Logan Caraway
May 1, 2020
EELE477 Final Project

Signal Denoising Report

Introduction

There are many causes of sinusoidal noise in an audio file. Unfortunately, an audio file can often not be rerecorded, so the original audio file must be cleaned for use. Cleaning audio files by hand requires graphing and calculation of frequencies to be nulled which can be a time consuming process when cleaning a large number of audio files. The purpose of this project was to automate the cleaning process such that when an audio voice file is presented, the code will automatically find the sinusoidal noise in the audio file, remove the noise, and return the cleaned audio file. I selected this project because I considered it to be the most interesting of the projects with an obvious real-world use.

Approach

Since speech is composed of numerous short sinusoidal signals of different frequencies throughout, the magnitude plot of the Fourier transform of an audio signal tends to be more smooth without any single frequency having a significantly larger magnitude. Sinusoidal noise, on the other hand, is a frequency occurring throughout the entire file. The Fourier transform of sinusoidal noise would appear as a delta. This means that the Fourier transform of a speech audio file with sinusoidal noise would appear somewhat smooth with deltas at the frequencies of the sinusoidal noise. Once these frequencies are found, nulling filters can be used to remove the noise. This same process is used regardless of if the cleaning is done by hand or is automated. To automate this process, a program needs to automatically find these frequencies and perform nulling.

In order to automatically remove the sinusoidal noise from the audio file, I wrote a matlab function called denoise.m that takes the name of a speech audio file and returns the cleaned sequence of reals and the sampling frequency of the file after having sinusoidal noise removed. It was specified for the problem that there would be at most 3 noise sinusoids. My function starts by opening the audio file into a sequence of reals, then has the main for loop, and finally returns the cleaned sequence. The main for loop executes a maximum of 6 times. The 6 was determined because there were at most 3 sinusoidal signals and some signals could take 2 loops to remove, so 6 was a worst case. The loop starts by taking the Fourier transform of the audio signal. Since sinusoidal noise appears as a delta, it should have the maximum magnitude in the frequency domain, so it finds the index of the maximum. The loop copies the dirty sequence and zeroes out the index of the max and a couple surrounding indexes. It then compares the sum of the original sequence and the reduced sequence; if zeroing out the max and a couple surrounding indexes removes 1% or more of total information then it is considered to be a delta (sinusoidal noise). 1% was used because if the threshold isn't harsh enough, then noise isn't removed, but if the threshold is too harsh, very minimal damage is done to the audio file. If the max is not a delta, then all noise has been removed and the loop is exited. If an index is determined to contain a delta, it is converted to a frequency using the length of the Fourier transform of the signal and the sampling frequency. This frequency is used in the creation of a nulling filter and the filter is convolved with the original signal to clean it. This process is repeated on the progressively cleaner signal until six loops have been completed or all deltas have been removed. This process was inspired solely by the material taught in EELE477 Digital Signal Processing and simple reasoning.

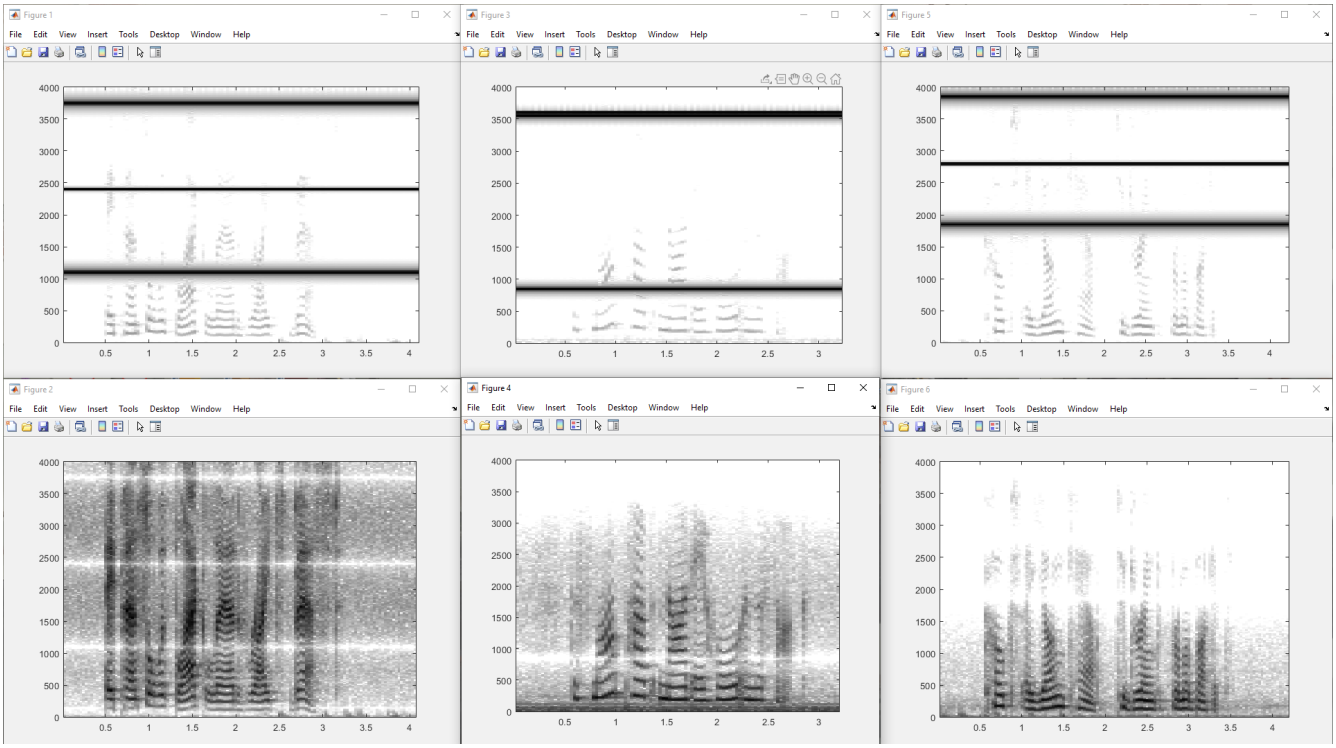

Figure 1: Decibel Spectrograms from running the code on 3 test files. Top row is before cleaning. Bottom row is after cleaning.

Results

Figure 1 shows the results of running the algorithm on test audio speech files. The top 3 images show the decibel spectrograms from before running denoise.m. The dark black lines are the sinusoidal noise. The bottom 3 images are the decibel spectrograms of the cleaned audio speech files. Notice that the dark lines have been removed. 1 audio speech file was used when creating the algorithm and 9 other audio files were used to confirm functionality. The code is available at https://github.com/LoganCaraway/Automated-Signal-Denoising.

Conclusion

Cleaning audio files by hand can be a time consuming process when cleaning a large number of audio files. The purpose of this project was to automate the cleaning process such that when an audio voice file is presented, the code will automatically find the sinusoidal noise in the audio file, remove the noise, and return the cleaned audio file. My matlab function, denoise.m, successfully removes the sinusoidal noise from an audio speech file. The current implementation expects at most 3 noise sinusoids. Increasing this may not be straitforward, since it would mean increasing the number of loops which can have an adverse effect when there aren't many noise sinusoids due to the over zealous nature of the selected threshold for identifying deltas. Throughout this process, I have gained a lot of experience in audio manipulation. The current implementation is a good proof of concept, but there are numerous ways to improve the design.