Military Institute of Science & Technology (MIST)

Department of Electrical, Electronic and Communication Engineering (EECE)

PROJECT REPORT

COURSE CODE: EECE 312

COURSE TITLE: DIGITAL SIGNAL PROCESSING-1 LAB

SEMESTER: FALL 2022

GROUP NO : 3

SECTION A

Submitted By :

Gourab Datta (202016010)

Lamiya Mahbub (202016016)

Mehrab Ibne Masud Badhon (202016033)

Naimujjaman (202016027)

Juthika Alamgir Orpa (202016013)

Muntasir Ahmed (201816148)

OBJECTIVE : **Digital Hearing Aid compensating for hearing loss of different level.**

WORKING PRINCIPLE:

A person is said to have hearing loss if they are not able to hear as well as someone with normal hearing, meaning hearing thresholds of 20 dB or better in both ears. It can be mild, moderate, moderately severe, severe or profound, and can affect one or both ears. Hearing aids are needed for most people on the basis of how much of hearing loss they are suffering from.

People with normal hearing can hear sounds between 20 and 20,000 Hz. Some people are not able to hear a certain frequency range which must be amplified by a gain that makes them those frequency audible.

CLASSIFICATION OF HEARING LOSS	HEARING LEVEL
Normal Hearing	-10dB -26dB
Mild Hearing Loss	27dB-40dB
Moderate Hearing Loss	40dB-70dB
Severe hearing loss	70dB – 90dB
Profound Hearing loss	>90dB

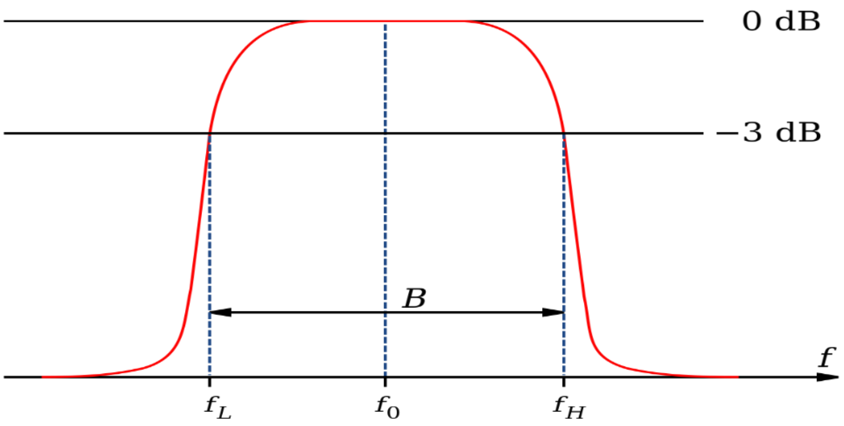
Hearing aids work by amplifying sound through a three-part system:

1.The microphone receives sound and converts it into a digital signal.

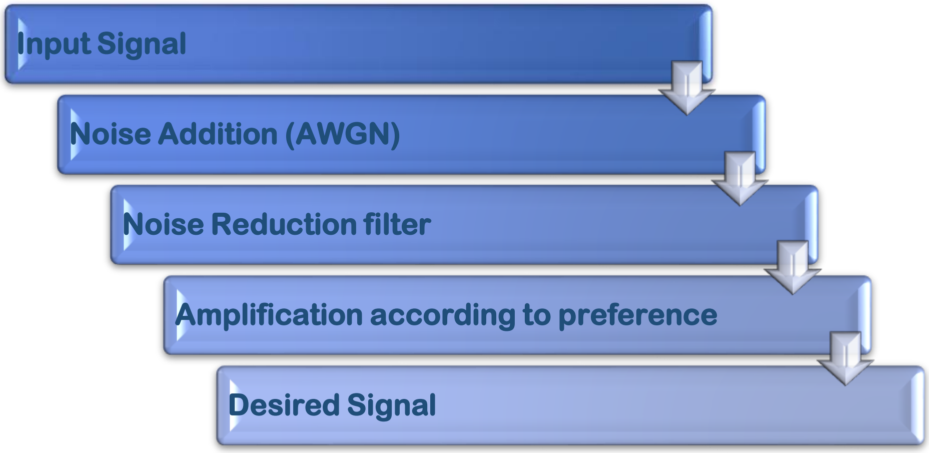
2.the amplifier increases the strength of the digital signal

3.The speaker produces the amplified sound into the ear.

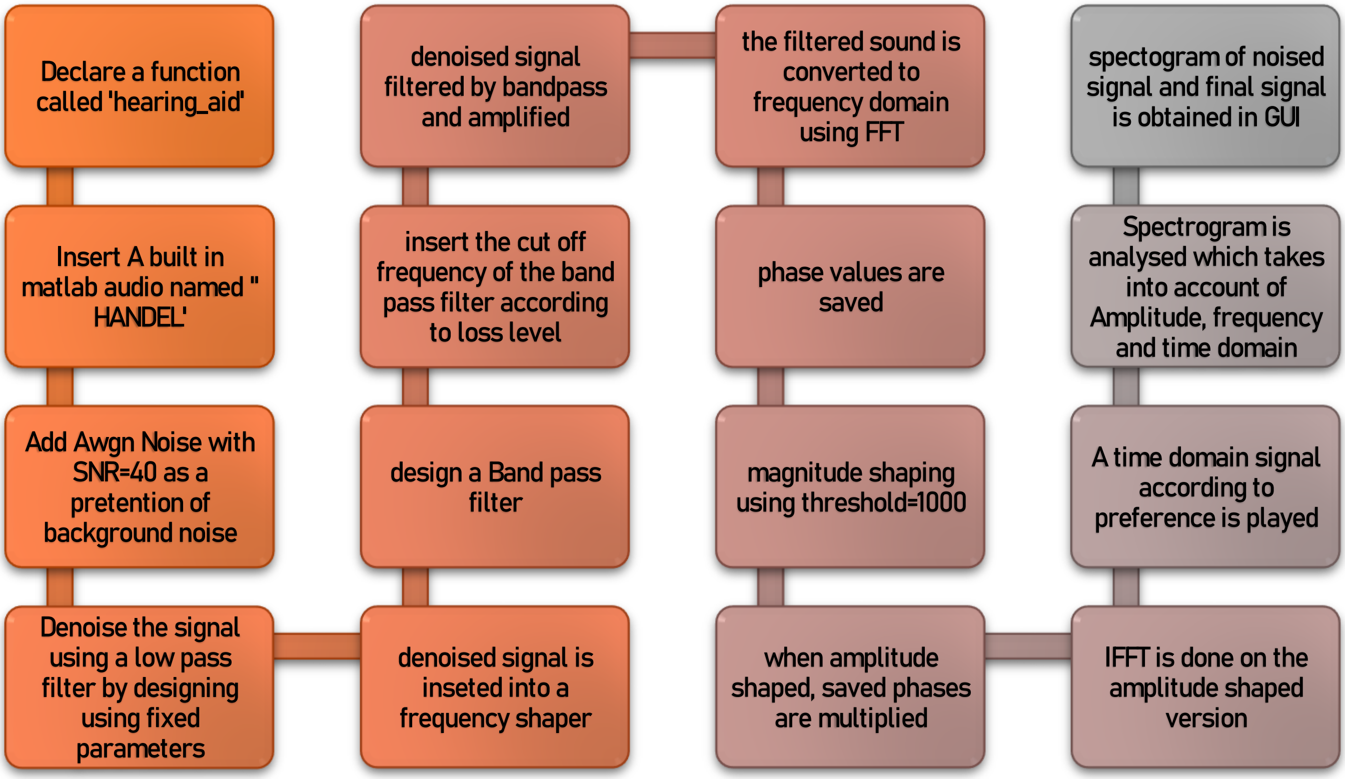
So for different people with different level of hearing problem require certain level of denoising and amplification based on their preference of frequency. A dynamic hearing aid is thus crucial for overcoming the shortcomings of this disease.



BLOCK DIAGRAM:



ALGORITM:


Declare a function called 'hearing_aid'
Insert A built in matlab audio named " HANDEL'
Add Awgn Noise with SNR=40 as a pretention of background noise
Denoise the signal using a low pass filter by designing using fixed parameters
denoised signal is inseted into a frequency shaper
design a Band pass filter
insert the cut off frequency of the band pass filter according to loss level
denoised signal filtered by bandpass and amplified
the filtered sound is converted to frequency domain using FFT
phase values are saved
magnitude shaping using threshold=1000
when amplitude shaped, saved phases are multiplied
IFFT is done on the amplitude shaped version
A time domain signal according to preference is played
Spectrogram is analysed which takes into account of Amplitude, frequency and time domain
spectogram of noised signal and final signal is obtained in GUI

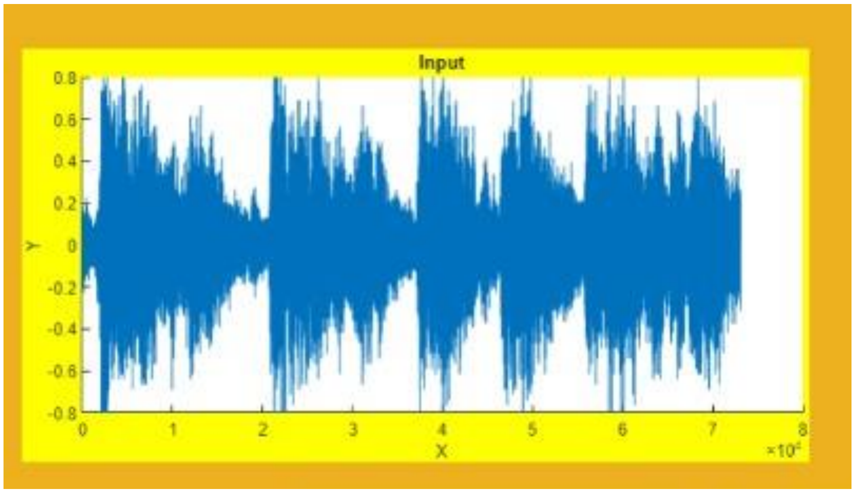
METHODOLOGY:

```
classdef app2 < matlab.apps.AppBase
% Properties that correspond to app components
properties (Access = public)
UIFigure matlab.ui.Figure
UIAxes matlab.ui.control.UIAxes
```

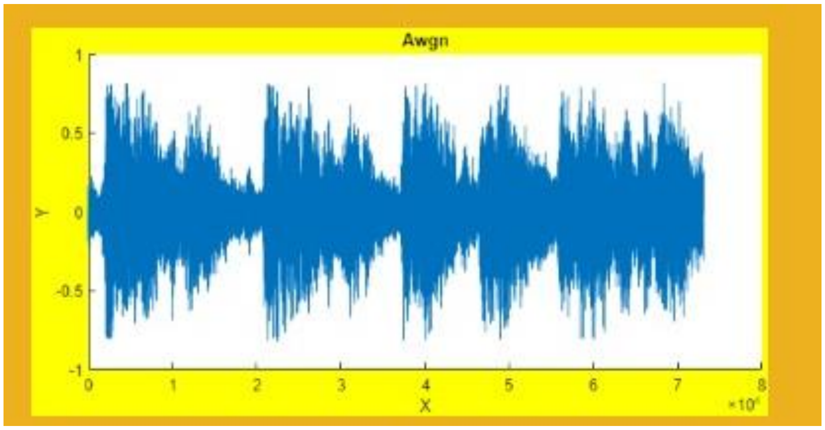
```
UIAxes2 matlab.ui.control.UIAxes
UIAxes3 matlab.ui.control.UIAxes
UIAxes4 matlab.ui.control.UIAxes
UIAxes5 matlab.ui.control.UIAxes
RUNButton matlab.ui.control.Button
Fst1EditFieldLabel matlab.ui.control.Label
Fst1EditField matlab.ui.control.NumericEditField
Fp1EditFieldLabel matlab.ui.control.Label
Fp1EditField matlab.ui.control.NumericEditField
Fp2EditFieldLabel matlab.ui.control.Label
Fp2EditField matlab.ui.control.NumericEditField
Fst2EditFieldLabel matlab.ui.control.Label
Fst2EditField matlab.ui.control.NumericEditField
Ast1EditFieldLabel matlab.ui.control.Label
Ast1EditField matlab.ui.control.NumericEditField
ApEditFieldLabel matlab.ui.control.Label
ApEditField matlab.ui.control.NumericEditField
Ast2EditFieldLabel matlab.ui.control.Label
Ast2EditField matlab.ui.control.NumericEditField
StateEditFieldLabel matlab.ui.control.Label
StateEditField matlab.ui.control.EditField
end
methods (Access = private)
function hearing_aid(app,Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2) %code declared as function

%% BOLDED VARIABLES ARE OUR USER INPUT
%% Insert a sample sound which is audible to normal people :

app.StateEditField.Value=('input sound')%shows in state when input sound is playing
load handel.mat
fs=Fs;%imprting data got from handel.mat
y = y(:, 1);%for 1 sec
plot(app.UIAxes,y);%uiaxes shows the graph in matlab gui
app.StateEditField.Value=('playing actual sound...')
sound(y);
pause(10)
```



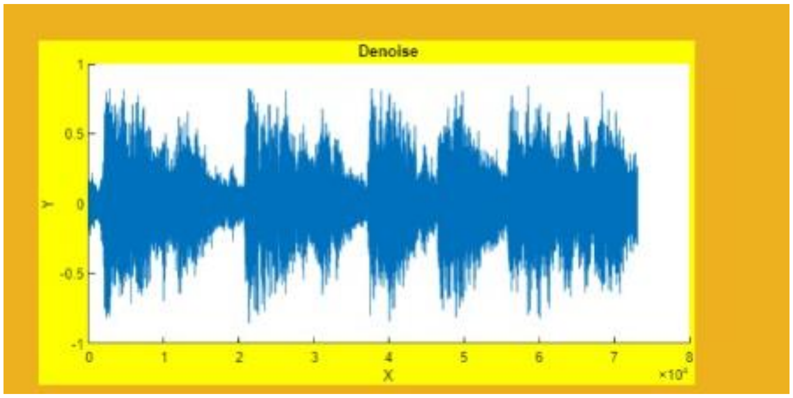
```
%% adding noise to the audio using awgn keeping the SNR=40
y = awgn(y,40);
noi = y;
plot(app.UIAxes2,y);
app.StateEditField.Value=('playing added noise...');%shows in state bar in gui
sound(y);%playing the sound
pause(10)%pausing the sound in 10 sec
```



```
%% denoising process using a low pass filter using a cutoff
hlpf = fdesign.lowpass('Fp,Fst,Ap,Ast',3.0e3,3.5e3,0.5,50,fs);%designing lowpass filter using buitin function
```

[Fp= passband frequency
Fst= stopband frequency
Ap=passband ripple
Ast = stopband ripple]

```
D = design(hlpf);%lowpass filter
%freqz(D);%needed if we want the spectrum of the filter
x = filter(D,y);%filtering the noised signal
app.StateEditField.Value=('playing denoised sound');%showed in state while playing the sound
plot(app.UIAxes3,x);% plotting in the 3rd graph in gui
sound(x,fs);%playing the denoised sound
pause(10)%pause after 10 sec
```



```
%% frequency shaper using band pass of denoised signal
T = 1/fs;%setting time
len = length(x);
p = log2(len);%level division denoised signal
p = ceil(p);% making p an integer
N = 2^p;% number of level
f1 = fdesign.bandpass(Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2,5*fs);%setting the values according to the requirement
hd = design(f1,'equiripple');%bandpass function
y = filter(hd,x);%denoised signal filtered by bandpass
%freqz(hd); %filter frequency response if needed
y = y*100;%amplification
app.StateEditField.Value=('playing frequency shaped...');
sound(y,fs);%playing the freq shaped sound
pause(10);%pause after 10 sec
```

here frequency shaping is done by ceiling the values of the denoised signals and passing it through a bandpass filter which will be according to the requirement of the user. Once passed through the filter, the filtered values are amplified and played for 10 second. This will give us very high pitched sound due to amplification

----

```
%% Amplitude Shaping of Frequency Shaped Signal Using Threshold


app.StateEditField.Value=('amplitude shaper')
out1=fft(y);
phse=angle(out1);%saving the phase values before amp shaping
mag=abs(out1)/N;%magnitude of freq shaped signal
[magsig,~]=size(mag);
threshold=1000;
out=zeros(magsig,1);
for i=1:magsig/2
if(mag(i)>threshold)
mag(i)=threshold;mag(magsig-i)=threshold;
end
out(i)=mag(i)*exp(j*phse(i));%setting the magnitudes same as the previous position by multiplying exp phase
out(magsig-i)=out(i);
end
```

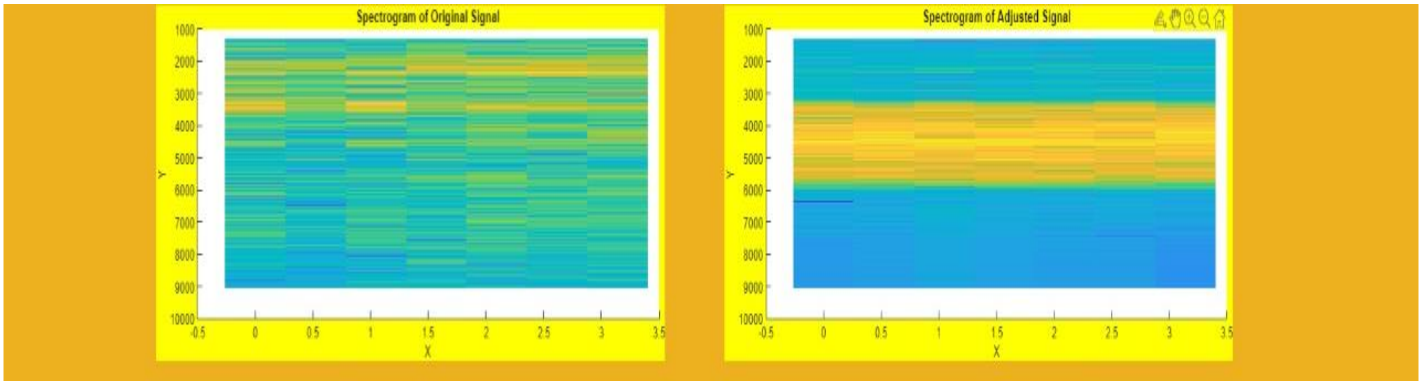
now the frequency shaped signal is fast fourier transformed. Which we know contains magnitude and phase. We keep the phases intact and take the absolute value of the signal. A threshold is set because every device needs one now the magnitude is compared to the threshold. If it is greater than threshold, we keep it as value of threshold, if lower, we keep it as that. Finally we multiply the value of the phases we saved prior in order to complete the amplitude shaping.

```
%%FINAL OUTPUT in time domain using IFFT
outfinal=real(ifft(out))*10000;%taking only real value after ifft of the amp shaped signal
app.StateEditField.Value=('playing amplitude shaped...');
sound(outfinal,fs); %playing the finally edited sound
pause(10);%pause after 10 sec

load handel.mat
%%code for spectrogram

myAxe = app.UIAxes4;
[S,F,T] = spectrogram(noi);
imagesc(myAxe, F, T, log(abs(S))); %plot the log spectrum as image ( noise added signal)
myAxe1 = app.UIAxes5;
[S1,F1,T1] = spectrogram(outfinal);
imagesc(myAxe1, F1, T1, log(abs(S1))); %plot the log spectrum as image ( final signal )
app.StateEditField.Value=('Thank you');
end
end
```

The spectrogram is a three-dimensional plot of signal amplitude versus time and frequency. From the first spectrogram it is evident that the NOISE and the ACTUAL signal are merged into each other. While the refined 2nd version shows the signal (yellow color) completely isolated from the noise.



Imagesc creates a picture in the myaxe direction. Each element inside this command is gives a specific color. Mixing here are the amplitude, frequency and time of the signals.

```
% Callbacks that handle component events
methods (Access = private)
% Button pushed function: RUNButton
function RUNButtonPushed(app, event)
Fst1=app.Fst1EditField.Value;
Fp1=app.Fp1EditField.Value;
Fp2=app.Fp2EditField.Value;
Fst2=app.Fst2EditField.Value;
Ast1=app.Ast1EditField.Value;
Ap=app.ApEditField.Value;
Ast2=app.Ast2EditField.Value;
hearing_aid(app,Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2)
end
end
%%code ends here
```

MATLAB BUILT IN FUNCTION FOR GUI

```
% Component initialization
methods (Access = private)
% Create UIFigure and components
function createComponents(app)
% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Color = [0.9294 0.6941 0.1255];
app.UIFigure.Position = [100 100 1141 792];
app.UIFigure.Name = 'MATLAB App';
% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, 'Input')
xlabel(app.UIAxes, 'X')
ylabel(app.UIAxes, 'Y')
app.UIAxes.BackgroundColor = [1 1 0];
app.UIAxes.Position = [15 571 300 185];
% Create UIAxes2
app.UIAxes2 = uiaxes(app.UIFigure);
title(app.UIAxes2, 'Awgn')
xlabel(app.UIAxes2, 'X')
ylabel(app.UIAxes2, 'Y')
app.UIAxes2.BackgroundColor = [1 1 0];
app.UIAxes2.Position = [375 570 300 185];
% Create UIAxes3
app.UIAxes3 = uiaxes(app.UIFigure);
title(app.UIAxes3, 'Denoise')
xlabel(app.UIAxes3, 'X')
ylabel(app.UIAxes3, 'Y')
app.UIAxes3.BackgroundColor = [1 1 0];
app.UIAxes3.Position = [733 572 300 185];
% Create UIAxes4
app.UIAxes4 = uiaxes(app.UIFigure);
title(app.UIAxes4, 'Spectrogram of Original Signal')
```

```
xlabel(app.UIAxes4, 'X')
ylabel(app.UIAxes4, 'Y')
app.UIAxes4.BackgroundColor = [1 1 0];
app.UIAxes4.Position = [209 366 300 185];


% Create UIAxes5
app.UIAxes5 = uiaxes(app.UIFigure);
title(app.UIAxes5, 'Spectrogram of Adjusted Signal')
xlabel(app.UIAxes5, 'X')
ylabel(app.UIAxes5, 'Y')
app.UIAxes5.BackgroundColor = [1 1 0];
app.UIAxes5.Position = [590 366 300 185];
% Create RUNButton
app.RUNButton = uibutton(app.UIFigure, 'push');
app.RUNButton.ButtonPushedFcn = createCallbackFcn(app, @RUNButtonPushed, true);
app.RUNButton.BackgroundColor = [1 1 0.0667];
app.RUNButton.FontWeight = 'bold';
app.RUNButton.Position = [509 262 99 22];
app.RUNButton.Text = 'RUN';
% Create Fst1EditFieldLabel
app.Fst1EditFieldLabel = uilabel(app.UIFigure);
app.Fst1EditFieldLabel.BackgroundColor = [0.9294 0.6941 0.1255];
app.Fst1EditFieldLabel.HorizontalAlignment = 'right';
app.Fst1EditFieldLabel.FontWeight = 'bold';
app.Fst1EditFieldLabel.FontAngle = 'italic';
app.Fst1EditFieldLabel.Position = [229 199 30 22];
app.Fst1EditFieldLabel.Text = 'Fst1';
% Create Fst1EditField
app.Fst1EditField = uieditfield(app.UIFigure, 'numeric');
app.Fst1EditField.FontWeight = 'bold';
app.Fst1EditField.BackgroundColor = [1 1 0];
app.Fst1EditField.Position = [194 170 100 22];
% Create Fp1EditFieldLabel
app.Fp1EditFieldLabel = uilabel(app.UIFigure);
app.Fp1EditFieldLabel.BackgroundColor = [0.9294 0.6941 0.1255];
app.Fp1EditFieldLabel.HorizontalAlignment = 'right';
app.Fp1EditFieldLabel.FontWeight = 'bold';
app.Fp1EditFieldLabel.FontAngle = 'italic';
app.Fp1EditFieldLabel.Position = [544 199 27 22];
app.Fp1EditFieldLabel.Text = 'Fp1';
% Create Fp1EditField
app.Fp1EditField = uieditfield(app.UIFigure, 'numeric');
app.Fp1EditField.FontWeight = 'bold';
app.Fp1EditField.BackgroundColor = [1 1 0];
app.Fp1EditField.Position = [509 170 100 22];
% Create Fp2EditFieldLabel
app.Fp2EditFieldLabel = uilabel(app.UIFigure);
app.Fp2EditFieldLabel.BackgroundColor = [0.9294 0.6941 0.1255];
app.Fp2EditFieldLabel.HorizontalAlignment = 'right';
app.Fp2EditFieldLabel.FontWeight = 'bold';
app.Fp2EditFieldLabel.FontAngle = 'italic';
app.Fp2EditFieldLabel.Position = [870 199 27 22];
app.Fp2EditFieldLabel.Text = 'Fp2';


% Create Fp2EditField
app.Fp2EditField = uieditfield(app.UIFigure, 'numeric');
app.Fp2EditField.FontWeight = 'bold';
app.Fp2EditField.BackgroundColor = [1 1 0];
app.Fp2EditField.Position = [833 170 99.9398166600239 22];
% Create Fst2EditFieldLabel
app.Fst2EditFieldLabel = uilabel(app.UIFigure);
app.Fst2EditFieldLabel.BackgroundColor = [0.9294 0.6941 0.1255];
app.Fst2EditFieldLabel.HorizontalAlignment = 'right';
app.Fst2EditFieldLabel.FontWeight = 'bold';
app.Fst2EditFieldLabel.FontAngle = 'italic';
app.Fst2EditFieldLabel.Position = [229 127 30 22];
app.Fst2EditFieldLabel.Text = 'Fst2';
% Create Fst2EditField
app.Fst2EditField = uieditfield(app.UIFigure, 'numeric');
app.Fst2EditField.FontWeight = 'bold';
app.Fst2EditField.BackgroundColor = [1 1 0];
app.Fst2EditField.Position = [194 106 100 22];
% Create Ast1EditFieldLabel
app.Ast1EditFieldLabel = uilabel(app.UIFigure);
app.Ast1EditFieldLabel.BackgroundColor = [0.9294 0.6941 0.1255];
app.Ast1EditFieldLabel.HorizontalAlignment = 'right';
app.Ast1EditFieldLabel.FontWeight = 'bold';
app.Ast1EditFieldLabel.FontAngle = 'italic';
app.Ast1EditFieldLabel.Position = [542 127 32 22];
app.Ast1EditFieldLabel.Text = 'Ast1';
% Create Ast1EditField
app.Ast1EditField = uieditfield(app.UIFigure, 'numeric');
app.Ast1EditField.FontWeight = 'bold';
app.Ast1EditField.BackgroundColor = [1 1 0];
app.Ast1EditField.Position = [509 106 100 22];
% Create ApEditFieldLabel
app.ApEditFieldLabel = uilabel(app.UIFigure);
app.ApEditFieldLabel.BackgroundColor = [0.9294 0.6941 0.1255];
app.ApEditFieldLabel.HorizontalAlignment = 'right';
app.ApEditFieldLabel.FontWeight = 'bold';
app.ApEditFieldLabel.FontAngle = 'italic';
app.ApEditFieldLabel.Position = [871 127 25 22];
app.ApEditFieldLabel.Text = 'Ap';
% Create ApEditField
app.ApEditField = uieditfield(app.UIFigure, 'numeric');
app.ApEditField.FontWeight = 'bold';
 app.ApEditField.BackgroundColor = [1 1 0];
app.ApEditField.Position = [834 106 99.9398166600239 22];
```

```
 % Create Ast2EditFieldLabel
app.Ast2EditFieldLabel = uilabel(app.UIFigure);
app.Ast2EditFieldLabel.BackgroundColor = [0.9294 0.6941 0.1255];
 app.Ast2EditFieldLabel.HorizontalAlignment = 'right';
app.Ast2EditFieldLabel.FontWeight = 'bold';
app.Ast2EditFieldLabel.Position = [541 56 32 22];
app.Ast2EditFieldLabel.Text = 'Ast2';
% Create Ast2EditField
app.Ast2EditField = uieditfield(app.UIFigure, 'numeric');
app.Ast2EditField.FontWeight = 'bold';
app.Ast2EditField.BackgroundColor = [1 1 0];
app.Ast2EditField.Position = [509 35 100 22];
% Create StateEditFieldLabel
app.StateEditFieldLabel = uilabel(app.UIFigure);
app.StateEditFieldLabel.BackgroundColor = [0.9294 0.6941 0.1255];
app.StateEditFieldLabel.HorizontalAlignment = 'right';
app.StateEditFieldLabel.FontWeight = 'bold';
app.StateEditFieldLabel.FontAngle = 'italic';
app.StateEditFieldLabel.Position = [419 317 35 22];
app.StateEditFieldLabel.Text = 'State';
% Create StateEditField
app.StateEditField = uieditfield(app.UIFigure, 'text');
app.StateEditField.BackgroundColor = [1 1 0];
app.StateEditField.Position = [469 317 224 22];
% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end
% App creation and deletion
methods (Access = public)
% Construct app
function app = app2
% Create UIFigure and components
createComponents(app)
% Register the app with App Designer
registerApp(app, app.UIFigure)
if nargout == 0
clear app
end
end
% Code that executes before app deletion
function delete(app)
% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
```

PERFORMANCE ANALYSIS:

Scenario 1:

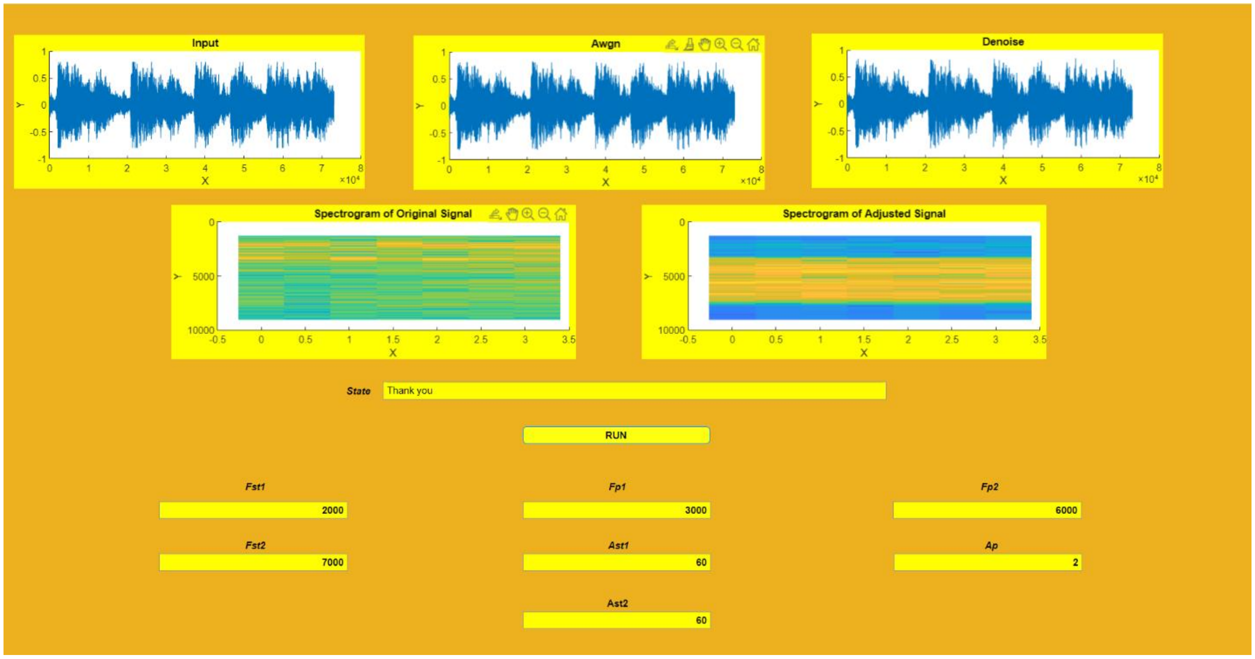
A patient has moderate hearing loss at medium frequency range of 3000-6000Hz. So the required gain is 40db-70db .

So the patient cannot hear this range, using the aid, we push the clear sound within that range with a gain of within 40-70db.

So we give input of

starting passband frequency ,Fp1= 3000Hz

Ending passband frequency Fp2= 6000Hz



State: Thank you

Fst1: 2000

Fp1: 3000

Fp2: 6000

Fst2: 7000

Ast1: 60

Ap: 2

Ast2: 60

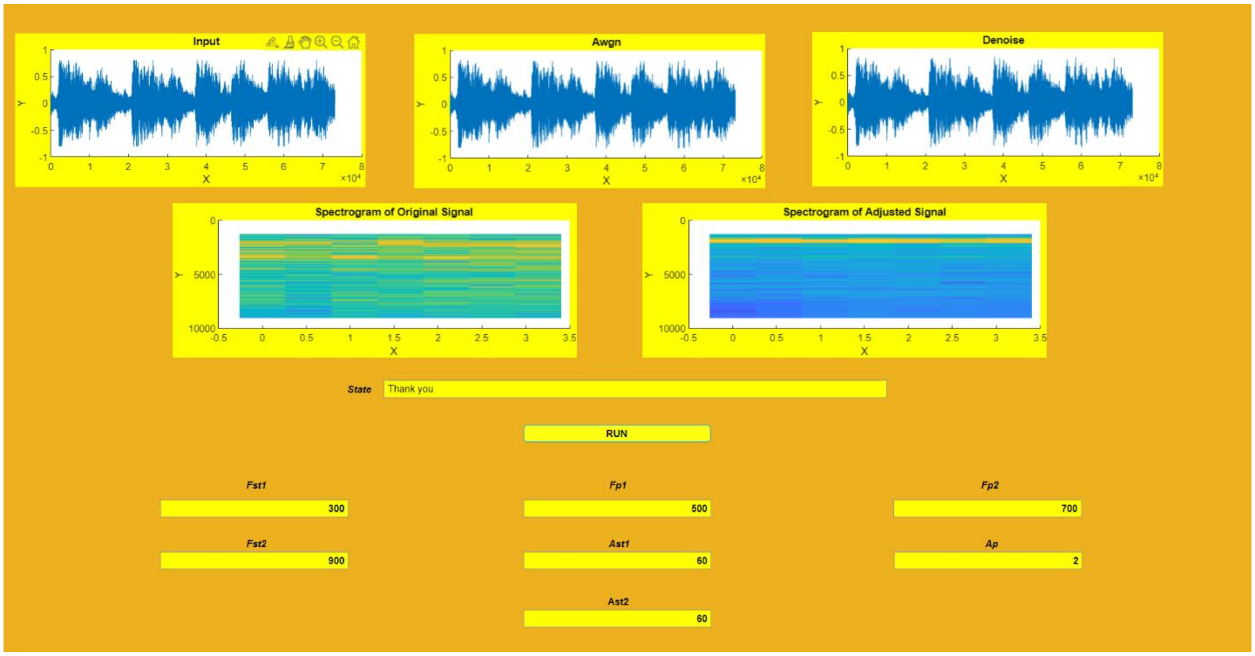
Scenario 2:

A patient has severe hearing loss at low frequency range of 500-900Hz . So the required gain is 70db-90db .

So the patient cannot hear this range, using the aid, we push the clear sound within that range with a gain of within 70-90db.

starting passband frequency ,Fp1= 500Hz

Ending passband frequency Fp2= 900Hz

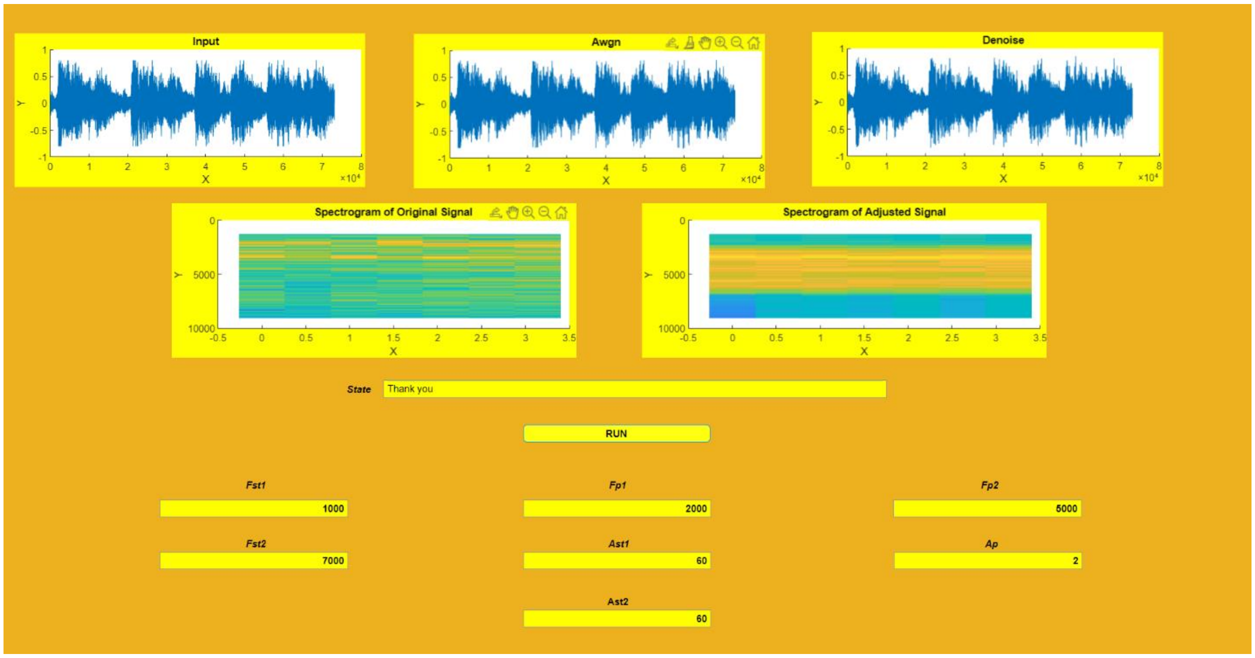

Input
Awgn
Denoise
Spectrogram of Original Signal
Spectrogram of Adjusted Signal
X
Y
×10⁴
RUN

State: Thank you

Fst1	Fp1	Fp2
300	500	700
Fst2	Ast1	Ap
900	60	2
	Ast2	
	60	

Scenario 3:

A patient has mild hearing loss at frequency range of 2000-5000Hz . So the required gain is 27db-40db .


Input
Awgn
Denoise
Spectrogram of Original Signal
Spectrogram of Adjusted Signal
X
Y
×10⁴
RUN

State: Thank you

Fst1	Fp1	Fp2
1000	2000	5000
Fst2	Ast1	Ap
7000	60	2
	Ast2	
	60	

So the patient cannot hear this range, using the aid, we push the clear sound within that range with a gain of within 27-40db.

starting passband frequency, Fp1= 2000Hz

Ending passband frequency Fp2=5000Hz

Scenario 4:

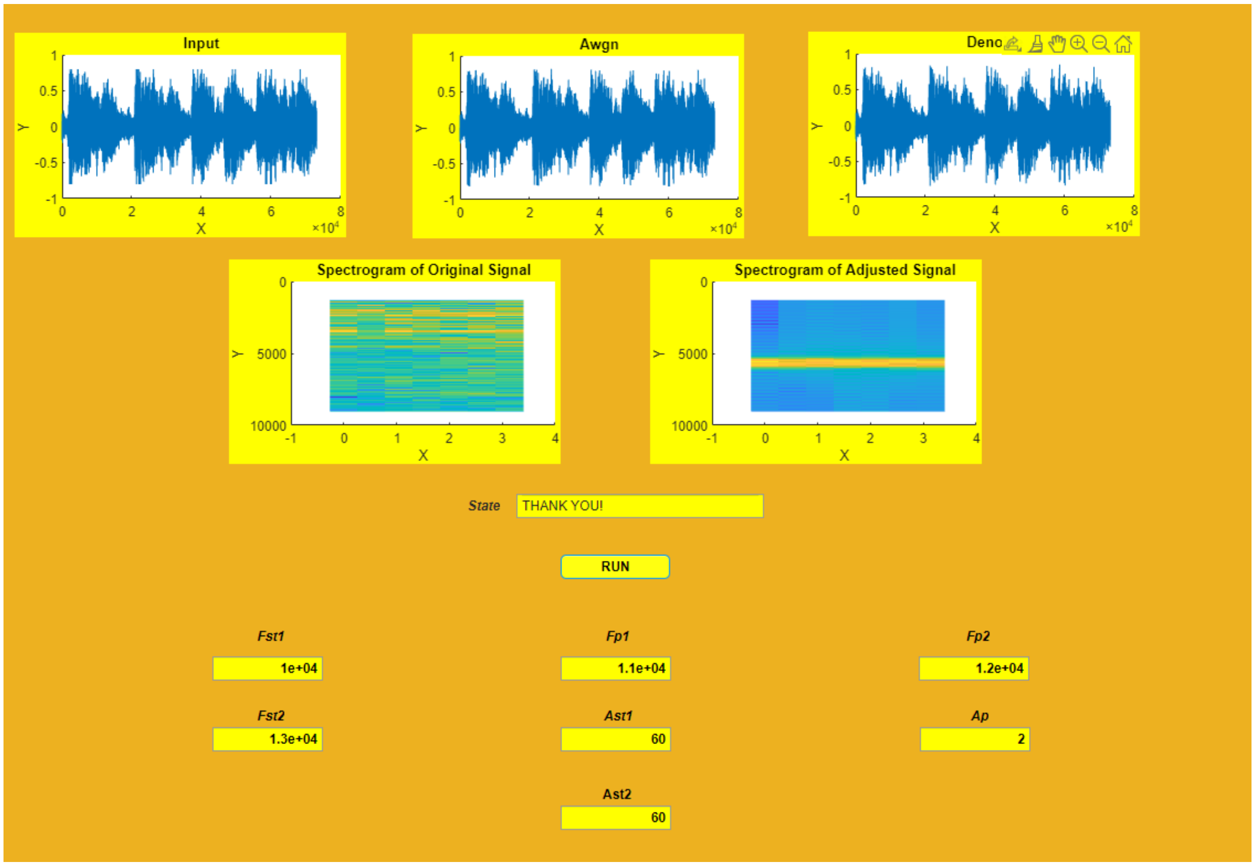
A patient has profound hearing loss at high frequency range of 11000-12000Hz. So the required gain is >90dB .

So the patient cannot hear this range, using the aid, we push the clear sound within that range with a gain of >90dB.

So we give input of

starting passband frequency ,Fp1= 11000Hz

Ending passband frequency Fp2= 12000Hz


Input
Awgn
Deno
1
0.5
0
-0.5
-1
Y
0
2
4
6
8
X
×10⁴
Spectrogram of Original Signal
Spectrogram of Adjusted Signal
0
5000
10000
Y
-1
0
1
2
3
4
X
RUN

State: THANK YOU!

Fst1: 1e+04

Fp1: 1.1e+04

Fp2: 1.2e+04

Fst2: 1.3e+04

Ast1: 60

Ap: 2

Ast2: 60

DISCUSSIONS:

- It was observed, from the above investigation that, by using the different functions like denoising, frequency shaping, amplitude limiter on speech signal, we can make the signal to be more suitable for hearing impaired people with less complexity and more flexibility.

- The gain on specific frequencies has been enhanced by adding frequency transfer/shaping function with the input discrete signal. T
- he frequency transfer/shaping function has been designed according to patient's requirement/hearing loss.
- In this project, four types of subjects have been taken with different types of hearing losses and tried to reduce the losses by providing gain on that particular frequency ranges with other desired functions

REFERENCES:

- http://en.wikipedia.org/wiki/Deafness
- http://www.hearingloss.org/content/types-causes-andtreatment
- Azhar Latif, Digital Hearing Aid Chip
- T.-B. Deng, "Three-channel variable filter-bank for digital hearing aids",published by IEEE,2009.
- http://www.clear.rice.edu/elec301/Projects01/dig_hear_aid/testingandresults.html