Cochlear Implant Project

Phase 1

Faduma Ahmed (20747788)
Elena Damus (20708019)
Martin Le (20628761)

BME 252
Prof. Nima Maftoon
June 10, 2019

Task 1: Evaluation Method

In order to evaluate the different signal processors that will be created in Phase 3, a variety of different metrics will be placed in an evaluation matrix and scores will be assigned for each metric. The signal processor that scores best compared to the others will be considered the best design. The idea of a cochlear implant is that people will be listening to the output signal and attempting to understand the audio. Therefore, subjective scoring is just as valuable as meeting objective sound-quality targets, and so both subjective and objective scoring methods should be used.

The first method of evaluation is a subjective score, assigned by test subjects listening to the output, based on their interpretation of the sound quality. Scores will be assigned on a scale from 1 to 10, with 1 being the worst score and 10 being the best. This is a valuable method because it provides direct information about how people interpret and understand the sound being produced, which varies depending on the person.

The second method of evaluation that will be used to evaluate the signal processing algorithm is the ability to distinguish between types of common noises. Listeners will be presented with output signals of a very high pitched noise and a very low pitched noise, and asked to attempt to distinguish between the two. Scores will be assigned on a scale of 1 to 10, where the score is the number of speakers, out of 10 total, that were correctly identified by the listener.

A third evaluation metric will be used that is very similar to the previous method outlined above. The metric will evaluate the signal processors' abilities to distinguish when the type of sound has changed in one audio file. Listeners will be asked to determine at which point in the signal the first noise stopped and another type of noise started. An example of this could be a car horn that is switched to the sound of a person walking down the hallway. Again, 10 changes in sounds will be played, so the score will be a numerical value between 1 and 10 representing how many the listener was able to correctly identify.

In addition to the listener-based evaluation methods, two more sophisticated methods will be used that will evaluate the signal against specific sound quality targets. The first is the ability of the signal processors to remove noise from the signal by removing certain frequencies. This will improve the overall quality and the signal to noise ratio. By applying a noise averaging algorithm to reduce electronic noise from unknown sources such as the recording microphone, this may ultimately improve the signal to noise ratio and qualitative sound quality. The percent decrease in noise for each signal processor design will be assigned a score from 1 to 10, where 10 represents a 100% decrease, based on the change in average amplitude of the noise from the input signal to the output signal. Both signals will need to be plotted in order for this calculation to be made.

The second target sound-quality evaluation method is the use of cross correlation algorithm. Cross correlation algorithms measure the similarity between two different signals. In comparison to convolution, the cross-correlation algorithms undergo the same convolution sum process but do not require the initial step of time inversion. Cross-correlation can be visualized by overlaying two signals, and sliding one signal across the other in order to 'match' the template signal. Mathematically, the correlation coefficient can be determined in a variety of techniques such as measuring the sum of the squares of differences between the aligned values of the sliding and template signal. This determined value would increase as the differences between the two signals increase. In this project, a cross correlation algorithm will allow for a quantitative measurement of the accuracy between the original input signal and the processed output signal. The determined cross correlation coefficient will be converted into a percentage and will be assigned a score from 1 to 10, where 10 represents a 100% match.

An example evaluation matrix including all of the metrics outlined above is displayed in Table 1 below. The values that will be inserted as the scores for each metric and design will be on a scale of 1 to 10 for each box, and each score will be an average. For example, if 5 test listeners are used to evaluate each design, then each value inputted will be the average score assigned by the 5 test listeners for that specific metric and design. Example scores for a design have been inputted in Table 1 to better illustrate this scoring method.

Weights have also been added to the matrix to better improve the usefulness of the evaluation matrix. Total scores will be calculated by summing, for each design, the products of the initial scores and the weights for the corresponding metrics. Weights were assigned according to the signal processing abilities that were determined to be priorities through discussion between team members. It was decided that the overall quality of sound as determined by listeners would be the most important metric, and assigned a weight of 24%, since the goal of a signal processor in a cochlear implant is to filter a signal such that the sound transmitted to the user is clear and comprehensible. Noise reduction was also determined to be important, assigned a weight of 22%, since it is known that large amounts of noise are irritating to a user and decrease the comprehensiveness of audio files, particularly ones involving speech. The other three metrics are all weighted identically, at 18% each. The 5 weights add up to 1.0, or 100%.

Table 1: Example Evaluation Matrix for Signal Processor Designs

Evaluation metric	**Weight**	**Design 1**	**Design 2**	**Design 3**
Quality of sound	0.24	7		
Distinction between higher and lower pitches	0.18	8		
Detection of sound changes	0.18	9		
Noise reduction	0.22	7.4		
Cross correlation	0.18	8		
Total (Max 10)	1.0	7.808		

Task 2: Acquired Sound Files To Be Processed

Three different sets of sound files were obtained or recorded in order to be used with the evaluation methods as outlined in Task 1. The first series of sound files contains 10 various sounds, played one at a time, in succession. This will be used for the third evaluation metric, to determine whether the signal processor can distinguish between common sounds like car horns, loud footsteps, and simple melodies. The second series of sound files contains a variety of high and low pitched sounds, to be used in the second evaluation metric. This will be useful as these sounds contain a wide variety of frequencies which will test the algorithm's ability to differentiate different combinations of frequencies.

The third series of files will consist of both male and female speech of Harvard sentences. Harvard sentences are 'phonetically balanced' sentences which are commonly used for standardized testing for communication devices. The phonemes, or sounds which distinguish words from one another, within this sentence appear at the same frequency as they appear in the English language. Audio samples of Harvard sentences will be sampled online or self-recorded. This will be used with all evaluation metrics.

All 3 files sets will be used to test to test the quality of audio, as users will be given the input and output files and will be asked to rank the difference in quality. These scores will range from 1-10 based on the average of the listeners' responses.

Task 3: Plots and Matlab File

The plots of the processed signal and two waveforms of the cosine signal are displayed in Figures 1 and 2 below, respectively. The Matlab files required are attached at the end of this report. There are two files; the first is the Phase 1 script that includes the code for plotting the two graphs, and the second is the script for resampling to 16KHz.

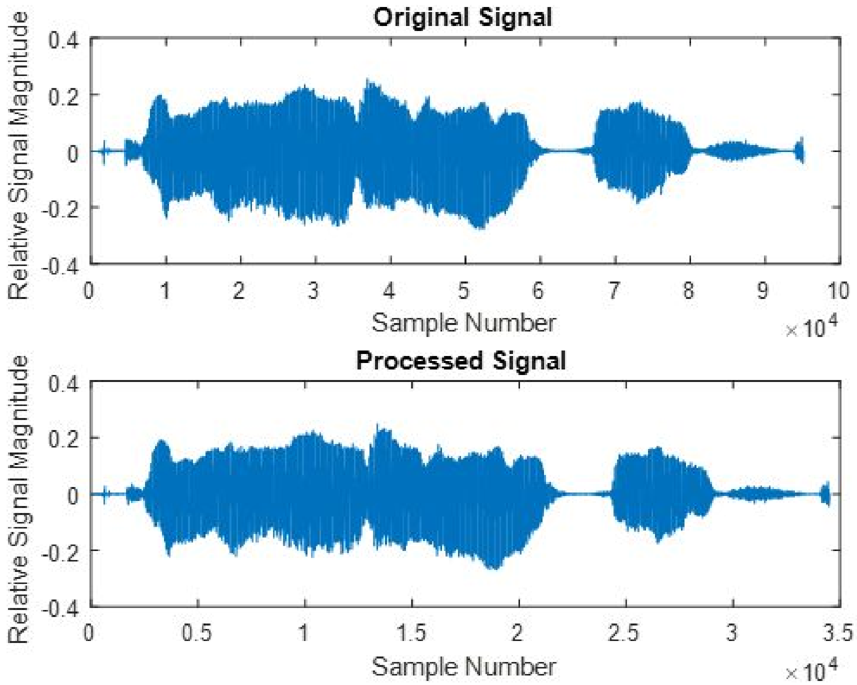

Figure 1: Processed and Unprocessed Signal of Sampled Voice Recording

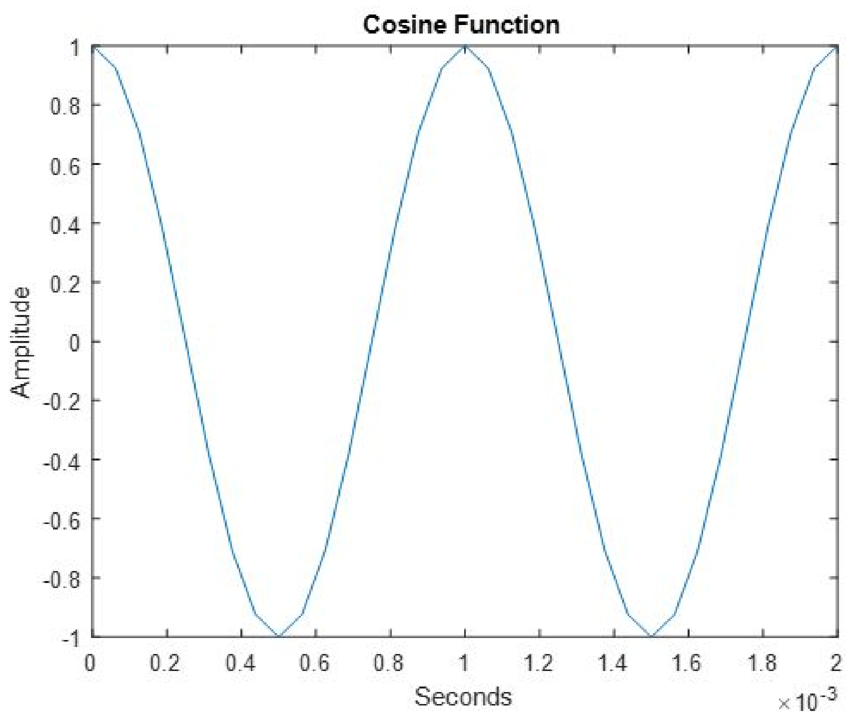

Figure 2: Cosine Signal

```
clc; close all; clear all;
```

```
filename = ('Sample2.mp3');
[reMonoY,samplesize] = Convertto16khz(filename);

FS = 16000;
Freq = 1000;                              %in Hz
dt = 1/Freq;
t = 0:(1/FS):(samplesize(1:1)/FS);
x = cos(2*pi*1000*t);

figure
plot(t,x)
xlim([0 2/Freq]);
title('Cosine Function');
xlabel('Time (s)');
ylabel('Relative Amplitude');
```

```
function [reMonoY,samplesize] = Convertto16khz(filename)

[StereoY,RealFS] = audioread(filename);
samplesize = size(StereoY);
FS = 16000;

%%%%%%%%%%%% Convert to  Mono and then Resample%%%%%%%%%%%%%
if samplesize(1,2) > 1
    MonoY = sum(StereoY,2) / size(StereoY,2);
end

reMonoY = resample(MonoY,FS,RealFS);
samplesize = size(reMonoY);

%%%%%%%%%%%% Save Audio Files %%%%%%%%%%%%%
audiowrite('write1.wav',reMonoY,FS);

%%%%%%%%%%%% Graph %%%%%%%%%%

subplot(2,1,1);
plot(MonoY)
title('Original Signal')
xlabel('Sample Number')
ylabel('Relative Signal Magnitude')

subplot(2,1,2);
plot(reMonoY)
title('Processed Signal')
xlabel('Sample Number')
ylabel('Relative Signal Magnitude')

end
```