Digital Hearing Aid System for Hearing Impaired

Vaishnavi P. Mane, Vaishnavi R. Mane, Aishwarya Menon, Gauri Bhalsing

Course instructor- Dr. Ashwini Deshpande

E&TC Department, MKSSS's CCEW, Pune

Email ID: vaishnavi.p.mane@cumminscollege.in, ashwini.deshpande@cumminscollege.in

Problem statement:

- According to biological concepts, the Hearing process is very much important for human beings for every second. After hearing a sound our brain processes that information and then we start doing activities with the help of our body.
- When we hear very low volume sound or information at that time we aren't able to understand what we should do next.
- Low volume sound can be originally low sound but sometimes it is loud and we can't hear properly so this problem is known as hearing loss.
- Hearing loss is a measure of shift in the auditory system compared to that of normal ear for detection of a pure tone. The huge scale of the human population suffers from hearing loss.

Introduction:

- One of the most important issues for human beings is aid in hearing. They are actually small electronic instruments which make sound louder and make speech easier to hear and understand.
- Hearing loss is typically measured as the shift in auditory threshold relative to that of a normal ear for detection of a pure tone.
- The aim of the hearing aid is to amplify sound signals in such a way that they become audible for the hearing impaired person.
- With the availability of modern day technologies and the recent developments in the signal processing area, sophisticated artificial hearing aid systems can be designed that relax the job of damaged auditory systems to a great extent and make much of the sound available to the hearing impaired.
- In this project, the simulation of the simple digital hearing aid is developed in the MATLAB programming language.

- The implementation of this configurable digital hearing aid (DHA) system includes noise reduction filter, frequency-dependent amplification and amplitude compression.
- We will test our filters on a mock patient and successfully reduce white Gaussian noise, increase the gain for frequencies which were difficult to hear, and shape the amplitude to prevent any of the frequencies from becoming too loud.

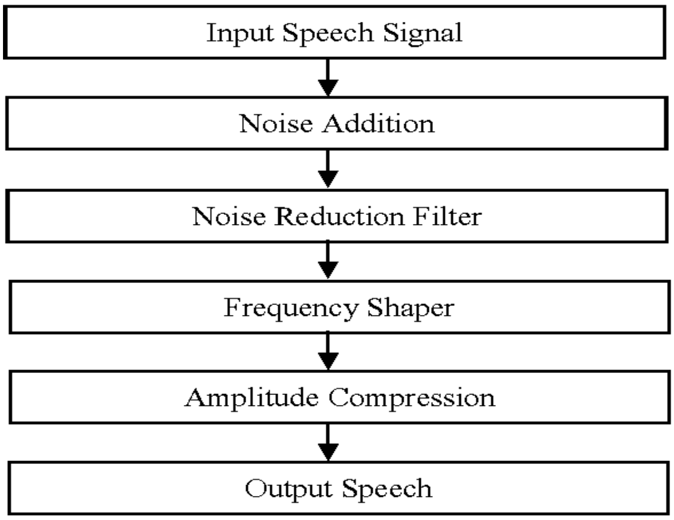
Literature Survey:

S. No	Title of Paper	Name of Conference/ Journal	Year of Publication	Summary and Results
1	Digital filtering in hearing aid system for the hearing impaired	IEEE, ICEEOT	2016	Block diagram representation of the hearing aid system.
2	Enhancement of speech signals for Hearing Aid Devices using digital Signal processing	IEEE, 4th ISMSIT	2013	The digital hearing aids can be programmed to match patients' hearing individually according to a specific Frequency. The aids are programmed using the human audiogram. Digital hearing aids work with very low power batteries, approximately in mW.
3	Comparative analysis of wavelet transform filtering systems for noise reduction	PLoS ONE journal	2022	The selection of WT system settings significantly affects the efficiency of de-noising procedures.

4	Denoising Speech Signals for Digital Hearing aids: A wavelet Based Approach	ResearchGate	2011	This paper describes the research developing a wavelet based, single microphone noise reduction algorithm for use in digital hearing aids.

Methodology:

- **System Block Diagram:**

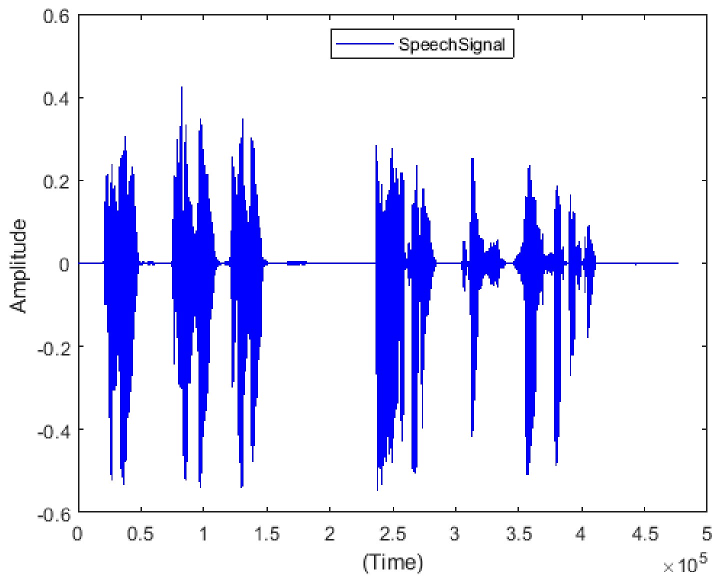


- **Study of real-life signal characteristics:**

The speech is a one - dimensional function of time. Speech signal is a non-stationary signal. The generally accepted standard range of **Audible frequencies is 20 to 20,000 Hz**, although the range of frequencies individuals hear is greatly influenced by environmental factors. Frequencies below 20 Hz are generally felt rather than heard, assuming the amplitude of the vibration is great enough. Frequencies above 20,000 Hz can sometimes be sensed by young people. High frequencies are the first to be affected by hearing loss due to age and/or prolonged exposure to very loud noises.

Classification of Hearing loss	Range
Normal hearing	-10 dB to 26 dB
Mild hearing loss	27 dB to 40 dB
Moderate hearing loss	40 dB to 70 dB
Severe hearing loss	70 dB to 90 dB

Waveform of Speech Signal



- **Selection of sampling rate:**

We have used standard sampling rate for recorded signal i.e. speech signal

Fs = 8000 Hz

For inbuilt chirp signal Fs = 8192 Hz

- **DSP Technique used:**

- Discrete Wavelet Transform(DWT) -

  Discrete wavelet transform is used by wavelet filters. This transforms Analyze the signal into different frequencies at different resolutions. Wavelet transform function is

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right)$$

- Fast Fourier Transform (FFT) -
  It is used to compare with DWT. DFT equation used in FFT is $X(k) = \sum x(n)W_N^{nk}$
- To decide which filter gives better signal we have used Hamming and Hanning Window Technique of FIR filter as well as Butterworth and Chebyshev window technique of IIR.
- Hamming window function

  $$H(\theta) = 0.54 + 0.46\ \cos\left[\left(\frac{2\pi}{N}\right)n\right]$$

- Hanning window function

  $$w(n) = 0.5\left(1 - \cos\left(2\pi\frac{n}{N}\right)\right),\quad 0 \leq n \leq N.$$

- **Detail Design (steps):**

  The input speech signal takes the form of a human voice. For producing an adjusted output speech signal which can be audible to the hearing impaired person.
  The input speech signal will pass through several functions i.e., noise addition, noise reduction filter, frequency shaper and amplitude compression before producing an adjusted output speech signal which is audible to the hearing impaired person.

- **Noise Addition -**
  In this system, To simulate a real situation, the Additive White Gaussian Noise (AWGN) and random noise are added to the input speech signal. The noise (AWGN) has a continuous and uniform frequency spectrum over a specified frequency band and has equal power per Hertz of this band. It consists of all frequencies at equal intensity and has a normal (Gaussian) probability density function.
  Equation for AWGN -
  Y= s + 10^(-Eb_N0_dB/20)*n;
  where ‘s’ is the transmitted sequence, Eb_N0_dB is the SNR and ‘n’ is the Additive White Gaussian Noise.

- **Noise Reduction Filter -**

  A major anxiety for the people with hearing loss is the capability of hearing aids to differentiate intended speech signals in a noisy environment. Hence, to eliminate the noise, a reduction filter function is used in this design. To suppress the noise in the signal, the wavelet filter function is used.

  Reason behind Wavelet filter -

  - To reduce the noise of speech signals many techniques are available like digital filters (FIR or IIR), adaptive methods and wavelet transform thresholding methods. Digital filters and adaptive methods can be applied to signals whose statistical characteristics are stationary in many cases. Recently the wavelet transform has been proven to be a useful tool for non-stationary signal analysis.
  - Wavelet filters localize features in our data to different scales, we can preserve important signals while removing noise. The basic idea behind wavelet denoising is that the wavelet transform leads to a sparse representation for many real-world signals. Means the wavelet transform concentrates signal in a few large-magnitude wavelet coefficients. Wavelet coefficients which are small in value are typically noise and you can "shrink" those coefficients or remove them without affecting the signal. After we threshold the coefficients, we reconstruct the data using the inverse wavelet transform

  Mathematical expression for Wavelet filter is -

$$D\boxed{[a,b]} = \frac{1}{\sqrt{b}} \sum_{m=0}^{p-1} f[t_m]\psi\left[\frac{t_m - a}{b}\right]$$

a = τ
b = s

$$\boxed{a = k2^{-j} \quad b = 2^{-j}}$$

- **Frequency Shaper -**

  One major complaint of hearing aid users is that the hearing aid amplifies all signals rather than the significant signal that they desire to hear. Most hearing impaired people have difficulties hearing high frequency signals. Therefore, the frequency shaper is designed to correct for loss of hearing at certain frequencies. It applies high gain for higher frequencies and vice versa.

Bandpass filter is used as a frequency shaper. Some specifications of the filter are decided by reading many research papers and observing audiograms. So for severe hearing loss specifications are as follows

First stopband frequency - 2000 Hz

First passband frequency - 3000 Hz

Second passband frequency - 4000 Hz

Second stopband frequency - 5000 Hz

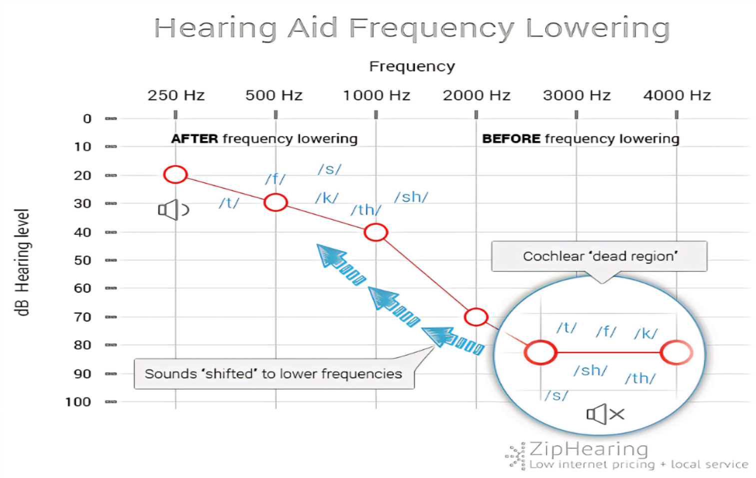
Passband ripple - 60 dB

Stopband attenuation - 2 dB

Order of filter

$$N = 2^{\log2(\text{length}(\text{signal}))}$$

Image of audiogram -



- **Amplitude Compression -**

  Amplitude compression function is 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.

  When we compared the neural code for speech with normal hearing and hearing loss at the same high sound levels produced by the hearing aid, we found that the activity patterns elicited by different speech syllables were equally distinguishable with or without hearing loss.

We finally concluded that rather than suffering from supra-threshold effects of hearing loss, listeners with hearing aids are simply experiencing the same rollover effect as those with normal hearing.

And when we looked back through the literature on previous studies of speech perception with normal hearing and hearing loss at high sound levels, we found plenty of evidence to support this conclusion.

Algorithm/ Flowchart:

1. Take the input audio
2. Plot the audio signal (call it original signal)
3. Add the **AWGN noise** and plot it again.
4. Denoise the audio signal using a **wavelet filter** with desired specifications.
5. To increase the pitch of the sound, use **frequency shaper**.
6. To make sure that output power does not exceed a given saturation level after increasing the frequency, **an amplitude shaper** is used.

Results:

Programming Tool Selected:

MATLAB - It is used to implement and test algorithm.

Advantages -

- Symbolic computation can be easily done
- Call external libraries
- Perform extensive data analysis and visualization
- Develop application with graphics user interface

Wavelet Toolbox - Wavelet Denoising signal app

Advantages - Provides apps and functions for analyzing and synthesizing signals and images.

GUI

Advantages -

- Very much helpful in visualizing the output
- Easy to use and Attractive

Program/Code:

```
clc

clf

clear

close all
```

Input Audio Signal

```
promptMessage = sprintf('Would you like to record your own audio for further processing?');

titleBarCaption = 'Record Sound';

button = questdlg(promptMessage, titleBarCaption, 'Yes', 'No', 'Yes');

if strcmpi(button, 'Yes')

  recObj = audiorecorder;

  recDuration = 5;

  disp('Start Speaking');

  recordblocking(recObj, recDuration);

  disp('End recording');

  disp('Playing Recorded Audio...');

  y = getaudiodata(recObj);

  figure,plot(y);

  title('Input Audio Signal');
```

```
  xlabel('Time');

  ylabel('Amplitude');

  play(recObj);

  %pause(10);

else

  disp('In-built sound track playing...')

  load chirp.mat  %Pre-defined matlab audio

  figure,plot(y);

  title('Input Audio Signal');

  xlabel('Time');

  ylabel('Amplitude');

  fs=Fs;

  disp('The sampling frequency of the signal:');

  disp(fs);

  sound(y);

  pause(5);

end
```

Adding Noise to the Audio Signal

```
promptMessage = sprintf('Do you want me to add noise to the audio file?');

titleBarCaption = 'Adding Noise';

button = questdlg(promptMessage, titleBarCaption, 'Yes', 'No', 'Yes');

if strcmpi(button, 'Yes')

  noi = awgn(y,40);

  sound(noi);
```

```
figure,plot(noi);

title('Noisy Audio Signal');

xlabel('Time');

ylabel('Amplitude');

y_fft = fft(y);

fy = (0:length(y_fft)-1)*8000/length(y_fft);

noi_fft = fft(noi);

fnoi = (0:length(noi_fft)-1)*8000/length(noi_fft);

[cy1, cy2] = dwt(y,'sym4');

fy2 = (0:length(cy1)-1)*8000/length(cy1);

[cnoi1, cnoi2] = dwt(noi,'sym4');

fnoi2 = (0:length(cnoi1)-1)*8000/length(cnoi1);

figure

subplot(3,2,1);

plot(y);

xlabel('Time (s)')

ylabel('Amplitude')

title('Time Domain of Audio')

subplot(3,2,2);

plot(noi);

xlabel('Time (s)')

ylabel('Amplitude')

title('Time Domain of Audio+Noise')

subplot(3,2,3);
```

```
  plot(fy, abs(y_fft));
  xlabel('Frequency (Hz)')
  ylabel('Magnitude')
  title('FFT of Original Signal')
  subplot(3,2,4);
  plot(fnoi,abs(noi_fft));
  xlabel('Frequency (Hz)')
  ylabel('Magnitude')
  title('FFT of Audio+Noise Singal')
  subplot(3,2,5);
  plot(fy2,abs(cy1));
  xlabel('Frequency (Hz)')
  ylabel('Magnitude')
  title('DWT of Original Signal')
  subplot(3,2,6);
  plot(fnoi2,abs(cnoi1));
  xlabel('Frequency (Hz)')
  ylabel('Magnitude')
  title('DWT of Audio+Noise Singal')
else
  noi = y;
end
```

Denoising the Audio Signal

```
promptMessage = sprintf('Would you like to recover the original signal?');
titleBarCaption = 'Denoise the Noisy Signal';
button = questdlg(promptMessage, titleBarCaption, 'Yes', 'No', 'Yes');
if strcmpi(button, 'Yes')
  [thr,sorh,keepapp] = ddencmp('den','wv',noi);
  denoi = wdencmp('gbl',noi,'sym4',5,thr,sorh,keepapp);
  figure
  subplot(3,1,1);
  plot(y);
  title('Original Signal')
  subplot(3,1,2);
  plot(noi);
  title('Noisy Signal')
  subplot(3,1,3);
  plot(denoi);
  title('Denoised Signal')
  sound(denoi);
else
end
```

Wavelet Comparison (with FIR & IIR)

```
PlayAgain = true;
while PlayAgain
  promptMessage = sprintf('With what would you like to compare the Wavelet Filter?');
```

```
titleBarCaption = 'Comparison of Wavelet, FIR & IIR Filters';

button = questdlg(promptMessage, titleBarCaption, 'FIR', 'IIR', 'FIR');

if strcmpi(button, 'FIR')

    promptMessage = sprintf('Choose one FIR Filter.');

    titleBarCaption = 'Wavelet vs. FIR';

    button = questdlg(promptMessage, titleBarCaption, 'Hamming', 'Hanning',
'Hamming');

    if strcmpi(button, 'Hamming')

        Whamm = hamming(5);

        FIR_denoi = filter(Whamm, 1, noi);

    else

        Whan = hanning(5);

        FIR_denoi = filter(Whan, 1, noi);

    end

    figure

    subplot(2,2,1);

    plot(y);

    title('Original Signal')

    subplot(2,2,2);

    plot(noi);

    title('Noisy Signal')

    subplot(2,2,3);

    plot(denoi);

    title('Denoised by Wavelet Filter')

    subplot(2,2,4);
```

```
    plot(FIR_denoi);
    title('Denoised by FIR Filter')
    sound(FIR_denoi);
  else
    promptMessage = sprintf('Choose one IIR Filter.');
    titleBarCaption = 'Wavelet vs. IIR';
    button = questdlg(promptMessage, titleBarCaption, 'Butterworth', 'Chebyshev', 
'Butterworth');
    if strcmpi(button, 'Butterworth')
      n = 5; %order of filter
      %Wn = (700*2)/8000;
      Wn = 0.6;
      [b,a] = butter(n,Wn);
      %figure, freqz(b,a,1024,8000);
      IIR_denoi = filter(b,a,noi);
    else
      n = 5;
      %Wp = 2000*2/8000;
      Wp = 0.7;
      Rp = 0.5;
      [b,a] = cheby1(n,Rp,Wp);
      %figure, freqz(b,a,1024,8000);
      IIR_denoi = filter(b,a,noi);
    end
    figure
```

```
    subplot(2,2,1);
    plot(y);
    title('Original Signal')
    subplot(2,2,2);
    plot(noi);
    title('Noisy Signal')
    subplot(2,2,3);
    plot(denoi);
    title('Denoised by Wavelet Filter')
    subplot(2,2,4);
    plot(IIR_denoi);
    title('Denoised by IIR Filter')
    sound(IIR_denoi);
  end
  promptMessage = sprintf('Do you want to compare with any other filter?');
  titleBarCaption = 'Continue?';
  button = questdlg(promptMessage, titleBarCaption, 'Yes', 'No', 'Yes');
  if strcmpi(button, 'No')
    playAgain = false;
    break;
  else
    playAgain = true;
  end
end
```

Amplify the Signal (using Frequency and Amplitude Shaper)

```
amplifyAgain = true;
while amplifyAgain
  promptMessage = sprintf('Do you want to amplify the signal?');
  titleBarCaption = 'Amplify Signal';
  button = questdlg(promptMessage, titleBarCaption, 'Yes', 'No', 'Yes');
  if strcmpi(button, 'Yes')
     promptMessage = sprintf('For whom do you want to amplify the signal?');
     titleBarCaption = 'Frequency & Amplitude Shaping';
     button = questdlg(promptMessage, titleBarCaption,'For Hearing Impaired ppl', 'For Normal Person', 'For Hearing Impaired ppl');
     if strcmpi(button, 'For Hearing Impaired ppl')
        % freq shaper using bandpass
        fs = 8000;
        T = 1/fs;
        len = length(denoi);
        disp(len);
        p = log2(len);
        p = ceil(p);
        N = 2^p;
        disp(N);
        f1 = fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2',2000,3000,4000,5000,60,2,60,2*fs);
```

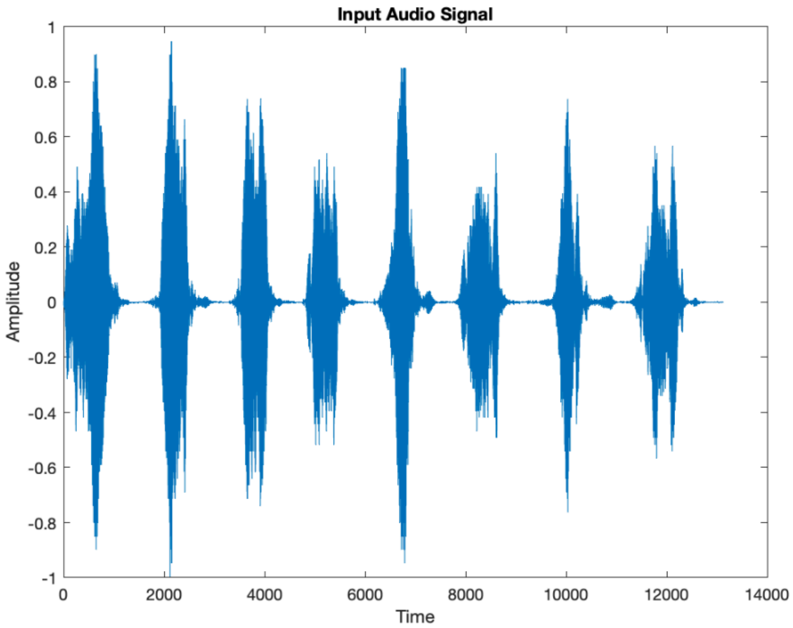
```
hd = design(f1);

yfinal = filter(hd, denoi);

freqz(hd);

yfinal = yfinal*100;

disp('playing frequency shaped...');

sound(yfinal,fs);

pause(10);

% amplitude compression

disp('amplitude compression')

out1=fft(yfinal);

%[out1, out2] = dwt(yfinal,'sym4');

phase=angle(out1);

mag=abs(out1)/N;

plot(mag);

[magsig,~]=size(mag);

disp(magsig);

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*phase(i));
```

```
            out(magsig-i)=out(i);
        end
        %out_idwt = idwt(out1,out2,'sym4');
        %outfinal = real(out_idwt(out))*1000;
        outfinal=real(ifft(out))*10000;
        disp('playing amplitude shaped...');
        sound(outfinal,fs);
        pause(10);
    else
        promptMessage = sprintf('By how much do you want to amplify the signal?');
        %titleBarCaption = 'Amplification Value';
        answer = inputdlg(promptMessage);
        dB = str2double(answer);
        fs = 8192;
        convertFromDb = 10^(dB/20);
        outfinal = denoi * convertFromDb;
        figure
        subplot(2,1,1);
        plot(denoi);
        xlabel('Time')
        ylabel('Amplitude')
        title('Denoised Signal')
        subplot(2,1,2);
        plot(outfinal);
```

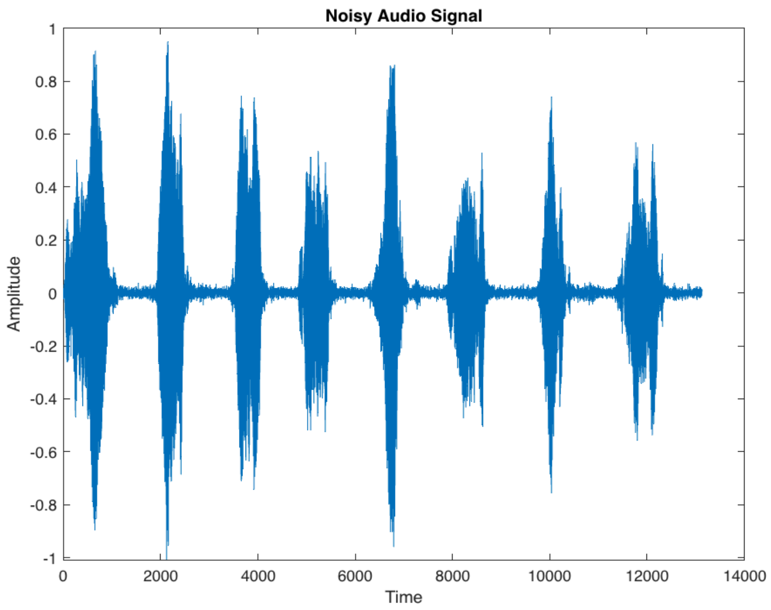
```
        xlabel('Time')
        ylabel('Amplitude')
        title('Amplified Signal')
        sound(outfinal,fs);
    end
    figure;
    subplot(2,1,1);
    specgram(denoi);
    title('Spectrogram of Original Signal');
    subplot(2,1,2);
    specgram(outfinal);
    title('Spectrogram of Adjusted Signal');
  else
  end
  promptMessage = sprintf('Do you wish to amplify the signal again?');
  titleBarCaption = 'Continue?';
  button = questdlg(promptMessage, titleBarCaption, 'Yes', 'No', 'Yes');
  if strcmpi(button, 'No')
    amplifyAgain = false;
    break;
  else
    amplifyAgain = true;
  end
end
```

Simulation Results:

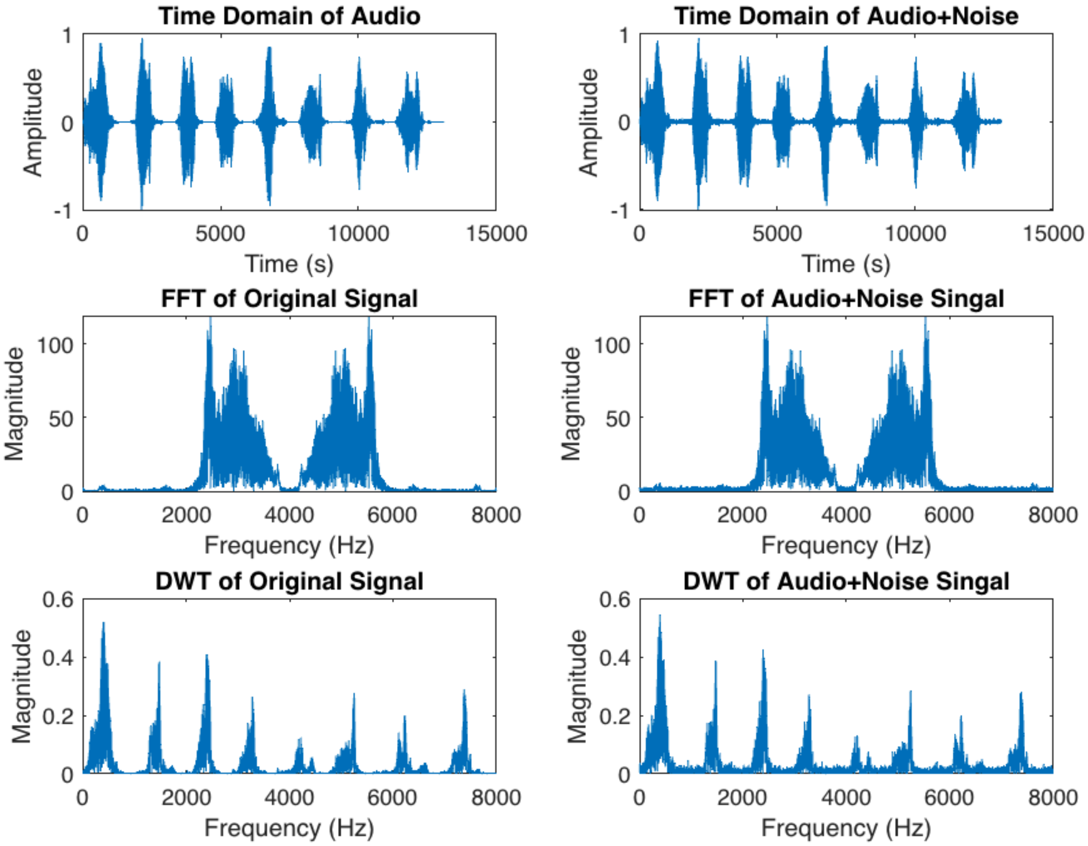
Input Audio Signal -



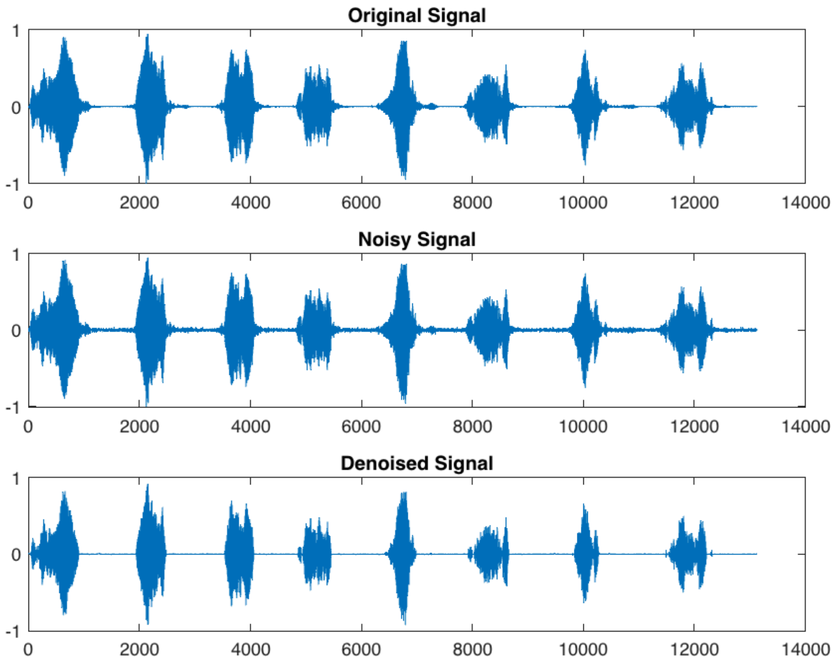
Input Audio Signal + Noise -



FFT & DWT on input as well as noisy signal -

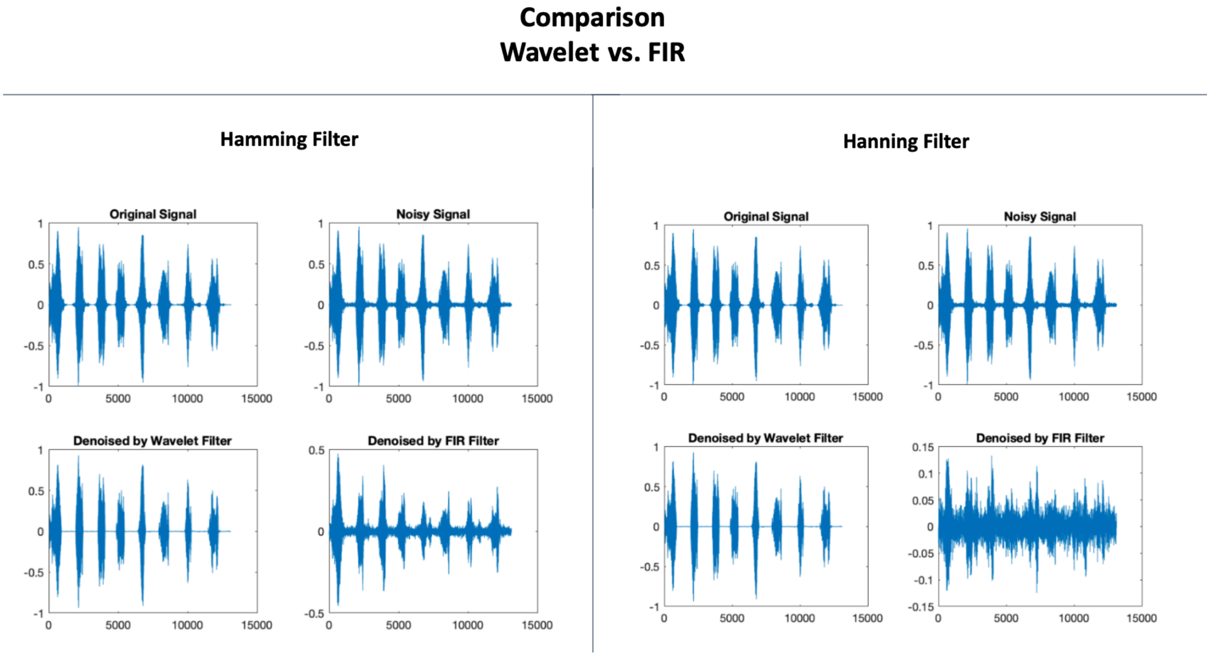


Denoised signal (using Wavelet Filter 'sym4') -

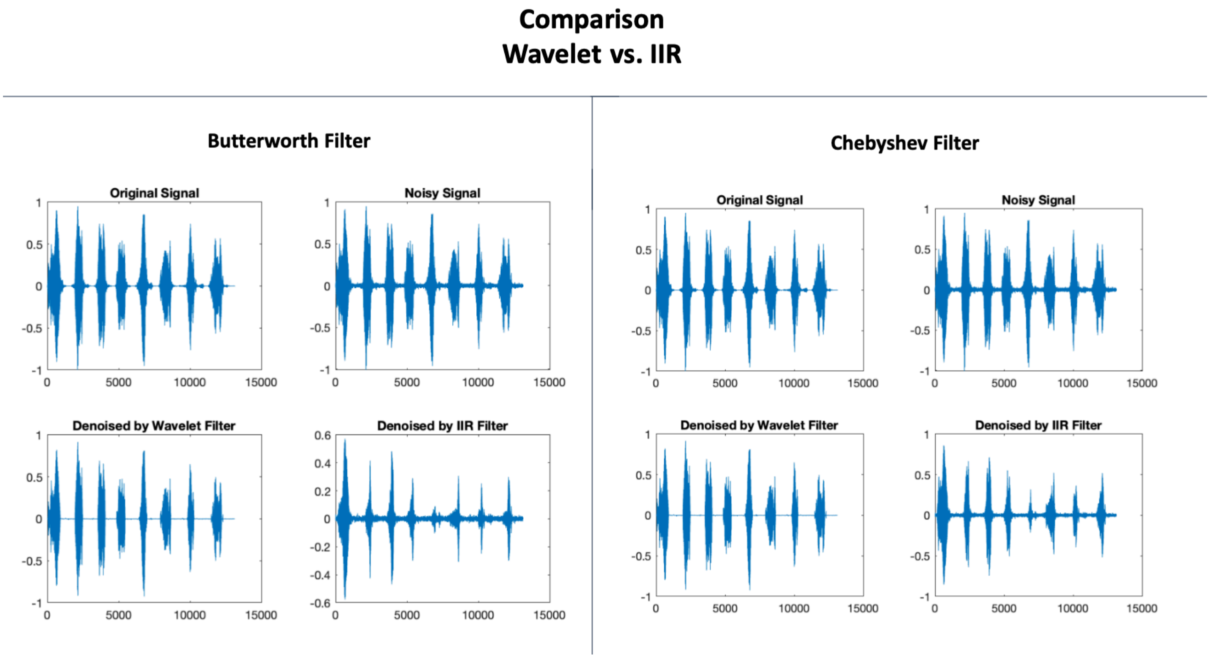


Wavelet Filter Comparison

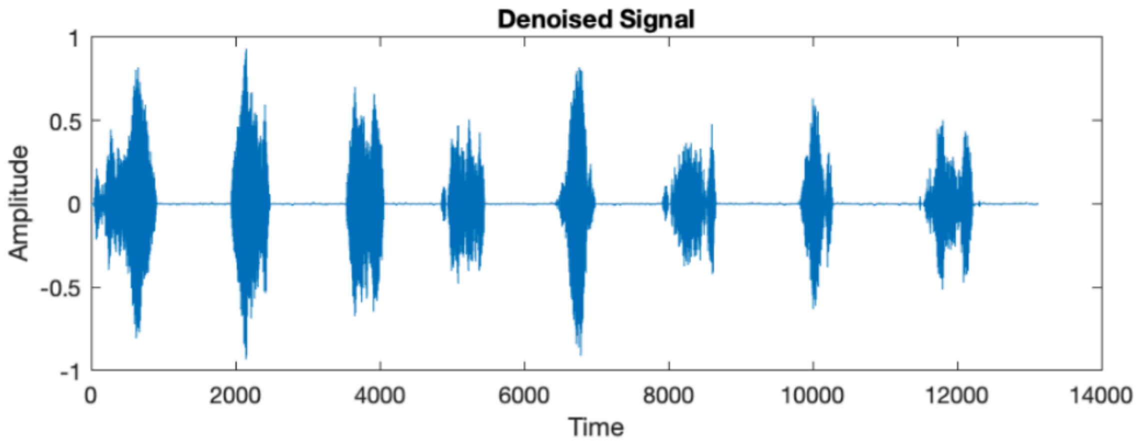
1. FIR Filter -

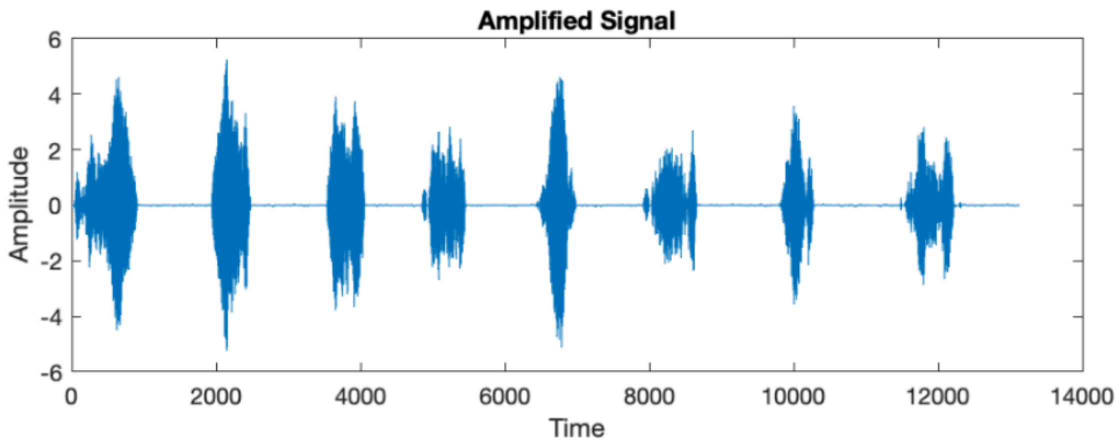


2. IIR Filter -

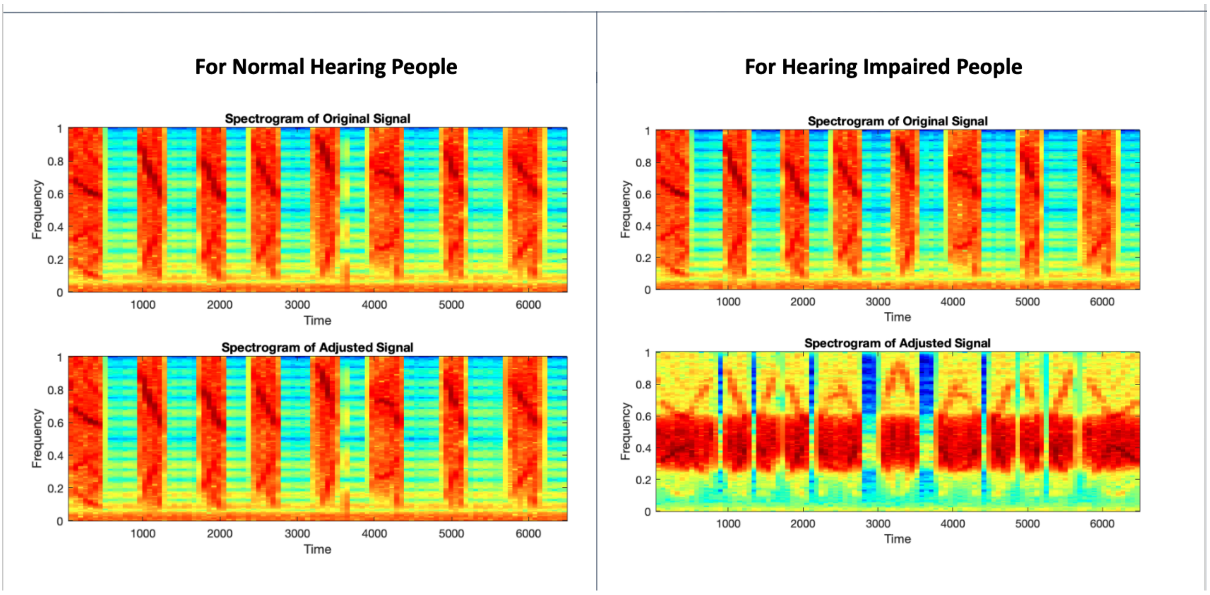


Amplification for People with Normal Hearing -





Spectrogram Analysis -



System Analysis:

Compared the result of FIR filter and Wavelet filter Or IIR filter and Wavelet - Wavelet is best.

Wavelet filters use wavelet transform technique.

Discrete Wavelet Transform

- Wavelet filter decomposes the signal with discrete wavelet transform
- Analyze the signal into different frequencies at different resolution

$$D\boxed{[a,b]} = \frac{1}{\sqrt{b}}\sum_{m=0}^{p-1} f[t_m]\psi\left[\frac{t_m - a}{b}\right]$$

a = τ
b = s

$$\boxed{a = k2^{-j} \quad b = 2^{-j}}$$

In comparison we have used the hamming and hanning window technique of FIR and Butterworth, Chebyshev window technique of IIR.

Bandpass filter is used as a Frequency shaper. It allows only the desired frequency so that depending on hearing loss we can listen to the voice clearly.

Conclusion:

- The newer digital aid is more capable of fine-tuning the sound without distorting the quality. In this digital hearing aids system implementation using MATLAB, sound processing is digitalized. Thus, it is possible to refine the sound signal, for instance by reducing noise and improving speech signals.

- In addition, by using digital technology, the amplification can be done only at the frequencies that the user needs to amplify. This will eliminate the problems with conventional amplifiers which amplified the whole signal including the noise. In general, digital hearing aid converts the incoming signals to digital signals.

- This digitalization makes its possible to precisely analyze & filter the signals. The signals can be processed in one or more frequency channels. The benefits of using digital aids can improve quality of life by improving sound quality.

Applications of the project:

- Helps Hearing Impaired People
- Directional Microphones
- Noise Suppression
- Abrupt Noise Cancellation

References: (Follow references template as given in PPT)

[1] Ritwik Dhawan and P. Mahalakshmi, "Digital filtering in hearing aid system for the hearing impaired", 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), November 2016.

[2] Satoko Yoshikawa, Yasushi Ishihara, Naoki Ohnuma, Masaaki Nagabuchi, “Fitting of Hearing Aids for Severe Hearing Loss”, Japan Journal of Logopedics and Phoniatrics, October 1998.

[3] Nariman Rahbar, “Frequency lowering”, Department of Audiology, School of Rehabilitation Sciences, Iran University of Medical Sciences, Tehran, Iran, January 2017.

[4] Robert J. Barsanti, and Jordon Gilmore, “Comparing noise removal in the wavelet and Fourier domains” in *Citadel Charleston*, SC, 2011, pp. 1-6.

[5] Joshua M. Alexander, “Frequency Compression and Transposition”, The SAGE Encyclopedia of Human Communication Sciences and Disorders, May 2019.

[6] Astrid Haastrup, M.A., “IMPROVING HIGH FREQUENCY AUDIBILITY WITH SOUND SHAPER”.

[7] Navdeep Kaur, Dr. Hardeep Singh Ryait, “Study Of Digital Hearing Aid Using Frequency Shaping Function”, International Journal of Engineering Research & Technology (IJERT), May - 2013.

[8] Alfaouri M, Daqrouq K. ECG Signal Denoising By Wavelet Tranform Thresholding. Am

J Appl Sci. 2008;5:276-281.

[9] Navdeep Kaur and Hardeep Singh Ryait, "Study of Digital Hearing Aid Using Frequency Shaping Function", International Journal of Engineering Research and Technology (IJERT), vol. 2, no. 5, May 2013.

[10] John G. Proakis and G.D. Manolakis, "Digital Signal Processing" in , New Jersey:Prentice Hall, 2010.

GROUP PICTURE: