Communication Systems

(EC - 206)

PROJECT REPORT

DELHI TECHNOLOGICAL UNIVERSITY

4TH SEMESTER

SUBMITTED TO:

DR. SACHIN TARAN

SUBMITTED BY:

MD ZEESHAN EQBAL	**2K20/EC/123**
NAVEEN BEMAD	**2K20/EC/130**

Simple Digital Hearing Aid

CONTENTS

Abstract

People with hearing impairment use devices that amplify sound to improve their hearing capability. Such devices are called Hearing aids or deaf aid. Approximately 5% of the world's population suffers from some hearing loss, but only a tiny percentage of them use a hearing aid.

In this report, we have explained the working and results of a Simple Digital Hearing Aid created in MATLAB. We can tell the difference between speech and background noise through digital technology, with the virtue of which we can filter out the unwanted background noise. In contrast, the traditional analog hearing aids are just like radio because they can only tune and adjust them for volume, bass, and treble for the sound. Through digital signal processing, digital hearing aids now offer what analog hearing aids cannot provide.

This project mainly contains a noise reduction filter that can filter out the noise from the input speech signal and adjust its gain to make it comfortable to hear for a person who has ski-slope hearing loss.

Introduction

Hearing aids are devices that deaf people mainly use to compensate for their hearing loss. A hearing aid is used, which fits the dynamic range of the speech signal into the restricted dynamic range of the impaired ear. Some sounds can be completely inaudible, whereas some can be detected because their spectra are audible. Still, the ear cannot correctly identify them because these parts are at higher frequencies. Thus, a hearing aid mainly amplifies the weak signals rather than intense sounds. Sensorineural hearing loss is one the most common type of hearing loss. People suffering from this type of loss cannot bear any loud sound. A slight increase in the sound level above the threshold can be dangerous for them, and the low-intensity sounds are inaudible completely. Thus, a hearing aid system makes the sound louder and makes the speech easy to understand the impaired people.

It picks up the sound signals with the help of a microphone, converts weak signals into loud signals, and then finally sends it to the ear through a speaker.

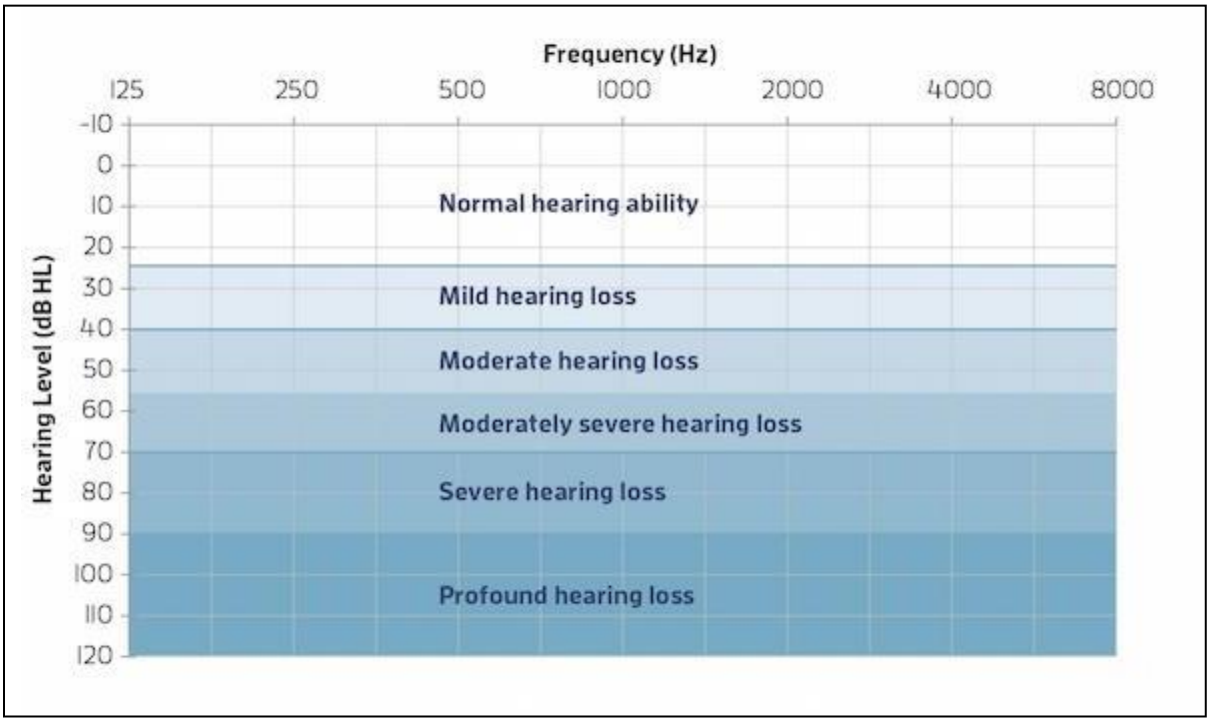


Figure 1 Different degrees of hearing loss

All hearing aids were using analog technology for the treatment of the speech signal. But through digital technology, improvements have been made for the efficiency of hearing aids. Nowadays, digital hearing aids are used because they are small and can stay hidden inside the ear. They have an almost perfect sound reproduction while also reducing the background noise.

Nowadays, a tiny programmable computer capable of amplifying millions of different sound signals had been constructed in the devices, thus improving people's hearing ability with hearing loss. Digital Hearing Aid was first launched in the mid-'80s, but these early models were slightly unpractical. Ten years later, the digital hearing aids became prosperous, with tiny digital devices placed either inside or discreetly behind the ear.

Today, digital technology is a part of our daily lifestyle. Almost every household has various digital products, such as smartphones and laptops. Hearing aids also was benefited from the emergence of digital technology, among the advantages of digital Signal Processing that allows hands-free operation. The digital aids automatically adjust the volume and pitch on their own. It performs thousands of adjustments per second, resulting in reduced background noise, improved listening in noisy situations, sound quality, and multiple program settings. The user can switch between varieties of modes to calibrate it for different listening situations.

Methodology

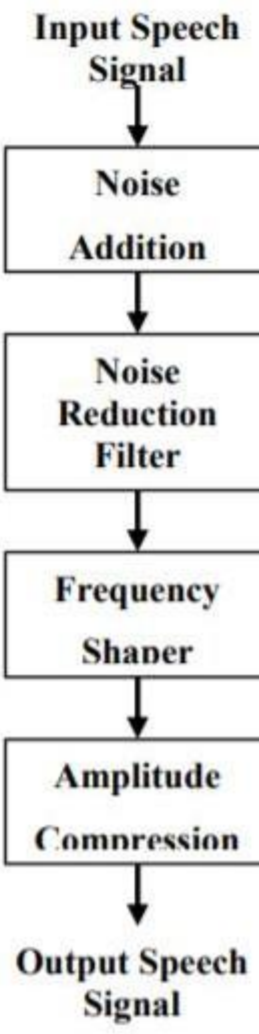


Figure 2 Flow diagram of Simple Digital Hearing Aid

1. Input

An input signal is taken with some noise to model a real-life situation.

2. Noise Addition

If the input speech signal is clean, then some noise is added. So we add the Additive White Gaussian Noise (AWGN) to the input speech signal.

3. Noise Reduction Filter

A reduction filter function is used for noise reduction in the input speech signal to eliminate all sorts of background noise. To suppress the noise in the input signal, and a wavelet filter function is used.

4. Frequency Shaper

The frequency shaper function is designed to correct for the loss of hearing at specific frequencies.

One major complaint of users of hearing aids is that the hearing aid amplifies all signals rather than the significant signal they desire to hear. Most hearing-impaired people have difficulty hearing the high-frequency signal. So the frequency shaper corrects for the loss of hearing at specific frequencies. And it does so by applying high gain for higher frequencies and vice versa.

5. Amplitude Compression

Amplitude compression function has the task of controlling the overall gain of a speech amplification system. Amplitude compression will ensure that the amplified signal will not exceed saturation power. Saturation power is where the sound signal begins to become uncomfortable to the listener.

6. Output

The output signal with increased and reduced noise can be comfortably heard by someone suffering from ski slope hearing loss.

Code

The code for the above steps of our system for the Simple Digital Hearing Aid on MATLAB is as following:

1. Denoising Function

```
function y = denoise(x,fs)
% y = denoise(x);
% method to denoise a given signal using wavelets
% x is the input Matlab sound file
%THR is the threshold, SORH is for soft or hard thresholding, KEEPAPP allows you to keep

% approximation coefficients
[thr,sorh,keepapp]=ddencmp('den','wv',x);
% returns a de-noised version xc of input signal x (our one-dimensional speech signal)
[y, ~, ~, ~, ~]=wdencmp('gbl',x,'db3',2,thr,sorh,keepapp);

% comparing noisy and denoise signal
x_length = length(x);
t=(0:1/fs:(x_length-1)/fs);
figure;
subplot(2,1,1);
plot(t,x,'r');
title('Signal with Noise');
subplot(2,1,2);
plot(t,y);
title('Signal after Denoising');
```

2. Frequency Shaping Function

```
function y = freqshape(x,g,transitionV,fs)
% y = applySkiSlope(x,g,transitionV,fs)
% Creates the gain filter for a patient with ski slope hearing loss.
% The maximum gain will be g and the minimum gain will be one.  The magnitude
% of the gain function will be the concatenation of preset piecewise functions.
% However the time of the transitions from one piecewise function to another can
% be set by the user in the elements of the transitionV.  The final frequency used
% will be fs/2 since that's the highest frequency that the input signal will contain.
% The output will be the filtered signal
first = transitionV(1);
second = transitionV(2);
third = transitionV(3);
fourth = transitionV(4);

x_length  =  length(x);
n = nextpow2(x_length);
N = 2^n;
T = 1/fs;
X = fft(x,N);
gain = zeros(N,1);

% Sets the gain for the first stage of frequencies
firstC = (.3*(g-1))/first;
k=0;
while(k/N <= first/fs)
   gain(k+1) = firstC*k/(N*T) + 1;
   gain(N-k) = gain(k+1);
   k=k+1;
end;

% Sets the gain for the second stage of frequencies
secondC = firstC*first +1;
secondC2 = (second-first)/5;
while(k/N <= second/fs)
   gain(k+1) = 1 + (secondC-1)*exp(-((k/(N*T))-first)/secondC2);
```

```
   gain(N-k) = gain(k+1);
   k=k+1;
end;

% Sets the gain for the third stage of frequencies
thirdC = 1 + (secondC-1)*exp(-second/secondC2);
thirdC2 = (third-second)/5;
while(k/N <= third/fs)
   gain(k+1) = g + (thirdC-g)*exp(-((k/(N*T)-second))/thirdC2);
   gain(N-k) = gain(k+1);
   k=k+1;
end;

% Sets the gain for the fourth stage of frequencies
while(k/N <= fourth/fs)
   gain(k+1) = g;
   gain(N-k) = gain(k+1);
   k=k+1;
end;

% Sets the gain for the fifth stage of frequencies
fifthC = g;
fifthC2 = (fs/2-fourth)/5;
while(k/N <= .5)
   gain(k+1) = 1 + (fifthC-1)*exp(-((k/(N*T))-fourth)/fifthC2);
   gain(N-k) = gain(k+1);
   k=k+1;
end;

k_v = (0:N-1)/N;
figure; %non-redundant filter transfer function
k_v = k_v*fs;
k_v = k_v(1:N/2+1);
plot(k_v,gain(1:N/2+1));
title('Frequency Shaper Transfer Function');
xlabel('Frequency (Hertz)');
ylabel('Gain');
xlim([0 10000]);
Y = X+gain; % for X refer line no.27
y = real(ifft(Y,N));
y = y(1:x_length);
```

3. Power Compression Function

```
function y = powerCompress(input, Psat,Fs)
% y = powerCompress(input, Psat,Fs)
% Takes in a a signal makes sure that the maximum power in any frequency
% is less than or equal to Psat.  Also had some denoising capabilities, by
% zeroing out very low power frequencies.
% input - input Matlab sound file
% Psat  - Saturation power
% FS    - Sampling frequency of the input signal
x=input;
len=Fs*0.1;
iter=floor(length(x)/len);
Plow=0.008;

for rg=0:1:iter;
 start=rg*len+1;
 en= rg*len+len;
 if rg*len+len>length(x)
    en=length(x);
 end

clear signal X X_pow Y_pow Y y z;
signal=x(start:en);
n = nextpow2(len);
N = 2^n;
X = fft(signal,N);
X_phase=angle(X);                 % Save the old phase information
X_pow = abs(X)/N;
Y_pow = X_pow;
Y=zeros(N,1);
for k=0:N/2
   if Y_pow(k+1)<Plow               % Take out noise
      Y_pow(k+1)=0;
```

```
      Y_pow(N-k)=0;
   elseif Y_pow(k+1)>Psat          % Clip amplitudes higher than Psat
      Y_pow(k+1)=Psat;
      Y_pow(N-k)=Psat;
   end;
    Y(k+1) = Y_pow(k+1)*(cos(X_phase(k+1))+1i*sin(X_phase(k+1)));
    Y(N-k) = Y_pow(N-k)*(cos(X_phase(N-k))+1i*sin(X_phase(N-k)));
end;
y = real(ifft(Y,N));
z = y(1:en-start+1);
sig_out(start:en)=z;
end;
y = sig_out*5000; % Multiplying 5000 is just increasing the intensity of the o/p signal
```

4. Power Compression Function

```
function y = hearingAidF(input,g,Psat,transitionV)
% input - the input signal to the system. Should be a wave file.
% g - the maximum gain that will be applied to the signal
% Psat - the cut-off power.  The output power will not be higher than this
% transitionV - 4 element vector that has the values of where the gain changes
%               to the next piecewise function

% Inputing and reading audio file
[x,fs] = audioread(input);
x = x(:, 1);
% Add Noise to the signal (Step 1)
x = awgn(x,20);
% Denoising filter (Step 2)
xc = denoise(x,fs);
% Frequency shaping filter (Step 3)
xf = freqshape(xc,g,transitionV,fs);
% Amplitude shaping filter (Step 4)
y = powerCompress(xf, Psat,fs);

% Comparing Spectrograms
figure;
subplot(2,1,1);
specgram(x);
title('Spectrogram of Original Signal 2');

subplot(2,1,2);
specgram(y);
title('Spectrogram of Adjusted Signal 2');

% Output sound (Step 5)
disp('Adjusted Sound');
sound(y,fs);
```

5. Demo File

```
clear all;
close all;
clc;

% Input sound
input = 'Sample/late_noise.wav';

% Medical Parameters
g = 50;
transitionV = [1000, 1500, 2550, 5000];
Psat=100;

% Sound Output
y = hearingAidF(input,g,Psat,transitionV);
```

Simulation Result

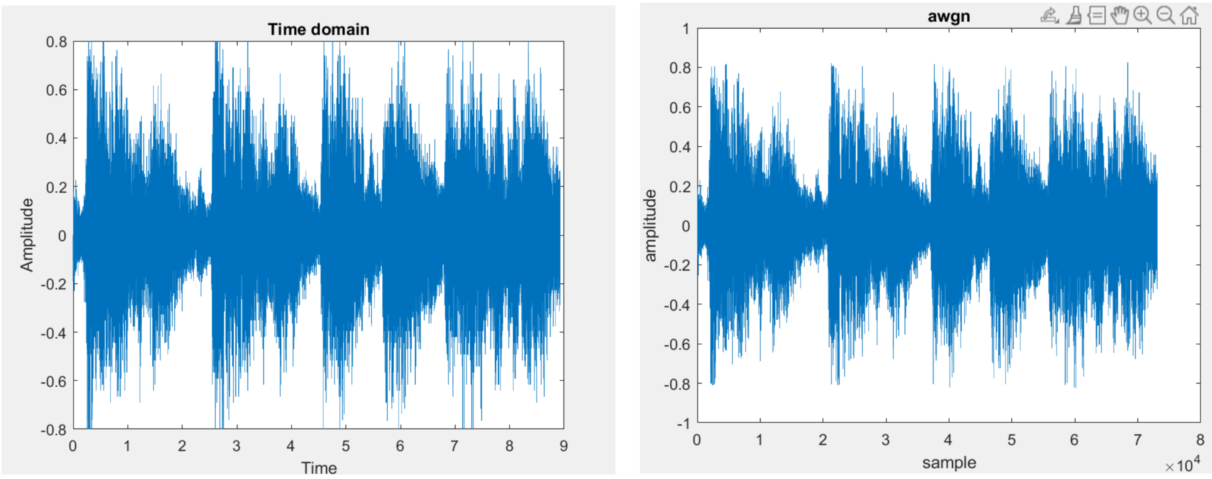
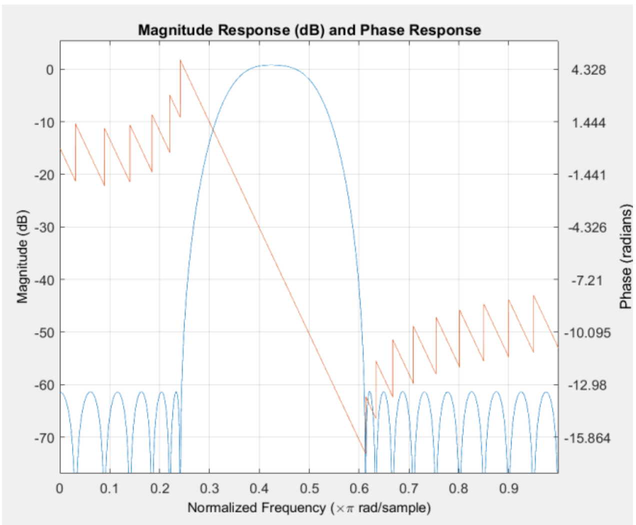


Figure 3 Comparison between Noise and Denoised Signal

Figure 3 consists of two plots in the time domain. The first plot is of the input signal which has noise in it. From observation, the time domain plot of the input signal has a substantial number of wavelets and has a maximum amplitude great than 1. The second plot is of the denoised signal that we obtain from the denoising function used in our Simple Digital Hearing Aid. The number of wavelets is substantially low and the maximum amplitude is also less than that of the input signal.

On comparing both the plots, it is very evident that the output signal has a low high Signal to noise ratio (SNR) as compared to the input signal. Therefore, the output signal is successfully denoised.



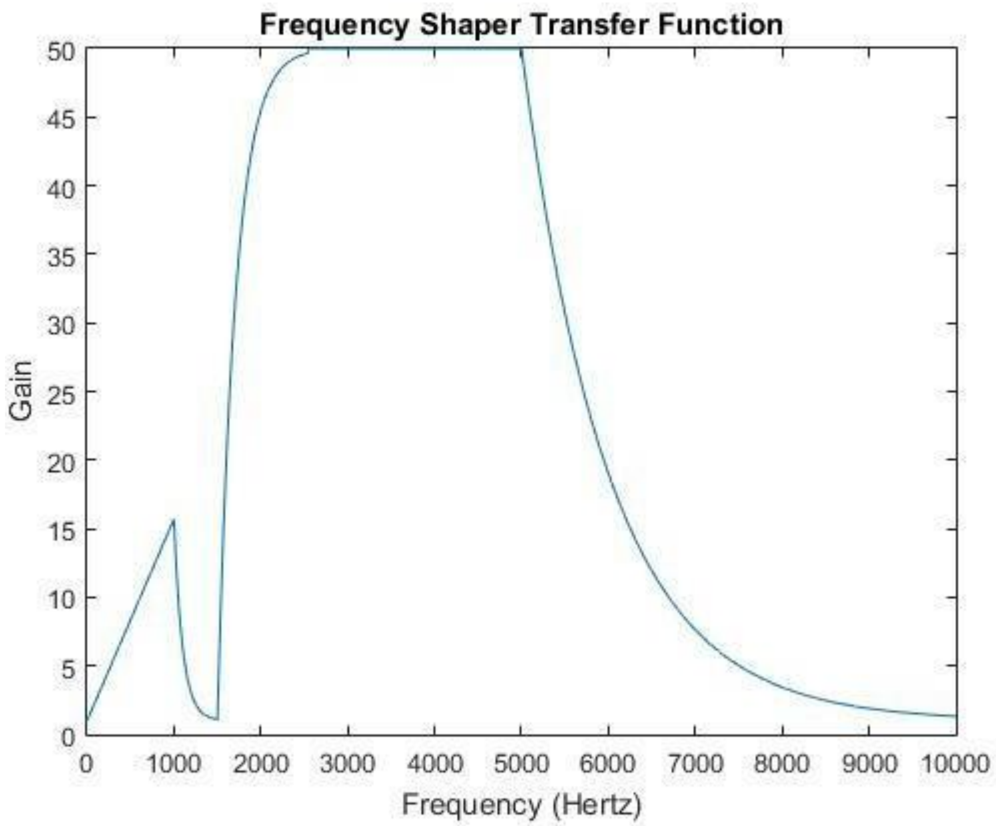


Figure 4 Frequency Shaper Function for Gain

Figure 4 showcases the frequency shaper transfer function that we obtain in the frequency shape function used in our algorithm. This function is used to add gain following the transition frequency values that are input and are based on the medical diagnosis of the patient using the heading aid.

The function has a specific shape in between the four transition frequencies and has a maximum gain amplitude equal to the gain that is needed to be added into the input signal to deal with ski-slope hearing loss. This gain helps in maintaining comfort, maintaining sound quality, and audibility enhancement in transition regions. The shape of this predefined function is medically determined.

As it can be seen from the figure, the minimum gain is unity and the maximum is 50. The curve transitions at values 1000, 1500, 2550, 5000 Hz.

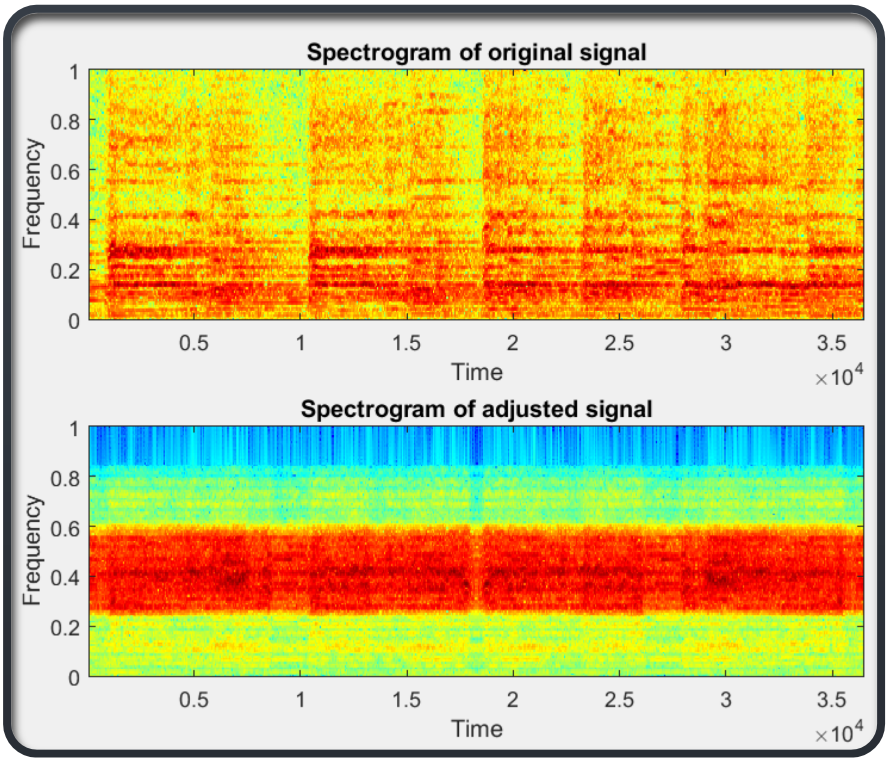


Figure 5 Comparison between Spectrogram of Actual and Output Signal

A spectrogram is a visual depiction of the spectrum of frequencies present in the signal waveform across the time domain. It showcases the strength or loudness of the signal through this graph.

Figure 5 showcases the spectrograms of the original signal and the final obtained signal from our system. The spectrograms, on the comparison, show how the output signal is denoised as it has low strength of redundant frequencies present across the time axis. The majority of intensity lies in a particular spectrum of frequencies where important useful information exists that has to be carried by the signal.

Moreover, it is visible that the spectrum in which the strength of the signal lies is clipped at a certain value showcasing that the power is saturated and the loudness is thresholded.

This difference in the two spectrograms can also be analyzed by comparing the audios of input and output. The input audio contains a lot of noise with the loudness of speech variable and uncontrolled. The output audio is denoised and compressed at a certain strength.

Conclusion

In this project, we have implemented a Simple Digital Hearing Aid in MATLAB. Through this system, we were able to simulate a hearing aid that is much better than an analog hearing aid that is conventionally used. The digital hearing aid builds on the shortcoming of the analog hearing aid and can filter the useful information in the audio signal and denoise or remove the useless information. Moreover, it can control the loudness of speech in the audio signal without amplifying the noise present. This makes it a perfect replacement for an analog hearing aid.

Through our simulations, we were able to showcase the denoising capabilities of the system at various stages of the algorithm. We also showcased the use of Frequency shaping to provide gain to the signal and Power compression to support amplitude shaping.

Finally, we were able to execute an algorithm that could work in coherence with the medical requirements to limit the loudness and make the signal comfortable to hear for someone who has hearing loss. It also denoises the redundant parts of the signal without making the output shrill or painful to hear.

References

- http://www.isca.in/IJES/Archive/v5/i6/5.ISCA-RJEngS-2016-076.pdf
- https://github.com/divyaKh/DigitalHearingAid
- https://www.hear-it.org/ski-slope-hearing-loss-1#:~:text=What%20is%20ski%20slope%20hearing,or%20high%20pitched%20female%20voices.
- https://www.hearingreview.com/inside-hearing/research/compression-in-hearing-aids-why-fast-multichannel-processing-systems-work-well#:~:text=Fast%20multichannel%20amplitude%20compression%20in,the%20processing%20reduces%20speech%20intelligibility.
- https://www.audiologyonline.com/articles/complex-versus-standard-fittings-part-21836-21836
- https://floridamedicalhearing.com/main/hearing-aids/difference-analog-digital-hearing-aids/