4S3pushaudio'; 'P4S3pullaudio'; 'P4S4pushaudio';
'P4S4pullaudio'};
    elseif name_pack == 5
        name_pack = 'Pack5_';
        c = {'P5S1pushaudio'; 'P5S1pullaudio'; 'P5S2pushaudio';
'P5S2pullaudio'; 'P5S3pushaudio'; 'P5S3pullaudio'; 'P5S4pushaudio';
'P5S4pullaudio'; 'P5S5pushaudio'; 'P5S5pullaudio'; 'P5S6pushaudio';
'P5S6pullaudio'; 'P5S7pushaudio'; 'P5S7pullaudio'};
     elseif name_pack == 6
        name_pack = 'Pack6_';
        c = {'P6S1pushaudio'; 'P6S1pullaudio'; 'P6S2pushaudio';
'P6S2pullaudio'; 'P6S3pushaudio'; 'P6S3pullaudio'; 'P6S4pushaudio';
'P6S4pullaudio'};
    elseif name_pack == 7
        name_pack = 'Pack7_';
        c = {'P7S1pushaudio'; 'P7S1pullaudio'; 'P7S2pushaudio';
'P7S2pullaudio'; 'P7S3pushaudio'; 'P7S3pullaudio'; 'P7S4pushaudio';
'P7S4pullaudio'; 'P7S5pushaudio'; 'P7S5pullaudio'; 'P7S6pushaudio';
'P7S6pullaudio'};
    elseif name_pack == 8
        name_pack = 'Pack8_';
```

```
        c = {'P8S1pushaudio'; 'P8S1pullaudio'; 'P8S2pushaudio';
'P8S2pullaudio'; 'P8S3pushaudio'; 'P8S3pullaudio'; 'P8S4pushaudio';
'P8S4pullaudio'; 'P8S5pushaudio'; 'P8S5pullaudio'; 'P8S6pushaudio';
'P8S6pullaudio'};
    elseif name_pack == 9
        name_pack = 'Pack9_';
        c = {'P9S1pushaudio'; 'P9S1pullaudio'; 'P9S2pushaudio';
'P9S2pullaudio'; 'P9S3pushaudio'; 'P9S3pullaudio'; 'P9S4pushaudio';
'P9S4pullaudio'; 'P9S5pushaudio'; 'P9S5pullaudio'};
    elseif name_pack == 10
        name_pack = 'Pack10_';
        c = {'P10S1pushaudio'; 'P10S1pullaudio'; 'P10S2pushaudio';
'P10S2pullaudio'; 'P10S3pushaudio'; 'P10S3pullaudio'; 'P10S4pushaudio';
'P10S4pullaudio'; 'P10S5pushaudio'; 'P10S5pullaudio'; 'P10S6pushaudio';
'P10S6pullaudio'; 'P10S7pushaudio'; 'P10S7pullaudio'; 'P10S8pushaudio';
'P10S8pullaudio'};
    elseif name_pack == 11
        name_pack = 'Pack11_';
        c = {'P11S1pushaudio'; 'P11S1pullaudio'; 'P11S2pushaudio';
'P11S2pullaudio'; 'P11S3pushaudio'; 'P11S3pullaudio'; 'P11S4pushaudio';
'P11S4pullaudio'; 'P11S5pushaudio'; 'P11S5pullaudio'; 'P11S6pushaudio';
'P11S6pullaudio'; 'P11S7pushaudio'; 'P11S7pullaudio'; 'P11S8pushaudio';
'P11S8pullaudio'; 'P11S9pushaudio'; 'P11S9pullaudio'};
    end
elseif name_type == 't'
    name_type = 'Tactile';
    if name_pack == 1
        name_pack = 'Pack1_';
        c = {'P1S1pushtactile'; 'P1S1pulltactile'; 'P1S2pushtactile';
'P1S2pulltactile'; 'P1S3pushtactile'; 'P1S3pulltactile';
'P1S4pushtactile'; 'P1S4pulltactile'; 'P1S5pushtactile';
'P1S5pulltactile'; 'P1S6pushtactile'; 'P1S6pulltactile';
'P1S7pushtactile'; 'P1S7pulltactile'};
    elseif name_pack == 2
        name_pack = 'Pack2_';
        c = {'P2S1pushtactile'; 'P2S1pulltactile'; 'P2S2pushtactile';
'P2S2pulltactile'; 'P2S3pushtactile'; 'P2S3pulltactile';
'P2S4pushtactile'; 'P2S4pulltactile'; 'P2S5pushtactile';
'P2S5pulltactile'};
    elseif name_pack == 3
        name_pack = 'Pack3_';
        c = {'P3S1pushtactile'; 'P3S1pulltactile'; 'P3S2pushtactile';
'P3S2pulltactile'; 'P3S3pushtactile'; 'P3S3pulltactile';
'P3S4pushtactile'; 'P3S4pulltactile'; 'P3S5pushtactile';
'P3S5pulltactile'};
    elseif name_pack == 4
        name_pack = 'Pack4_';
        c = {'P4S1pushtactile'; 'P4S1pulltactile'; 'P4S2pushtactile';
'P4S2pulltactile'; 'P4S3pushtactile'; 'P4S3pulltactile';
'P4S4pushtactile'; 'P4S4pulltactile'};
    elseif name_pack == 5
        name_pack = 'Pack5_';
        c = {'P5S1pushtactile'; 'P5S1pulltactile'; 'P5S2pushtactile';
'P5S2pulltactile'; 'P5S3pushtactile'; 'P5S3pulltactile';
'P5S4pushtactile'; 'P5S4pulltactile'; 'P5S5pushtactile';
```

```
'P5S5pulltactile'; 'P5S6pushtactile'; 'P5S6pulltactile';
'P5S7pushtactile'; 'P5S7pulltactile'};
     elseif name_pack == 6
        name_pack = 'Pack6_';
        c = {'P6S1pushtactile'; 'P6S1pulltactile'; 'P6S2pushtactile';
'P6S2pulltactile'; 'P6S3pushtactile'; 'P6S3pulltactile';
'P6S4pushtactile'; 'P6S4pulltactile'};
    elseif name_pack == 7
        name_pack = 'Pack7_';
        c = {'P7S1pushtactile'; 'P7S1pulltactile'; 'P7S2pushtactile';
'P7S2pulltactile'; 'P7S3pushtactile'; 'P7S3pulltactile';
'P7S4pushtactile'; 'P7S4pulltactile'; 'P7S5pushtactile';
'P7S5pulltactile'; 'P7S6pushtactile'; 'P7S6pulltactile'};
    elseif name_pack == 8
        name_pack = 'Pack8_';
        c = {'P8S1pushtactile'; 'P8S1pulltactile'; 'P8S2pushtactile';
'P8S2pulltactile'; 'P8S3pushtactile'; 'P8S3pulltactile';
'P8S4pushtactile'; 'P8S4pulltactile'; 'P8S5pushtactile';
'P8S5pulltactile'; 'P8S6pushtactile'; 'P8S6pulltactile'};
    elseif name_pack == 9
        name_pack = 'Pack9_';
        c = {'P9S1pushtactile'; 'P9S1pulltactile'; 'P9S2pushtactile';
'P9S2pulltactile'; 'P9S3pushtactile'; 'P9S3pulltactile';
'P9S4pushtactile'; 'P9S4pulltactile'; 'P9S5pushtactile';
'P9S5pulltactile'};
    elseif name_pack == 10
        name_pack = 'Pack10_';
        c = {'P10S1pushtactile'; 'P10S1pulltactile';
'P10S2pushtactile'; 'P10S2pulltactile'; 'P10S3pushtactile';
'P10S3pulltactile'; 'P10S4pushtactile'; 'P10S4pulltactile';
'P10S5pushtactile'; 'P10S5pulltactile'; 'P10S6pushtactile';
'P10S6pulltactile'; 'P10S7pushtactile'; 'P10S7pulltactile';
'P10S8pushtactile'; 'P10S8pulltactile'};
    elseif name_pack == 11
        name_pack = 'Pack11_';
        c = {'P11S1pushtactile'; 'P11S1pulltactile';
'P11S2pushtactile'; 'P11S2pulltactile'; 'P11S3pushtactile';
'P11S3pulltactile'; 'P11S4pushtactile'; 'P11S4pulltactile';
'P11S5pushtactile'; 'P11S5pulltactile'; 'P11S6pushtactile';
'P11S6pulltactile'; 'P11S7pushtactile'; 'P11S7pulltactile';
'P11S8pushtactile'; 'P11S8pulltactile'; 'P11S9pushtactile';
'P11S9pulltactile'};
    end
else
    error('Please enter either "a" or "t" for audio or tactile data')
end

% make sure the data is ordered correctly
check_order = input('Is the data in your imported excel file in the
correct format? [y/n] ','s');
if check_order == 'n'
    img = imread('CSV format example.png');
    name_title = 'CSV Format Example';
    set(figure,'name',name_title,'numbertitle','off')
    image(img)
```

```
    axis off
    error('Please reorder your excel file correctly before continuing.
Use this example image as a guide.')
end

[R,C] = size(raw_data);
T = raw_data(1,2);
FS = 100*ceil(T^-1/100);
name_FS = ['_FS-' num2str(FS)];
t = raw_data(4:end,1);
noise = raw_data(4:end,2);

S = 5:3:C;
All_Data = zeros(R-3,length(S));
for i = 1:((C-3)/3)
    signal = raw_data(4:end,S(i));
    %eval(sprintf('A%d = signal', i));
    All_Data(:,i) = signal;
end

str = [name_pack name_type name_FS];
save(str,'All_Data','c','t','noise');

clear all
```

## Appendix O.6. MATLAB Code: Switch Pack Output Analysis

```
%% Matt Edwards | Switch Pack Build rev2 | Last Update: 2/12/15

%***********************************************************************
****
%   - The following script is designed to analyze the audio and tactile
%     output of the switch packs.
%   - When prompted, answer the questions asked to determine which
switch
%     pack to analyze
%   - The script is designed to load files stored in pre-determined
%     locations on Matt's office computer. If you wish to run the
script on
%     another machine, you must first modify the "Data Import" section
to
%     accomadate for the change in file locations.
%   - The script will ONLY work properly if the .mat file you load was
%     created using the csv2mat.m script (All of the files provided by
%     Matt were created like this and will work).
%   - This script will output:
%     - A wav file for each switch at a sampling frequency of 11025 Hz.
%     - An amplitude vs time plot for each switch
%     - An amplitude vs frequency plot for each switch
%     - A signal-to-noise ratio plot for each switch
%     - A single-value SNR level for each switch, housed in an array
where
%       the column # corresponds to the switch # for that SNR value.
%***********************************************************************
****

clear all
close all
clc

%% Master folder location

master_folder = 'D:\Users\Matt\Documents\Data\';

%% Data Import

name_pack = input('Please enter the Pack # you are analyzing: ');
if name_pack > 11 || name_pack < 1
    error('Please enter a Pack # between 1 and 11')
end
name_type = input('Is this audio or tactile data? [a/t] ','s');
if name_type == 'a'
    name_type = 'audio';
    fig_type = 0;
elseif name_type == 't'
    name_type = 'tactile';
    fig_type = 1;
else
    error('Please enter either "a" or "t" for audio or tactile data')
end
```

```
disp(' ')
disp('Would you like the wav file to be sampled at')
freq = input('11,025 Hz or 44,100 Hz? [11025,44100] ');
if freq == 11025
    name_FS = '11025';
elseif freq == 44100
    name_FS = '44100';
else
    error('Please enter a sampling frequency of either 11025 or 44100')
end

% Audio signal switch pack files names
file_p1a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack1_Audio_FS-44100.mat'];
file_p2a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack2_Audio_FS-44100.mat'];
file_p3a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack3_Audio_FS-44100.mat'];
file_p4a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack4_Audio_FS-44100.mat'];
file_p5a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack5_Audio_FS-44100.mat'];
file_p6a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack6_Audio_FS-44100.mat'];
file_p7a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack7_Audio_FS-44100.mat'];
file_p8a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack8_Audio_FS-44100.mat'];
file_p9a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack9_Audio_FS-44100.mat'];
file_p10a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack10_Audio_FS-44100.mat'];
file_p11a = [master_folder 'Switch Packs\Switch Pack Raw Data\Audio\New
mat files\Pack11_Audio_FS-44100.mat'];

% Tactile signal switch pack files names
file_p1t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack1_Tactile_FS-44100.mat'];
file_p2t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack2_Tactile_FS-44100.mat'];
file_p3t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack3_Tactile_FS-44100.mat'];
file_p4t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack4_Tactile_FS-44100.mat'];
file_p5t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack5_Tactile_FS-44100.mat'];
file_p6t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack6_Tactile_FS-44100.mat'];
file_p7t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack7_Tactile_FS-44100.mat'];
file_p8t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack8_Tactile_FS-44100.mat'];
file_p9t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack9_Tactile_FS-44100.mat'];
```

```
file_p10t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack10_Tactile_FS-44100.mat'];
file_p11t = [master_folder 'Switch Packs\Switch Pack Raw
Data\Tactile\New mat files\Pack11_Tactile_FS-44100.mat'];

% load the switch pack data based on user inputs
if strcmp(name_type,'audio')
    if name_pack == 1
        load(file_p1a);
    elseif name_pack == 2
        load(file_p2a);
    elseif name_pack == 3
        load(file_p3a);
    elseif name_pack == 4
        load(file_p4a);
    elseif name_pack == 5
        load(file_p5a);
    elseif name_pack == 6
        load(file_p6a);
    elseif name_pack == 7
        load(file_p7a);
    elseif name_pack == 8
        load(file_p8a);
    elseif name_pack == 9
        load(file_p9a);
    elseif name_pack == 10
        load(file_p10a);
    elseif name_pack == 11
        load(file_p11a);
    end
elseif strcmp(name_type,'tactile')
    if name_pack == 1
        load(file_p1t);
    elseif name_pack == 2
        load(file_p2t);
    elseif name_pack == 3
        load(file_p3t);
    elseif name_pack == 4
        load(file_p4t);
    elseif name_pack == 5
        load(file_p5t);
    elseif name_pack == 6
        load(file_p6t);
    elseif name_pack == 7
        load(file_p7t);
    elseif name_pack == 8
        load(file_p8t);
    elseif name_pack == 9
        load(file_p9t);
    elseif name_pack == 10
        load(file_p10t);
    elseif name_pack == 11
        load(file_p11t);
    end
end
```

```
num_pack = name_pack;
name_pack = mat2str(name_pack);

% build the time array
FS = 44100;
signal = zeros(FS*0.1,3);
T = 1/FS;
t_short = [0:T:0.1]';
t_short = t_short(1:end-1);
signal(:,1) = t_short;

% build the windows
win1 = ceil(FS*0.01); % leading push window
win2 = ceil(FS*0.06); % trailing push window
win3 = ceil(FS*0.01); % leading pull window
win4 = ceil(FS*0.03); % trailing pull window

[R,C] = size(All_Data); % the size of the combined data array from all
signals

% build the strings for naming the wav files
packn = 'pack';
under = '_';
switchn = 'switch';
name_pack = [packn, name_pack, under];
name_wav = '.wav';

N = [1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9];

for S = 1:2:C
    %% push
    push = zeros(R,2);
    push(:,1) = noise;
    push(:,2) = All_Data(:,S);

    name_switch = N(S);
    name_switch = mat2str(name_switch);
    name_switch = [switchn, name_switch, under];

    % find the index location of the signal peak
    MAX = max(push(:,2));
    MIN = min(push(:,2));
    if abs(MAX) > abs(MIN)
        MAX = MAX;
    elseif abs(MAX) < abs(MIN)
        MAX = MIN;
    end
    idx = find(ismember(push(:,2),MAX)==1);

    % window the signal to only +/- win s around the peak
    push(1:idx-win1,:) = 0;
    push(idx+win2:end,:) = 0;

    % replace any zeros inside the windowed signal with nonzero values
```

```
    ind1 = find(push(idx-win1+1:idx+win2-1,1)==0);
    ind1 = ind1+(idx-win1);
    push(ind1,1) = 10^-100;
    ind2 = find(push(idx-win1+1:idx+win2-1,2)==0);
    ind2 = ind2+(idx-win1);
    push(ind2,2) = 10^-100;

    % shorten the signal to only the push
    push = push(idx-win1+1:idx+win2-1,:);
    push = padarray(push,(win1+win2)-length(push),push(1,1),'pre');

    %% pull
    % The following if statement will ignore the pull section
entirelely
    % for P2S4, P9S2, & P9S3 since those switches are push only.
    if num_pack == 2
        if S == 7
            disp(' ')
        elseif S == 9
            disp(' ')
        else
            pull = zeros(R,2);
            pull(:,1) = noise;
            pull(:,2) = All_Data(:,S+1);

            % find the index location of the signal peak
            MAX = max(pull(:,2));
            MIN = min(pull(:,2));
            if abs(MAX) > abs(MIN)
                MAX = MAX;
            elseif abs(MAX) < abs(MIN)
                MAX = MIN;
            end
            idx = find(ismember(pull(:,2),MAX)==1);

            % window the signal to only +/- 0.05 s around the peak
            pull(1:idx-win3,:) = 0;
            pull(idx+win4:end,:) = 0;

            % replace any zeros inside the windowed signal with nonzero
values
            ind1 = find(pull(idx-win3+1:idx+win4-1,1)==0);
            ind1 = ind1+(idx-win3);
            pull(ind1,1) = 10^-100;
            ind2 = find(pull(idx-win3+1:idx+win4-1,2)==0);
            ind2 = ind2+(idx-win3);
            pull(ind2,2) = 10^-100;

            % shorten the signal to only the pull
            pull = pull(idx-win3+1:idx+win4-1,:);
            pull = padarray(pull,(win3+win4)-
length(pull),pull(end,1),'post');
        end
    elseif num_pack == 9
        if S == 3
```

```
            disp(' ')
        elseif S == 5
            disp(' ')
        else
            pull = zeros(R,2);
            pull(:,1) = noise;
            pull(:,2) = All_Data(:,S+1);

            % find the index location of the signal peak
            MAX = max(pull(:,2));
            MIN = min(pull(:,2));
            if abs(MAX) > abs(MIN)
                MAX = MAX;
            elseif abs(MAX) < abs(MIN)
                MAX = MIN;
            end
            idx = find(ismember(pull(:,2),MAX)==1);

            % window the signal to only +/- 0.05 s around the peak
            pull(1:idx-win3,:) = 0;
            pull(idx+win4:end,:) = 0;

            % replace any zeros inside the windowed signal with nonzero
values
            ind1 = find(pull(idx-win3+1:idx+win4-1,1)==0);
            ind1 = ind1+(idx-win3);
            pull(ind1,1) = 10^-100;
            ind2 = find(pull(idx-win3+1:idx+win4-1,2)==0);
            ind2 = ind2+(idx-win3);
            pull(ind2,2) = 10^-100;

            % shorten the signal to only the pull
            pull = pull(idx-win3+1:idx+win4-1,:);
            pull = padarray(pull,(win3+win4)-
length(pull),pull(end,1),'post');
        end
    else
        pull = zeros(R,2);
        pull(:,1) = noise;
        pull(:,2) = All_Data(:,S+1);

        % find the index location of the signal peak
        MAX = max(pull(:,2));
        MIN = min(pull(:,2));
        if abs(MAX) > abs(MIN)
            MAX = MAX;
        elseif abs(MAX) < abs(MIN)
            MAX = MIN;
        end
        idx = find(ismember(pull(:,2),MAX)==1);
        idx = idx(1);

        % window the signal to only +/- 0.05 s around the peak
        pull(1:idx-win3,:) = 0;
        pull(idx+win4:end,:) = 0;
```

```
        % replace any zeros inside the windowed signal with nonzero
values
        ind1 = find(pull(idx-win3+1:idx+win4-1,1)==0);
        ind1 = ind1+(idx-win3);
        pull(ind1,1) = 10^-100;
        ind2 = find(pull(idx-win3+1:idx+win4-1,2)==0);
        ind2 = ind2+(idx-win3);
        pull(ind2,2) = 10^-100;

        % shorten the signal to only the pull
        pull = pull(idx-win3+1:idx+win4-1,:);
        pull = padarray(pull,(win3+win4)-
length(pull),pull(end,1),'post');
    end

    %% wav file
    if num_pack == 2
        if S == 7
            signal(1:(win1+win2),2:3) = push;
        elseif S == 9
            signal(1:(win1+win2),2:3) = push;
        else
            signal(1:(win1+win2),2:3) = push;
            signal((length(signal)-(win3+win4))+1:end,2:3) = pull;
        end
    elseif num_pack == 9
        if S == 3
            signal(1:(win1+win2),2:3) = push;
            signal((win1+win2)+1:end,2:3) = 0;
        elseif S == 5
            signal(1:(win1+win2),2:3) = push;
            signal((win1+win2)+1:end,2:3) = 0;
        else
            signal(1:(win1+win2),2:3) = push;
            signal((length(signal)-(win3+win4))+1:end,2:3) = pull;
        end
    else
        signal(1:(win1+win2),2:3) = push;
        signal((length(signal)-(win3+win4))+1:end,2:3) = pull;
    end
    signal16 = signal(:,3)./max(abs(signal(:,3)));
    if freq == 11025
        wavwrite(downsample(signal16,4),11025,16,[name_pack name_switch
name_type name_FS name_wav])
    elseif freq == 44100
        wavwrite(signal16,44100,16,[name_pack name_switch name_type
name_FS name_wav])
    end

    %% Output data analysis
    if length(signal(:,3)) < 8192
        switch_padded = padarray(signal(:,3),floor((8192-
length(signal(:,3)))/2)); % zero-pads it up to a length of 8192
```

```
        noise_padded = noise(1:7*length(switch_padded));
% the same length as the padded output array
    else switch_padded = signal(:,3);
        noise_padded =
noise(1:floor(length(noise)/length(switch_padded))*length(switch_padded
));
    end

    L = length(switch_padded);    % length of switch output signal
    NFFT = 2^nextpow2(L);         % Next power of 2 from length of
switch output signal

    % Averageing the noise vector
        % Divide the background noise array into 7 equal segments and
average
        % them. This will give the average background noise level vs.
frequency
        if L==8192
            n1=noise_padded(1:floor(1*L));
            n2=noise_padded(floor(1*L):floor(2*L));
            n3=noise_padded(floor(2*L):floor(3*L));
            n4=noise_padded(floor(3*L):floor(4*L));
            n5=noise_padded(floor(4*L):floor(5*L));
            n6=noise_padded(floor(5*L):floor(6*L));
            n7=noise_padded(floor(6*L):floor(7*L));
            % Take the FFT of the background noise segments
            N1 = (fft(n1,NFFT)/L).^2;
            N2 = (fft(n2,NFFT)/L).^2;
            N3 = (fft(n3,NFFT)/L).^2;
            N4 = (fft(n4,NFFT)/L).^2;
            N5 = (fft(n5,NFFT)/L).^2;
            N6 = (fft(n6,NFFT)/L).^2;
            N7 = (fft(n7,NFFT)/L).^2;
            % Find the total magnitude and divide by the number of
segments
            NOISE = sqrt(N1+N2+N3+N4+N5+N6+N7)./7;
        elseif floor(length(noise)/length(switch_padded))==1
            n1=noise_padded(1:floor(1*L));
            N1 = (fft(n1,NFFT)/L).^2;
            NOISE = sqrt(N1)./1;
        elseif floor(length(noise)/length(switch_padded))==2
            n1=noise_padded(1:floor(1*L));
            n2=noise_padded(floor(1*L):floor(2*L));
            N1 = (fft(n1,NFFT)/L).^2;
            N2 = (fft(n2,NFFT)/L).^2;
            NOISE = sqrt(N1+N2)./2;
        elseif floor(length(noise)/length(switch_padded))==3
            n1=noise_padded(1:floor(1*L));
            n2=noise_padded(floor(1*L):floor(2*L));
            n3=noise_padded(floor(2*L):floor(3*L));
            N1 = (fft(n1,NFFT)/L).^2;
            N2 = (fft(n2,NFFT)/L).^2;
            N3 = (fft(n3,NFFT)/L).^2;
            NOISE = sqrt(N1+N2+N3)./3;
        elseif floor(length(noise)/length(switch_padded))==4
```

```
            n1=noise_padded(1:floor(1*L));
            n2=noise_padded(floor(1*L):floor(2*L));
            n3=noise_padded(floor(2*L):floor(3*L));
            n4=noise_padded(floor(3*L):floor(4*L));
            N1 = (fft(n1,NFFT)/L).^2;
            N2 = (fft(n2,NFFT)/L).^2;
            N3 = (fft(n3,NFFT)/L).^2;
            N4 = (fft(n4,NFFT)/L).^2;
            NOISE = sqrt(N1+N2+N3+N4)./4;
        elseif floor(length(noise)/length(switch_padded))==5
            n1=noise_padded(1:floor(1*L));
            n2=noise_padded(floor(1*L):floor(2*L));
            n3=noise_padded(floor(2*L):floor(3*L));
            n4=noise_padded(floor(3*L):floor(4*L));
            n5=noise_padded(floor(4*L):floor(5*L));
            N1 = (fft(n1,NFFT)/L).^2;
            N2 = (fft(n2,NFFT)/L).^2;
            N3 = (fft(n3,NFFT)/L).^2;
            N4 = (fft(n4,NFFT)/L).^2;
            N5 = (fft(n5,NFFT)/L).^2;
            NOISE = sqrt(N1+N2+N3+N4+N5)./5;
        elseif floor(length(noise)/length(switch_padded))==6
            n1=noise_padded(1:floor(1*L));
            n2=noise_padded(floor(1*L):floor(2*L));
            n3=noise_padded(floor(2*L):floor(3*L));
            n4=noise_padded(floor(3*L):floor(4*L));
            n5=noise_padded(floor(4*L):floor(5*L));
            n6=noise_padded(floor(5*L):floor(6*L));
            N1 = (fft(n1,NFFT)/L).^2;
            N2 = (fft(n2,NFFT)/L).^2;
            N3 = (fft(n3,NFFT)/L).^2;
            N4 = (fft(n4,NFFT)/L).^2;
            N5 = (fft(n5,NFFT)/L).^2;
            N6 = (fft(n6,NFFT)/L).^2;
            NOISE = sqrt(N1+N2+N3+N4+N5+N6)./6;
        elseif floor(length(noise)/length(switch_padded))==7
            n1=noise_padded(1:floor(1*L));
            n2=noise_padded(floor(1*L):floor(2*L));
            n3=noise_padded(floor(2*L):floor(3*L));
            n4=noise_padded(floor(3*L):floor(4*L));
            n5=noise_padded(floor(4*L):floor(5*L));
            n6=noise_padded(floor(5*L):floor(6*L));
            n7=noise_padded(floor(6*L):floor(7*L));
            N1 = (fft(n1,NFFT)/L).^2;
            N2 = (fft(n2,NFFT)/L).^2;
            N3 = (fft(n3,NFFT)/L).^2;
            N4 = (fft(n4,NFFT)/L).^2;
            N5 = (fft(n5,NFFT)/L).^2;
            N6 = (fft(n6,NFFT)/L).^2;
            N7 = (fft(n7,NFFT)/L).^2;
            NOISE = sqrt(N1+N2+N3+N4+N5+N6+N7)./7;
        end

    SWITCH = fft(switch_padded,NFFT)/L;  % Perform a fourier transform
on the output data
```

```
    NOISE = 2*abs(NOISE(1:NFFT/2+1));     % fft returns complex
values...
    SWITCH = 2*abs(SWITCH(1:NFFT/2+1));   % ...Find the magnitude and
remove the folded values
    f = FS/2*linspace(0,1,NFFT/2+1);      % Create the frequency array.

    % Correctly spaces the X axis in the frequency plots
    if f(14) > 80
        NOISE([1:13]) = 0;
        SWITCH([1:13]) = 0;
    elseif f(27) > 80
        NOISE([1:26]) = 0;
        SWITCH([1:26]) = 0;
    else NOISE([1:51]) = 0;
        SWITCH([1:51]) = 0;
    end

    % Normalized the plots
%     signal(:,2) = signal(:,2)./max(abs(signal(:,3)));
%     signal(:,3) = signal(:,3)./max(abs(signal(:,3)));
%     NOISE = NOISE./max(SWITCH);
%     SWITCH = SWITCH./max(SWITCH);

    % Plot output & noise vs time
    name_title = [name_pack ' ' name_switch ' ' name_type ' ' ': Output
& Background Noise vs Time'];
    set(figure,'name',name_title,'numbertitle','off')
    plot(signal(:,1),signal(:,3)./max(abs(signal(:,3))),'r-
',signal(:,1),signal(:,2)./max(abs(signal(:,3))),'b-')
    title(name_title)
    xlabel('time (s)')
    ylabel('Normalized Amplitude')
    legend('Output','Background Noise','Location','Best')
    axis([0 0.1 -1 1])
    grid on

    % Plot output & noise vs log frequency
    name_title = [name_pack ' ' name_switch ' ' name_type ' ' ': Output
& Background Noise vs Frequency'];
    set(figure,'name',name_title,'numbertitle','off')
    semilogx(f,10.*log10(SWITCH./mean(NOISE)),'r-
',f,10.*log10(NOISE./mean(NOISE)),'b-')
    title(name_title)
    xlabel('Frequency (Hz)')
    ylabel('Amplitude (dB)')
    legend('Output','Background Noise','Location','Best')
    grid on
    axis([80 10000 -15 37])

    out_freq = 10.*log10(SWITCH./mean(NOISE));
out_freq(isinf(out_freq)) = NaN;
    out_noise = 10.*log10(NOISE./mean(NOISE));
out_noise(isinf(out_noise)) = NaN;
    freq_lim(S,1:2) = [nanmin([nanmin(out_freq) nanmin(out_noise)])
nanmax(out_freq)];
```

```
    % Plot Signal-to-Noise ratio (SNR) vs log frequency
    SNR = 20.*log10(SWITCH./NOISE);
    name_title = [name_pack ' ' name_switch ' ' name_type ' ' ':
Signal-to-Noise Ratio (SNR) vs Frequency'];
    set(figure,'name',name_title,'numbertitle','off')
    semilogx(f,SNR,'r-')
    title(name_title)
    xlabel('Frequency (Hz)')
    ylabel('Output / Noise (dB)')
    grid on
    axis([80 10000 -30 82])

    SNR_lim(S,1:2) = [min(SNR) max(SNR)];

    % Find the single-value SNR
    SNR(isnan(SNR)) = -inf;
    a = 0;
    M = length (SNR);
    for n = 1:M;
        a = a + (10.^(SNR(n)./10));
    end
    SNR_tot(N(S)) = 10.*log10(a/length(SNR));
end

SNR_tot = SNR_tot'

cascade

freq_lim = [min(freq_lim) max(freq_lim)]
SNR_lim = [min(SNR_lim) max(SNR_lim)]

%% Save the figures as .tiff files
idx = 2;

if fig_type == 0
    if C == idx*4 || C == idx*5 || C == idx*6 || C == idx*7 || C ==
idx*8 || C == idx*9
        print -f1 -dtiff Switch1_15cm_audio_output_vs_time.tiff
        print -f2 -dtiff Switch1_15cm_audio_output_vs_freq.tiff
        print -f3 -dtiff Switch1_15cm_audio_SNR.tiff

        print -f4 -dtiff Switch2_15cm_audio_output_vs_time.tiff
        print -f5 -dtiff Switch2_15cm_audio_output_vs_freq.tiff
        print -f6 -dtiff Switch2_15cm_audio_SNR.tiff

        print -f7 -dtiff Switch3_15cm_audio_output_vs_time.tiff
        print -f8 -dtiff Switch3_15cm_audio_output_vs_freq.tiff
        print -f9 -dtiff Switch3_15cm_audio_SNR.tiff

        print -f10 -dtiff Switch4_15cm_audio_output_vs_time.tiff
        print -f11 -dtiff Switch4_15cm_audio_output_vs_freq.tiff
        print -f12 -dtiff Switch4_15cm_audio_SNR.tiff
    end
```

```
    if C == idx*5 || C == idx*6 || C == idx*7 || C == idx*8 || C ==
idx*9
        print -f13 -dtiff Switch5_15cm_audio_output_vs_time.tiff
        print -f14 -dtiff Switch5_15cm_audio_output_vs_freq.tiff
        print -f15 -dtiff Switch5_15cm_audio_SNR.tiff
    end
    if C == idx*6 || C == idx*7 || C == idx*8 || C == idx*9
        print -f16 -dtiff Switch6_15cm_audio_output_vs_time.tiff
        print -f17 -dtiff Switch6_15cm_audio_output_vs_freq.tiff
        print -f18 -dtiff Switch6_15cm_audio_SNR.tiff
    end
    if C == idx*7 || C == idx*8 || C == idx*9
        print -f19 -dtiff Switch7_15cm_audio_output_vs_time.tiff
        print -f20 -dtiff Switch7_15cm_audio_output_vs_freq.tiff
        print -f21 -dtiff Switch7_15cm_audio_SNR.tiff
    end
    if C == idx*8 || C == idx*9
        print -f22 -dtiff Switch8_15cm_audio_output_vs_time.tiff
        print -f23 -dtiff Switch8_15cm_audio_output_vs_freq.tiff
        print -f24 -dtiff Switch8_15cm_audio_SNR.tiff
    end
    if C == idx*9
        print -f25 -dtiff Switch9_15cm_audio_output_vs_time.tiff
        print -f26 -dtiff Switch9_15cm_audio_output_vs_freq.tiff
        print -f27 -dtiff Switch9_15cm_audio_SNR.tiff
    end
elseif fig_type == 1
    if C == idx*4 || C == idx*5 || C == idx*6 || C == idx*7 || C ==
idx*8 || C == idx*9
        print -f1 -dtiff Switch1_tactile_output_vs_time.tiff
        print -f2 -dtiff Switch1_tactile_output_vs_freq.tiff
        print -f3 -dtiff Switch1_tactile_SNR.tiff

        print -f4 -dtiff Switch2_tactile_output_vs_time.tiff
        print -f5 -dtiff Switch2_tactile_output_vs_freq.tiff
        print -f6 -dtiff Switch2_tactile_SNR.tiff

        print -f7 -dtiff Switch3_tactile_output_vs_time.tiff
        print -f8 -dtiff Switch3_tactile_output_vs_freq.tiff
        print -f9 -dtiff Switch3_tactile_SNR.tiff

        print -f10 -dtiff Switch4_tactile_output_vs_time.tiff
        print -f11 -dtiff Switch4_tactile_output_vs_freq.tiff
        print -f12 -dtiff Switch4_tactile_SNR.tiff
    end
    if C == idx*5 || C == idx*6 || C == idx*7 || C == idx*8 || C ==
idx*9
        print -f13 -dtiff Switch5_tactile_output_vs_time.tiff
        print -f14 -dtiff Switch5_tactile_output_vs_freq.tiff
        print -f15 -dtiff Switch5_tactile_SNR.tiff
    end
    if C == idx*6 || C == idx*7 || C == idx*8 || C == idx*9
        print -f16 -dtiff Switch6_tactile_output_vs_time.tiff
        print -f17 -dtiff Switch6_tactile_output_vs_freq.tiff
        print -f18 -dtiff Switch6_tactile_SNR.tiff
```

```
    end
    if C == idx*7 || C == idx*8 || C == idx*9
        print -f19 -dtiff Switch7_tactile_output_vs_time.tiff
        print -f20 -dtiff Switch7_tactile_output_vs_freq.tiff
        print -f21 -dtiff Switch7_tactile_SNR.tiff
    end
    if C == idx*8 || C == idx*9
        print -f22 -dtiff Switch8_tactile_output_vs_time.tiff
        print -f23 -dtiff Switch8_tactile_output_vs_freq.tiff
        print -f24 -dtiff Switch8_tactile_SNR.tiff
    end
    if C == idx*9
        print -f25 -dtiff Switch9_tactile_output_vs_time.tiff
        print -f26 -dtiff Switch9_tactile_output_vs_freq.tiff
        print -f27 -dtiff Switch9_tactile_SNR.tiff
    end
end
```

## Appendix O.7. MATLAB Code: Haptic Board Voltage Measurement

```
%% Matt Edwards | Board Direct Drive Measurement | 01/26/14

%***********************************************************************
****
%   - The following script is designed to analyze the voltage output
%     of the haptic board.
%   - The script allows multiple options based on what the user wishes
to
%     analyze: signal type, gain setting, volume setting, etc.
%   - The script is designed to load files stored in pre-determined
%     locations on Matt's office computer. If you wish to run the
script on
%     another machine, you must first modify the "master_folder"
variable
%     to accomadate for the change in file locations.
%***********************************************************************
****

clear all
close all
clc

%% Master folder location

master_folder = 'D:\Users\Matt\Documents\Data\';

%% Data Import

% Find impendance loading condition
type1 = input('Are you analyzing the volume at various gain settings?
[y/n] ','s');
if type1 == 'y'
    gain = input('What is the gain setting? [6/12/18/24] ');
    type2 = 'n';
    if gain == 6 || gain == 12 || gain == 18 || gain == 24
    else
        error('Please enter either 6, 12, 18, or 24')
    end
    vol = input('1, 35, or 36 vol? [1/35/36] ');
        if vol == 35 || vol == 36 || vol == 1
        else
            error('Please enter either 1, 35, or 36');
        end
elseif type1 == 'n'
    disp('What is the impedance loading configuration of the data?');
    imp = input('No load (n), resistor loaded (r) or exciter loaded
(e)? [n/r/e] ','s');
    if imp == 'n' || imp == 'r' || imp == 'e'
    else
        error('Please enter either n, r, or e');
    end
    % Find signal type
```

```
    type2 = input('Analyzing white noise or sine sweep data? [w/s]
','s');
    if type2 == 'w' || type2 == 's'
    else
        error('Please enter either w or s');
    end
    % Find volume setting
    vol = input('35 or 36 vol? [35/36] ');
    if vol == 35 || vol == 36
    else
        error('Please enter either 35 or 36');
    end
    % Find the number of samples to be averaged
    if type2 == 'w'
        num = input('how many white noise samples do you want to
average? ');
        if num > 10 || num < 2
            error('Please enter a number between 2 and 10.')
        end
    end
elseif type1 == 'y' || type1 == 'n'
else
    error('Please enter either y or n')
end

% File locations for the output data. If you move those files or rename
any
% folders in the file path, you need to update them here too.

% Sine sweep & white noise testing
file_es35 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.21.15 Board direct drive msmt - exciter load\Sine
Sweep\sine_sweep_11025_35vol_exciter_loaded.mat'];
file_es36 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.21.15 Board direct drive msmt - exciter load\Sine
Sweep\sine_sweep_11025_36vol_exciter_loaded.mat'];
file_ew35 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.21.15 Board direct drive msmt - exciter load\White
Noise\white_noise_11025_35vol_exciter_loaded.mat'];
file_ew36 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.21.15 Board direct drive msmt - exciter load\White
Noise\white_noise_11025_36vol_exciter_loaded.mat'];

file_ns35 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.26.15 board direct drive msmt - no load\Sine
Sweep\sine_sweep_11025_35vol_no_load.mat'];
file_ns36 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.26.15 board direct drive msmt - no load\Sine
Sweep\sine_sweep_11025_36vol_no_load.mat'];
file_nw35 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.26.15 board direct drive msmt - no load\White
Noise\white_noise_11025_35vol_no_load.mat'];
file_nw36 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.26.15 board direct drive msmt - no load\White
Noise\white_noise_11025_36vol_no_load.mat'];
```

```
file_rs35 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.26.15 Board direct drive msmt - resistor load\Sine
Sweep\sine_sweep_11025_35vol_resistor_loaded.mat'];
file_rs36 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.26.15 Board direct drive msmt - resistor load\Sine
Sweep\sine_sweep_11025_36vol_resistor_loaded.mat'];
file_rw35 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.26.15 Board direct drive msmt - resistor load\White
Noise\white_noise_11025_35vol_resistor_loaded.mat'];
file_rw36 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.26.15 Board direct drive msmt - resistor load\White
Noise\white_noise_11025_36vol_resistor_loaded.mat'];

% Gain testing
file_6dB1vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\6 dB
gain\sine_sweep_11025_6dB_1vol.mat'];
file_6dB35vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\6 dB
gain\sine_sweep_11025_6dB_35vol.mat'];
file_6dB36vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\6 dB
gain\sine_sweep_11025_6dB_36vol.mat'];

file_12dB1vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\12 dB
gain\sine_sweep_11025_12dB_1vol.mat'];
file_12dB35vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\12 dB
gain\sine_sweep_11025_12dB_35vol.mat'];
file_12dB36vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\12 dB
gain\sine_sweep_11025_12dB_36vol.mat'];

file_18dB1vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\18 dB
gain\sine_sweep_11025_18dB_1vol.mat'];
file_18dB35vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\18 dB
gain\sine_sweep_11025_18dB_35vol.mat'];
file_18dB36vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\18 dB
gain\sine_sweep_11025_18dB_36vol.mat'];

file_24dB1vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\24 dB
gain\sine_sweep_11025_24dB_1vol.mat'];
file_24dB35vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\24 dB
gain\sine_sweep_11025_24dB_35vol.mat'];
file_24dB36vol = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.29.15 Board direct drive msmt - gain testing\24 dB
gain\sine_sweep_11025_24dB_36vol.mat'];
```

```
% Input data
file_in_sine = [master_folder 'Exciter Measurement\Exciter Input WAV
files\New wav files\sine sweep\sine_sweep_11025.wav'];
file_in_white = [master_folder 'Exciter Measurement\Exciter Input WAV
files\New wav files\white noise\10 separete files
01.21.15\white_noise_source.mat'];

% load the proper data based on previously selected user inputs
if type1 == 'y'
    [in.sig in.fs] = audioread(file_in_sine);
    if gain == 6
        if vol == 1
            raw_data = load(file_6dB1vol);
        elseif vol == 35
            raw_data = load(file_6dB35vol);
        elseif vol == 36
            raw_data = load(file_6dB36vol);
        end
    elseif gain == 12
        if vol == 1
            raw_data = load(file_12dB1vol);
        elseif vol == 35
            raw_data = load(file_12dB35vol);
        elseif vol == 36
            raw_data = load(file_12dB36vol);
        end
    elseif gain == 18
        if vol == 1
            raw_data = load(file_18dB1vol);
        elseif vol == 35
            raw_data = load(file_18dB35vol);
        elseif vol == 36
            raw_data = load(file_18dB36vol);
        end
    elseif gain == 24
        if vol == 1
            raw_data = load(file_24dB1vol);
        elseif vol == 35
            raw_data = load(file_24dB35vol);
        elseif vol == 36
            raw_data = load(file_24dB36vol);
        end
    end
elseif type1 == 'n'
    if type2 == 's'
        [in.sig in.fs] = audioread(file_in_sine);
        if imp == 'n'
            if vol == 35
                raw_data = load(file_ns35);
            elseif vol == 36
                raw_data = load(file_ns36);
            end
        elseif imp == 'r'
            if vol == 35
                raw_data = load(file_rs35);
```

```
            elseif vol == 36
                raw_data = load(file_rs36);
            end
        elseif imp == 'e'
            if vol == 35
                raw_data = load(file_es35);
            elseif vol == 36
                raw_data = load(file_es36);
            end
        end
    elseif type2 == 'w'
        in = load(file_in_white);
        if imp == 'n'
            if vol == 35
                raw_data = load(file_nw35);
            elseif vol == 36
                raw_data = load(file_nw36);
            end
        elseif imp == 'r'
            if vol == 35
                raw_data = load(file_rw35);
            elseif vol == 36
                raw_data = load(file_rw36);
            end
        elseif imp == 'e'
            if vol == 35
                raw_data = load(file_ew35);
            elseif vol == 36
                raw_data = load(file_ew36);
            end
        end
    end
end
raw_data = struct2array(raw_data);
clc

%% Output data time analysis
out.t_all = raw_data(:,1);
if type2 == 's' || type1 == 'y'
    out.sig_all = raw_data(:,2);
    if size(raw_data) > 3
        out.sig = out.sig_all(raw_data(1,3):raw_data(1,4),1);
    else
        out.sig = sig_all;
    end
elseif type2 == 'w'
    % grab the white noise output signal arrays from the raw data
    for n = 1:num
        out.sig_all(:,n) = raw_data(:,n+1);
    end
    if size(raw_data) > 11
        % if the raw data array contains index locations of where the
        % white noise itself begins & ends in the signal array, utilize
        % these idecies to trim the signal array.
        for n = 1:num
```

```
            out.sig(:,n) =
out.sig_all(raw_data(n,12):raw_data(n,13),n);
        end
    else
        for n = 1:num
            out.sig(:,n) = out.sig_all(:,n);
        end
    end
end

%% Frequency and TF analysis
if type2 == 's' || type1 == 'y'
    in.t = 0:1/in.fs:length(in.sig)/in.fs; % sine sweep time vector
end
in.T = 1/in.fs; % sample time
in.L = length(in.t); % length of signal
in.NFFT = 2^nextpow2(in.L); % Next power of 2 from length of input
signal

out.fs = 44100; % sampling frequency
out.T = 1/out.fs; % sample time
out.L = length(out.sig); % length of  output signal
out.t = (0:out.L-1)*out.T; %  output time vector
out.NFFT = 2^nextpow2(out.L); % Next power of 2 from length of output
signal

if type2 == 's' || type1 == 'y'
    in.tf = in.L/in.fs; % final time
    in.SIG = fft(in.sig,in.NFFT); % FFT for input
    out.SIG_TF = fft(downsample(out.sig,4),out.NFFT/4); % FFT for
output (downsampled for TF)
    TF = out.SIG_TF./in.SIG; % Transfer function
    TF = 2/in.L*abs(TF(1:in.NFFT/2+1));
    out.SIG = fft(out.sig,out.NFFT); % FFT for output (not downsampled)

    in.SIG = (2/in.L)*abs(in.SIG(1:in.NFFT/2+1));
    in.f = in.fs/2*linspace(0,1,in.NFFT/2+1); % input frequency vector
    out.SIG = (2/out.L)*abs(out.SIG(1:out.NFFT/2+1));
    out.f = out.fs/2*linspace(0,1,out.NFFT/2+1); % output frequency
vector
elseif type2 == 'w'
    % FFT for input
    for n = 1:num
        in.SIG(:,n) = fft(in.sig(:,n),in.NFFT);
    end
    % remove folded values, then take the mean, then take the magnitude
    in.SIG_mean = 2/in.L*abs(mean(in.SIG(1:in.NFFT/2+1),1));
    % input frequency vector
    in.f = in.fs/2*linspace(0,1,in.NFFT/2+1);
    for n = 1:num
        % FFT for output (downsampled for TF)
        out.SIG_TF(:,n) = fft(downsample(out.sig(:,n),4),out.NFFT/4);
        % Transfer function
        TF(:,n) = out.SIG_TF(:,n)./in.SIG(:,n);
    end
```

```
    % take the mean, then remove folded values, then take the magnitude
    TF = mean(TF,2);
    TF = 2/in.L*abs(TF(1:in.NFFT/2+1));
    % FFT for output (not downsampled for TF)
    for n = 1:num
        out.SIG(:,n) = fft(out.sig(:,n),out.NFFT);
    end
    % remove folded values, then take the mean, then take the magnitude
    out.SIG_mean = 2/out.L*abs(mean(out.SIG(1:out.NFFT/2+1),1));
    % output frequency vector
    out.f = out.fs/2*linspace(0,1,out.NFFT/2+1);
end

%% Plotting
if type2 == 's' || type1 == 'y'
    % create the plot titles
    if type2 == 's'
        if imp == 'n'
            if vol == 35
                title1 = [{'Sine Sweep (no load): 35 vol, 6 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 35 vol, 6 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 35 vol, 6 dB gain'};
{'Transfer Function'}];
            elseif vol == 36
                title1 = [{'Sine Sweep (no load): 36 vol, 6 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 36 vol, 6 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 36 vol, 6 dB gain'};
{'Transfer Function'}];
            end
        elseif imp == 'r'
            if vol == 35
                title1 = [{'Sine Sweep (resistor loaded): 35 vol, 6 dB
gain'}; {'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (resistor loaded): 35 vol, 6 dB
gain'}; {'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (resistor loaded): 35 vol, 6 dB
gain'}; {'Transfer Function'}];
            elseif vol == 36
                title1 = [{'Sine Sweep (resistor loaded): 36 vol, 6 dB
gain'}; {'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (resistor loaded): 36 vol, 6 dB
gain'}; {'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (resistor loaded): 36 vol, 6 dB
gain'}; {'Transfer Function'}];
            end
        elseif imp == 'e'
            if vol == 35
                title1 = [{'Sine Sweep (exciter loaded): 35 vol, 6 dB
gain'}; {'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (exciter loaded): 35 vol, 6 dB
gain'}; {'Input & Output vs Frequency'}];
```

```
                title3 = [{'Sine Sweep (exciter loaded): 35 vol, 6 dB
gain'}; {'Transfer Function'}];
            elseif vol == 36
                title1 = [{'Sine Sweep (exciter loaded): 36 vol, 6 dB
gain'}; {'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (exciter loaded): 36 vol, 6 dB
gain'}; {'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (exciter loaded): 36 vol, 6 dB
gain'}; {'Transfer Function'}];
            end
        end
    elseif type1 == 'y'
        if gain == 6
            if vol == 1
                title1 = [{'Sine Sweep (no load): 1 vol, 6 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 1 vol, 6 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 1 vol, 6 dB gain'};
{'Transfer Function'}];
            elseif vol == 35
                title1 = [{'Sine Sweep (no load): 35 vol, 6 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 35 vol, 6 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 35 vol, 6 dB gain'};
{'Transfer Function'}];
            elseif vol == 36
                title1 = [{'Sine Sweep (no load): 36 vol, 6 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 36 vol, 6 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 36 vol, 6 dB gain'};
{'Transfer Function'}];
            end
        elseif gain == 12
            if vol == 1
                title1 = [{'Sine Sweep (no load): 1 vol, 12 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 1 vol, 12 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 1 vol, 12 dB gain'};
{'Transfer Function'}];
            elseif vol == 35
                title1 = [{'Sine Sweep (no load): 35 vol, 12 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 35 vol, 12 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 35 vol, 12 dB gain'};
{'Transfer Function'}];
            elseif vol == 36
                title1 = [{'Sine Sweep (no load): 36 vol, 12 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 36 vol, 12 dB gain'};
{'Input & Output vs Frequency'}];
```

```
                title3 = [{'Sine Sweep (no load): 36 vol, 12 dB gain'};
{'Transfer Function'}];
            end
        elseif gain == 18
            if vol == 1
                title1 = [{'Sine Sweep (no load): 1 vol, 18 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 1 vol, 18 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 1 vol, 18 dB gain'};
{'Transfer Function'}];
            elseif vol == 35
                title1 = [{'Sine Sweep (no load): 35 vol, 18 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 35 vol, 18 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 35 vol, 18 dB gain'};
{'Transfer Function'}];
            elseif vol == 36
                title1 = [{'Sine Sweep (no load): 36 vol, 18 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 36 vol, 18 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 36 vol, 18 dB gain'};
{'Transfer Function'}];
            end
        elseif gain == 24
            if vol == 1
                title1 = [{'Sine Sweep (no load): 1 vol, 24 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 1 vol, 24 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 1 vol, 24 dB gain'};
{'Transfer Function'}];
            elseif vol == 35
                title1 = [{'Sine Sweep (no load): 35 vol, 24 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 35 vol, 24 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 35 vol, 24 dB gain'};
{'Transfer Function'}];
            elseif vol == 36
                title1 = [{'Sine Sweep (no load): 36 vol, 24 dB gain'};
{'Input & Output vs Time'}];
                title2 = [{'Sine Sweep (no load): 36 vol, 24 dB gain'};
{'Input & Output vs Frequency'}];
                title3 = [{'Sine Sweep (no load): 36 vol, 24 dB gain'};
{'Transfer Function'}];
            end
        end
    end
    % Plot input & output vs time
    % shift the input signal array so that it starts at the same time
the
    % output signal does on the plot
```

```
    in.t_plot = 0:1/in.fs:((raw_data(1,3)/out.fs) + in.t(end));
    in.sig_plot = padarray(in.sig,length(in.t_plot) -
length(in.sig),0,'pre');
    set(figure,'name','Time','numbertitle','off')
    % Place the output for the 1 vol plots in the foreground since
they're
    % much smaller than their respective inputs
    if vol == 1
        plot(in.t_plot,in.sig_plot,'b-',out.t_all,out.sig_all,'r-')
        legend('Input','Output',0)
    else
        plot(out.t_all,out.sig_all,'r-',in.t_plot,in.sig_plot,'b-')
        legend('Output','Input',0)
    end
    xlabel('Time (s)')
    ylabel('Voltage (V)')
    title(title1)
    ylim([-6 6])
    grid on

    % Plot input & output vs frequency
    set(figure,'name','Frequency','numbertitle','off')

semilogx(out.f,10.*log10(out.SIG./mean(in.SIG)),'r',in.f,10.*log10(in.S
IG./mean(in.SIG)),'b')
    xlabel('Frequency (Hz)'), ylabel('Amplitude (dB)')
    xlim([10 10000])
    ylim([-50 12])
    title(title2)
    legend('Output','Input')
    grid on

    % Plot transfer function
    set(figure,'name','Transfer Function','numbertitle','off')
    semilogx(in.f,10.*log10(TF),'b')
    xlabel('Frequency (Hz)'), ylabel('Amplitdue (dB)')
    title(title3)
    xlim([10 11025/2])
    ylim([-60 -15])
    grid on

elseif type2 == 'w'
    % create plot titles
    if imp == 'n'
        if vol == 35
            title1 = [{'White Noise (no load): 35 vol, 6 dB gain'};
{'Input & Output vs Time (1 of 10 samples)'}];
            title2 = [{'White Noise (no load): 35 vol, 6 dB gain'};
{'Input & Output vs Frequency (10 samples averaged)'}];
            title3 = [{'White Noise (no load): 35 vol, 6 dB gain'};
{'Transfer Function (10 samples averaged)'}];
        elseif vol == 36
            title1 = [{'White Noise (no load): 36 vol, 6 dB gain'};
{'Input & Output vs Time (1 of 10 samples)'}];
```

```
            title2 = [{'White Noise (no load): 36 vol, 6 dB gain'};
{'Input & Output vs Frequency (10 samples averaged)'}];
            title3 = [{'White Noise (no load): 36 vol, 6 dB gain'};
{'Transfer Function (10 samples averaged)'}];
        end
    elseif imp == 'r'
        if vol == 35
            title1 = [{'White Noise (resistor load): 35 vol, 6 dB
gain'}; {'Input & Output vs Time (1 of 10 samples)'}];
            title2 = [{'White Noise (resistor load): 35 vol, 6 dB
gain'}; {'Input & Output vs Frequency (10 samples averaged)'}];
            title3 = [{'White Noise (resistor load): 35 vol, 6 dB
gain'}; {'Transfer Function (10 samples averaged)'}];
        elseif vol == 36
            title1 = [{'White Noise (resistor load): 36 vol, 6 dB
gain'}; {'Input & Output vs Time'}];
            title2 = [{'White Noise (resistor load): 36 vol, 6 dB
gain'}; {'Input & Output vs Frequency (10 samples averaged)'}];
            title3 = [{'White Noise (resistor load): 36 vol, 6 dB
gain'}; {'Transfer Function (10 samples averaged)'}];
        end
    elseif imp == 'e'
        if vol == 35
            title1 = [{'White Noise (exciter load): 35 vol, 6 dB
gain'}; {'Input & Output vs Time (1 of 10 samples)'}];
            title2 = [{'White Noise (exciter load): 35 vol, 6 dB
gain'}; {'Input & Output vs Frequency (10 samples averaged)'}];
            title3 = [{'White Noise (exciter load): 35 vol, 6 dB
gain'}; {'Transfer Function (10 samples averaged)'}];
        elseif vol == 36
            title1 = [{'White Noise (exciter load): 36 vol, 6 dB
gain'}; {'Input & Output vs Time (1 of 10 samples)'}];
            title2 = [{'White Noise (exciter load): 36 vol, 6 dB
gain'}; {'Input & Output vs Frequency (10 samples averaged)'}];
            title3 = [{'White Noise (exciter load): 36 vol, 6 dB
gain'}; {'Transfer Function (10 samples averaged)'}];
        end
    end
    % Plot input & output vs time
    % shift the input signal array so that it starts at the same time
the
    % output signal does on the plot
    in.t_plot = 0:1/in.fs:((raw_data(1,12)/out.fs) + in.t(end));
    in.sig_plot = padarray(in.sig(:,1),length(in.t_plot) -
length(in.sig(:,1)),0,'pre');
    set(figure,'name','Time','numbertitle','off')
    plot(out.t_all,out.sig_all(:,1),'r-',in.t_plot,in.sig_plot,'b-')
    xlabel('Time (s)')
    ylabel('Voltage (V)')
    title(title1)
    legend('Output','Input',0)
    grid on
    ylim([-5 5])

    % Plot input & output vs frequency
```

```
    set(figure,'name','Frequency','numbertitle','off')

semilogx(out.f,10.*log10(out.SIG_mean./mean(in.SIG_mean)),'r',in.f,10.*
log10(in.SIG_mean./mean(in.SIG_mean)),'b')
    xlabel('Frequency (Hz)'), ylabel('Amplitude (dB)')
    xlim([10 10000])
    ylim([-40 10])
    title(title2)
    legend('Output','Input')
    grid on

    % Plot transfer function
    set(figure,'name','Transfer Function','numbertitle','off')
    semilogx(in.f,10.*log10(TF),'b')
    xlabel('Frequency (Hz)'), ylabel('Amplitdue (dB)')
    title(title3)
    xlim([10 11025/2])
    ylim([-60 -15])
    grid on
end

cascade
```

## Appendix O.8. MATLAB Code: Exciter Voltage Measurement

```
%% Matt Edwards | Exciter Direct Drive Measurement | 01/30/15

%**********************************************************************
****
%   - The following script is designed to analyze the velocity output
%     of the exciter when it is not attached to the haptic board.
%   - The script allows multiple options based on what the user wishes
to
%     analyze: signal type, voltage setting, etc.
%   - The script is designed to load files stored in pre-determined
%     locations on Matt's office computer. If you wish to run the
script on
%     another machine, you must first modify the "master_folder"
variable
%     to accomadate for the change in file locations.
%**********************************************************************
****

clear all
close all
clc

%% Master folder location

master_folder = 'D:\Users\Matt\Documents\Data\';

%% Data Import

% Find impendance loading condition
type = input('Analyzing white noise or sine sweep data? [w/s] ','s');
if type == 'w' || type == 's'
else
    error('Please enter either w or s');
end
% Find volume setting
volt = input('What is the peak-to-peak voltage? [1/4/8] ');
if volt == 1 || volt == 4 || volt == 8
else
    error('Please enter either 1, 4, or 8');
end
% Find the number of samples to be averaged
if type == 'w'
    num = input('how many white noise samples do you want to average?
');
    if num > 40 || num < 2
        error('Please enter a number between 2 and 40.')
    end
end

% File locations for the data. If you move those files or rename any
% folders in the file path, you need to update them here too.
```

```
file_s1 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.30.15 exciter direct drive msmt\Sine
Sweep\sine_sweep_1V.mat'];
file_s4 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.30.15 exciter direct drive msmt\Sine
Sweep\sine_sweep_4V.mat'];
file_s8 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.30.15 exciter direct drive msmt\Sine
Sweep\sine_sweep_8V.mat'];

file_w1 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.30.15 exciter direct drive msmt\White
Noise\white_noise_1V.mat'];
file_w4 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.30.15 exciter direct drive msmt\White
Noise\white_noise_4V.mat'];
file_w8 = [master_folder 'Exciter Measurement\Exciter Raw Data\Direct
Drive\01.30.15 exciter direct drive msmt\White
Noise\white_noise_8V.mat'];
file_w4_40 = [master_folder 'Exciter Measurement\Exciter Raw
Data\Direct Drive\01.30.15 exciter direct drive msmt\White
Noise\white_noise_4V_40.mat'];

% load the proper data based on previously selected user inputs
if type == 's'
    if volt == 1
        raw_data = load(file_s1);
    elseif volt == 4
        raw_data = load(file_s4);
    elseif volt == 8
        raw_data = load(file_s8);
    end
    raw_data = struct2array(raw_data);
    in.raw_data = raw_data(:,1:2);
    out.raw_data(:,1) = raw_data(:,1); out.raw_data(:,2) =
raw_data(:,3);
elseif type == 'w'
   if volt == 1
        raw_data = load(file_w1);
   elseif volt == 4
%          raw_data = load(file_w4);
        raw_data = load(file_w4_40);
   elseif volt == 8
        raw_data = load(file_w8);
   end
   raw_data = struct2array(raw_data);
%    in.raw_data = raw_data(:,1:11);
   in.raw_data = raw_data(:,1:41);
%    out.raw_data(:,1) = raw_data(:,1); out.raw_data(:,2:11) =
raw_data(:,12:21);
   out.raw_data(:,1) = raw_data(:,1); out.raw_data(:,2:41) =
raw_data(:,42:81);
end
clc
```

```
%% Output data time analysis
if type == 's'
    in.sig_all = in.raw_data(:,2);
    in.t_all = in.raw_data(:,1);
    out.sig_all = out.raw_data(:,2);
    in.t_all = in.raw_data(:,1);
    in.sig = in.sig_all(raw_data(1,4):raw_data(1,5),1);
    out.sig = out.sig_all(raw_data(1,4):raw_data(1,5),1);
elseif type == 'w'
    % grab the white noise output signal arrays from the raw data
    for n = 1:num
        in.sig_all(:,n) = in.raw_data(:,n+1);
        in.sig(:,n) = in.sig_all(:,n);
        out.sig_all(:,n) = out.raw_data(:,n+1);
        out.sig(:,n) = out.sig_all(:,n);
    end
    in.t = in.raw_data(:,1);
    out.t = out.raw_data(:,1);
end

%% Frequency and TF analysis
if type == 's'
    in.t = 0:1/44100:(length(in.sig)/44100);
    in.t = in.t(1:end-1)';
    out.t = 0:1/44100:(length(out.sig)/44100);
    out.t = out.t(1:end-1)';
end
in.fs = 44100;
in.T = 1/in.fs; % sample time
in.L = length(in.t); % length of signal
in.NFFT = 2^nextpow2(in.L); % Next power of 2 from length of input
signal

out.fs = 44100; % sampling frequency
out.T = 1/out.fs; % sample time
out.L = length(out.sig); % length of  output signal
out.NFFT = 2^nextpow2(out.L); % Next power of 2 from length of output
signal

if type == 's'
    in.SIG = fft(in.sig,in.NFFT); % FFT for input
    out.SIG = fft(out.sig,out.NFFT); % FFT for output

    TF = out.SIG./in.SIG; % Transfer function
    TF = 2/in.L*abs(TF(1:in.NFFT/2+1));

    in.SIG = (2/in.L)*abs(in.SIG(1:in.NFFT/2+1));
    in.f = in.fs/2*linspace(0,1,in.NFFT/2+1); % input frequency vector

    out.SIG = (2/out.L)*abs(out.SIG(1:out.NFFT/2+1));
    out.f = out.fs/2*linspace(0,1,out.NFFT/2+1); % output frequency
vector
elseif type == 'w'
    % FFT for input
    for n = 1:num
```

```
        in.SIG(:,n) = fft(in.sig(:,n),in.NFFT);
    end
    % remove folded values, then take the mean, then take the magnitude
    in.SIG_mean = 2/in.L*abs(mean(in.SIG(1:in.NFFT/2+1),1));
    % input frequency vector
    in.f = in.fs/2*linspace(0,1,in.NFFT/2+1);
    for n = 1:num
        % FFT for output
        out.SIG(:,n) = fft(out.sig(:,n),out.NFFT);
        % Transfer function
        TF(:,n) = out.SIG(:,n)./in.SIG(:,n);
    end
    % take the mean, then remove folded values, then take the magnitude
    TF = mean(TF,2);
    TF = 2/in.L*abs(TF(1:in.NFFT/2+1));
    % remove folded values, then take the mean, then take the magnitude
    out.SIG_mean = 2/out.L*abs(mean(out.SIG(1:out.NFFT/2+1),1));
    % output frequency vector
    out.f = out.fs/2*linspace(0,1,out.NFFT/2+1);
end

% elseif type == 'w'
%     % FFT for input
%     for n = 1:num
%         in.SIG(:,n) = fft(in.sig(:,n),in.NFFT);
%     end
%     % remove folded values, then take the mean, then take the
magnitude
%     in.SIG_mean = 2/in.L*abs(mean(in.SIG(1:in.NFFT/2+1),1));
%     % input frequency vector
%     in.f = in.fs/2*linspace(0,1,in.NFFT/2+1);
%     for n = 1:num
%         % FFT for output (downsampled for TF)
%         out.SIG_TF(:,n) = fft(downsample(out.sig(:,n),4),out.NFFT/4);
%         % Transfer function
%         TF(:,n) = out.SIG_TF(:,n)./in.SIG(:,n);
%     end
%     % take the mean, then remove folded values, then take the
magnitude
%     TF = mean(TF,2);
%     TF = 2/in.L*abs(TF(1:in.NFFT/2+1));
%     % FFT for output (not downsampled for TF)
%     for n = 1:num
%         out.SIG(:,n) = fft(out.sig(:,n),out.NFFT);
%     end
%     % remove folded values, then take the mean, then take the
magnitude
%     out.SIG_mean = 2/out.L*abs(mean(out.SIG(1:out.NFFT/2+1),1));
%     % output frequency vector
%     out.f = out.fs/2*linspace(0,1,out.NFFT/2+1);
% end

%% Plotting
if type == 's'
    % create the plot titles
```

```
    if volt == 1
        title1 = [{'Sine Sweep: 1 V pk-pk'}; {'Input & Output vs
Time'}];
        title2 = [{'Sine Sweep: 1 V pk-pk'}; {'Input & Output vs
Frequency'}];
        title3 = [{'Sine Sweep: 1 V pk-pk'}; {'Transfer Function'}];
    elseif volt == 4
        title1 = [{'Sine Sweep: 4 V pk-pk'}; {'Input & Output vs
Time'}];
        title2 = [{'Sine Sweep: 4 V pk-pk'}; {'Input & Output vs
Frequency'}];
        title3 = [{'Sine Sweep: 4 V pk-pk'}; {'Transfer Function'}];
    elseif volt == 8
        title1 = [{'Sine Sweep: 8 V pk-pk'}; {'Input & Output vs
Time'}];
        title2 = [{'Sine Sweep: 8 V pk-pk'}; {'Input & Output vs
Frequency'}];
        title3 = [{'Sine Sweep: 8 V pk-pk'}; {'Transfer Function'}];
    end
    % Plot input & output vs time
    set(figure,'name','Time','numbertitle','off')
    % Place the output for the 1 volt plots in the foreground since
they're
    % much smaller than their respective inputs
    plot(in.t,in.sig./max(in.sig),'b-',out.t,out.sig./max(out.sig),'r-
')
    legend('Input','Output',0)
    xlabel('Time (s)')
    ylabel('Normalized Amplitude')
    title(title1)
    xlim([0 out.t(end)]), ylim([-1 1])
    grid on

    % Plot input & output vs frequency
    set(figure,'name','Frequency','numbertitle','off')

semilogx(out.f,10.*log10(out.SIG./mean(in.SIG)),'r',in.f,10.*log10(in.S
IG./mean(in.SIG)),'b')
    xlabel('Frequency (Hz)'), ylabel('Amplitude (dB)')
    xlim([10 10000])
    ylim([-50 10])
    title(title2)
    legend('Output','Input')
    grid on

    % Plot transfer function
    set(figure,'name','Transfer Function','numbertitle','off')
    semilogx(in.f,10.*log10(TF),'b')
    xlabel('Frequency (Hz)'), ylabel('Amplitdue (dB)')
    title(title3)
    xlim([10 10000])
    ylim([-95 -40])
    grid on

elseif type == 'w'
```

```
    % create the plot titles
    if volt == 1
        title1 = [{'White Noise: 1 V pk-pk'}; {'Input & Output vs Time
(1 of 10 samples)'}];
        title2 = [{'White Noise: 1 V pk-pk'}; {'Input & Output vs
Frequency (10 samples averaged)'}];
        title3 = [{'White Noise: 1 V pk-pk'}; {'Transfer Function (10
samples averaged)'}];
    elseif volt == 4
        title1 = [{'White Noise: 4 V pk-pk'}; {'Input & Output vs Time
(1 of 40 samples)'}];
        title2 = [{'White Noise: 4 V pk-pk'}; {'Input & Output vs
Frequency (40 samples averaged)'}];
        title3 = [{'White Noise: 4 V pk-pk'}; {'Transfer Function (40
samples averaged)'}];
    elseif volt == 8
        title1 = [{'White Noise: 8 V pk-pk'}; {'Input & Output vs Time
(1 of 9 samples)'}];
        title2 = [{'White Noise: 8 V pk-pk'}; {'Input & Output vs
Frequency (9 samples averaged)'}];
        title3 = [{'White Noise: 8 V pk-pk'}; {'Transfer Function (9
samples averaged)'}];
    end
    % Plot input & output vs time
    % shift the input signal array so that it starts at the same time
the
    % output signal does on the plot
    set(figure,'name','Time','numbertitle','off')
    plot(in.t,in.sig(:,1)./max(in.sig(:,1)),'b-
',out.t,out.sig(:,1)./max(out.sig(:,1)),'r-')
    xlabel('Time (s)'), ylabel('Amplitude')
    title(title1)
    legend('Input','Output',0)
    grid on
    xlim([0 out.t(end)]), ylim([-1 1])

    % Plot input & output vs frequency
    set(figure,'name','Frequency','numbertitle','off')

semilogx(out.f,10.*log10(out.SIG_mean./mean(in.SIG_mean)),'r',in.f,10.*
log10(in.SIG_mean./mean(in.SIG_mean)),'b')
    xlabel('Frequency (Hz)'), ylabel('Amplitude (dB)')
    xlim([10 10000])
    ylim([-50 10])
    title(title2)
    legend('Output','Input')
    grid on

    % Plot transfer function
    set(figure,'name','Transfer Function','numbertitle','off')
    semilogx(in.f,10.*log10(TF),'b')
    xlabel('Frequency (Hz)'), ylabel('Amplitdue (dB)')
    title(title3)
    xlim([10 10000])
    ylim([-95 -40])
```

```
    grid on
end

cascade
```

# REFERENCES


1. M.E. Altinsoy et al., “Knocking sound as quality sign for household appliances and the evaluation of the audio-haptic interaction,” in Proc. Haptic and Audio Interaction Design 7th International Conference, Berlin, Germany, 2012.

2. P. Etienne et al. “Analysis of Car Door Closing Sound Quality.” Applied Acoustics. [On-line]. 69, pp. 12-22, 2008.

3. G. Preyer et al. “Auditory Alert Characteristics: A Survey of Pilor Views,” The International Journal of Aviation Psychology, 15:3, pp. 233-250, Nov. 2009.

4. H. Fastl, “Psychoacoustic Basis of Sound Quality Evaluation and Sound Engineering,” in Proc. 13th International Congress on Sound and Vibration, Vienna, Austria, 2006.

5. S. Ishimitsu et al., "Study on the visualization of the impression of button sounds," International Journal of Innovative Computing, Information & Control, vol. 5, pp. 4189-4203, 2009.

6. K. Fujinoki and S. Ishimitsu, "Automated Sound Quality Evaluation of Push-Button Sounds Using Triangular Biorthogonal Wavelets," Journal of Signal Processing, vol. 16, pp. 433-441, 2012.

7. E. Hoggan et al., "Crossmodal congruence: the look, feel and sound of touchscreen widgets," in Proc. 10th International Conference on Mulitmodal Interfaces, New York, NY, 2008.

8. E. Hoggan et al., “Mobile Multi-Actuator Tactile Displays,” in Proc. 2nd International Workshop on Haptic and Audio Interaction Design, 2007, pp. 22-33.

9. K. Cunefare, Integrated Acoustics Laboratory [On-line]. Avaiable: http://www-old.me.gatech.edu/acoustics/IAL/resources.html

10. J. Wilson, “Development and Validation of a Laminant Flooring System Sound Quality Test Method,” M.S. thesis, Dept. Mech. Eng., Georgia Inst. of Tech., Atlanta, 2009.

11. D. Inman, Engineering Vibration. Prentice-Hall Inc., 2001, pp. 207, 504.

12. Lyon, R.H., “Product Sound Quality - from Perception to Design”. Journal of Sound and Vibration, 2003: p. 18-22.

13. Norm Otto, C.E., Scott Lake, “Guidelines for Jury Evaluations of Automotive Sounds”. Journal of Sound and Vibrations, 2001 (April).

14. Microtrack II Professional 2-Channel Mobile Digital Recorder User Guide [On-line]. Available: http://www.lib.utk.edu/ius/M-Track%20Audio%20User%20Guide.pdf

15. C. Zaiontz, (2014). Real Statistics Using Excel [On-line]. Available: http://www.realstatistics.com/free-download/real-statistics-resource-pack/

16. NIST/SEMATECH e-Handbook of Statistical Methods [On-line]. Available: http://www.itl.nist.gov/div898/handbook/