Audio Spectrum Analyzer and Manipulator  
 Software Design Document

Josh Borgman and Josh Reyes

Wednesday, December 10, 2014

Contents

[1. Introduction 3](#_Toc402433840)

[a. Purpose 3](#_Toc402433841)

[b. Scope 3](#_Toc402433842)

[c. Definitions and Acronyms 3](#_Toc402433843)

[2. System Overview 3](#_Toc402433844)

[3. System Architecture 3](#_Toc402433845)

[a. Architectural Design 3](#_Toc402433846)

[b. Design Rationale 3](#_Toc402433847)

[4. Human Interface Design 4](#_Toc402433848)

[a. Overview of User Interface 4](#_Toc402433849)

[b. Screen Images 4](#_Toc402433850)

# Introduction

## Purpose

This software design document describes the architecture and system design of an audio spectrum analyzer that is also capable of editing the audio data.

## Scope

This software aims to analyze the frequency spectrum of an audio file. The main goal is to develop an intuitive program for displaying the applications of audio manipulation through use of Fourier analysis.

## Definitions and Acronyms

ASA: Audio Spectrum Analyzer

FFT: Fast Fourier Transformation

LV: LabView 2014 32-bit

DFT: Discrete Fourier Transform

# System Overview

The basis of this software is the functional intake of audio perturbations that can be used in systematic manipulation of the input audio files in the frequency domain. Transition between the time domain and frequency domain is accomplished through the application of Discrete Fourier Transformation. Post transform, the magnitude of each frequency within the spectrum are available and can be plotted and manipulated. Time domain waveforms of all audio signals are plotted for visualization. When computationally analyzed, a discrete (rather than continuous) Fourier transform is used. Discrete Fourier Transformation is implemented by the computer. The transform takes in a finite list of samples of a function (i.e. our waveform) and outputs the list of coefficients off a finite combination of complex sinusoids sorted by their respective frequencies. In practice as with this. The algorithmic implementation of the DFT is known as the fast Fourier transform (FFT) algorithm which factors the discrete Fourier matrix into a product of predominantly zero factors. Real time manipulation of the input audio signal is accomplished through the use of a digital band pass filter (BPF). The first section of the software utilizes the NI DAQ interface with the addition of a microphone input to record a voltage representation of the air pressure fluctuations. The input signal is filtered using a band-pass filter with user specified cut-off frequencies. The band-pass filter functions analogously to its analog counterpart, passing frequencies within a designated range and rejecting frequencies outside that range. The cut-off inputs can be specified to turn the band-pass filter into a low-pass or high-pass filter. The option to play the recorded waveform is available on the UI. The output of this first section is file (in .wav form) record of the waveform. The second section of the software begins by reading in two audio files from specified directories, noting the format must be in an approved format (.wav). Once read in, the file waveforms are combined and normalized to avoid clipping of the resultant waveform. A FFT is used to obtain the frequency spectrum of the mixed audio file. A three dimensional plot of the decomposed frequency spectrum will be generated for the entirety of the audio file in sampling of one-second clips. Options to play the mixed audio files are available for auditory examination of the mixed file. Visualization of the mixed signal frequency contents are displayed in totality and in second long samples.

case the DFT takes real samples with zero-valued complex parts. The DFT functions as a truncated version of infinitely sampling discrete-time Fourier transform. The DFT is further defined by the formula

# System Architecture

## Architectural Design

The software is composed of two main sections with at least two subsections. The two major sections are connected by their .wav file outputs/inputs. The first section (refer to Figure 1: AudioMixAnalyzer.vi) has three subsections. The first subsection takes in an audio recording. The second sub-section (refer to Figure 2: soundIn\_out.vi) is an optional digital band-pass filter which accepts the audio signal input and applies a rigid restriction on the frequency content. The third sub-section plots the signal waveform after the optional filtering. The second major section of the software is composed of three subcomponents. The first sub-section is a mixing section which takes in two audio files and mixes them and plots the mixed output signal. The second section takes the mixed audio signal as an input and performs a frequency decomposition of the entire waveform, which is plotted. The third subsection of this component takes the mixed waveform and samples one second clips, plotting the frequency content of each clip versus time on a 3d scatter plot.

## Design Rationale

The two-VI format, with each VI particular to one of the aforementioned major sections, are designed autonomously so as to allow for better interchangeability and integration with other VIs or analytical packages. Thusly the user can choose to use one VI and not the other, or both is sequence. Each subsection of the components contributes to one of three purposes, namely the acquisition of data, the processing of that data, or the education of the user. The Three subsections of the first VI accomplishes all three purposes mentioned earlier, so as to have a practical purpose of taking in the data augmented with the option of performing real time manipulation via the filter, with the added benefit of an instantaneous and coherent output that is readily interpretable by the user. The second VI accomplishes all three purposes as well. The first subsection both processes and creates new data that is processed and displayed by the second and third subsections. Further, the differentiation between the third and second subsections allows for a higher-level analysis of the frequency content of the signal in the second sub-component with a more in depth analysis and visualization for the user available in the third sub-component.

# Human Interface Design

## Overview of User Interface

The software is broken into two separate VIs that have two separate UIs. The first, titles AudioMixAnalyzer allows the user to record a sound clip of a specified time length and frequency range. By pressing the “Record” button, the microphone picks up sound waves for the specified duration and records this wave as a .wav file with a user-selected title. The sound wave is then displayed in an amplitude (in Voltage RMS) vs. time graph. The user then has the option to listen to their recording by pressing the “Playback” button on the front panel. This plays back either the unfiltered sound wave or a filtered version according to the user’s specifications.

The second VI, titled soundIn\_Out, prompts the user for up to two audio file. After the user inputs the files of their choosing, the formatting of each recording is displayed and the sound waveforms of each file is displayed as “Input Audio 1 Waveform” and “Input Audio 2 Waveform” corresponding to files called by the user. If two samples are entered into the prompt, the VI then combines the two audio samples into a combined output sample by summing the amplitude of the two inputs and renormalizing in order to avoid any clipping, saving the output as a new, user-specified .wav file. Now the user has the option of listening to the combined audio sample with the “Play Mixed Output” button. In addition to the amplitude vs. time graph, this VI also analyzes the amplitude of the range of frequencies by applying them over a power spectrum in both two and three dimensions (amplitude vs. frequency and amplitude vs. frequency vs. time). These graphics are then displayed near the bottom of the front panel as “Frequency Power Spectrum” and “Time-Dependent Power Spectrum” respectively.

## Screen Images

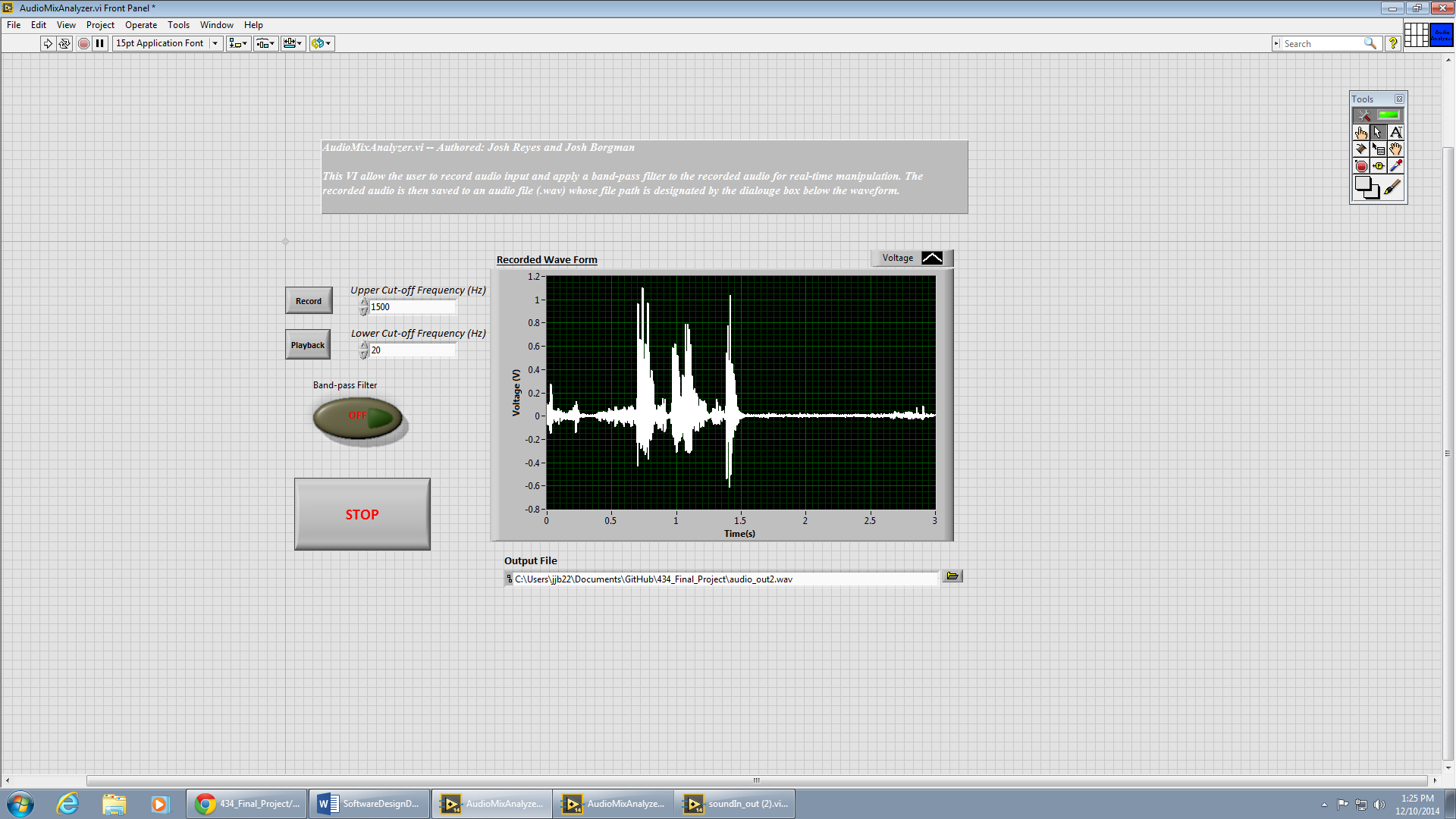


Figure : AudioMixAnalyzer Front Panel

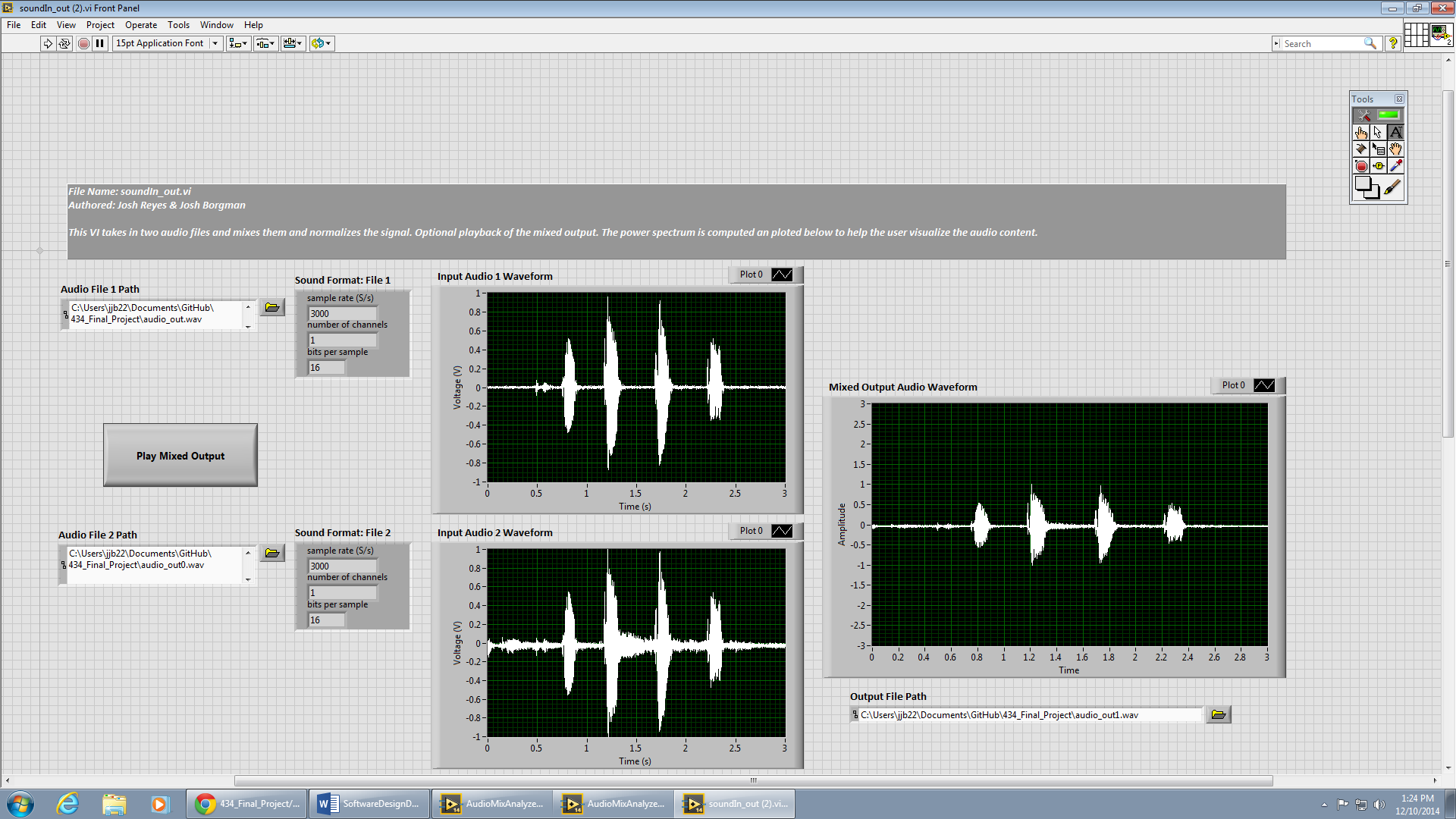


Figure :soundIn\_Out Front Panel (input and output sample waveforms)

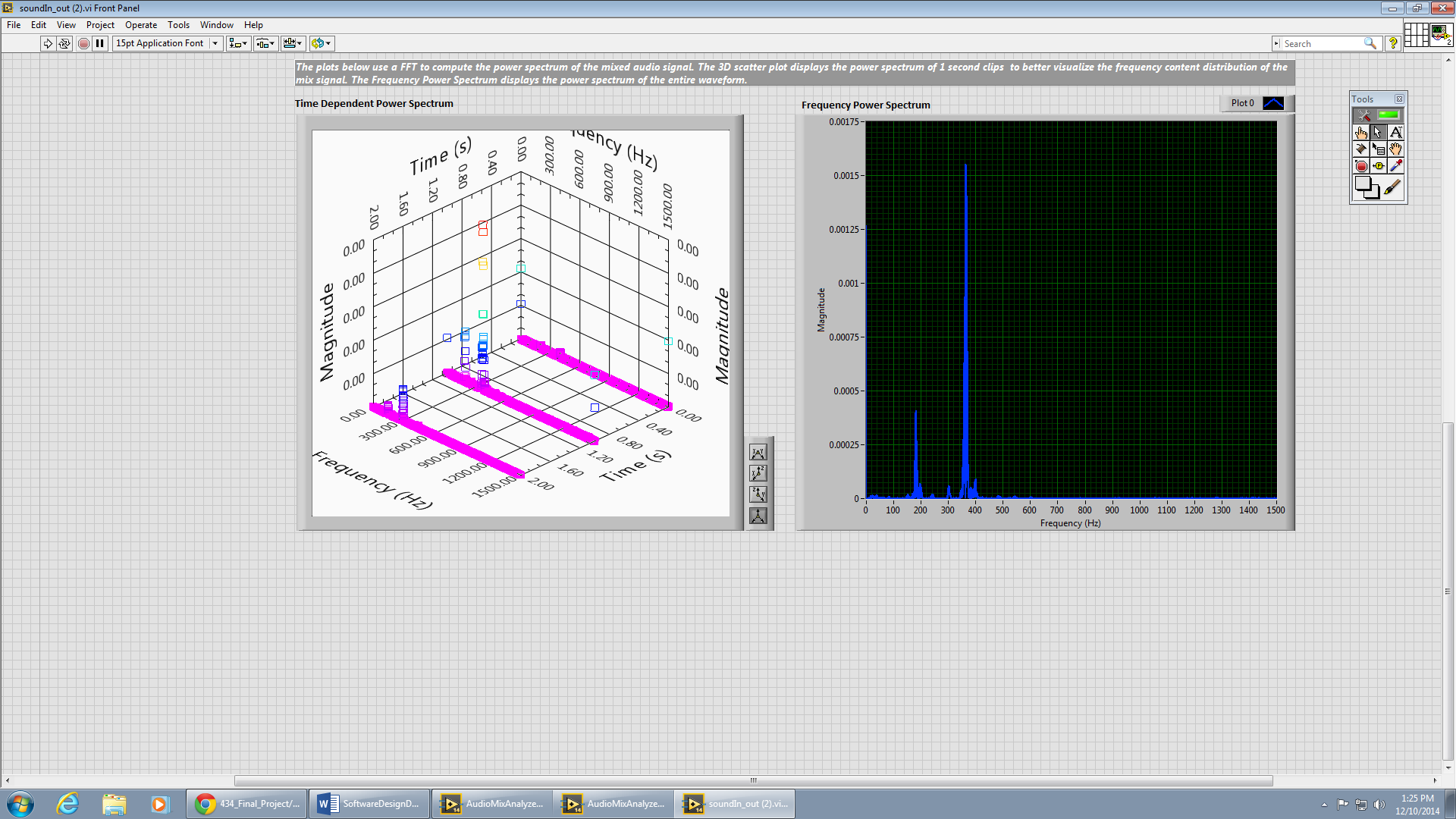


Figure : soundIn\_Out Front Panel (power spectrum analysis plots)