ABSTRACT:

In this project it is desired to perform frequency analysis on the two speech recordings. Specifically, it is required to compute and display the spectrum of one segment of each of your two signals. Record ourself by saying `yes' and `no' and create a wav files. The recordings should be at 8000 samples per second.Using MATLAB. Extract a 50 millisecond segment of voiced speech from your `yes'signal. You should select the segment during the `e' sound of `yes'.The segment should be roughly periodic.Compute and display the spectrum (DTFT) of your 50 millisecond speech segment.

Repeat same for your `no' signal.Based on the spectra that you compute what is the pitch frequency of your speech? (The pitch frequency is the fundamental frequency of the quasi-periodic voiced speech signal.). Plot the pole-zero diagram of your filter in Matlab. Verify that the poles and zero match where they were designed to be. Plot the frequency response magnitude of your filter |[Hf]| versus physical frequency. Plot the impulse response h[n]of your filter. You can create an impulse signal and then use the Matlab command filter to apply your filter to the impulse signal.

Apply your filter to your speech signal Extract a short segment of your speech signal before and after filtering. Plot both the original speech waveform x[n]and filtered speech waveform y[n]. Also plot the difference between these two waveforms d[n] = y(n) -x[n].

Objectives

(a) Generate and display plot of speech signals in time domain.

(b)Design a notch filter to eliminate the pre-dominant frequency components.

(c)Compute and display the spectrum of these signals

.

CHAPTER 1:

INTRODUCTION

Basic Theory: Speech is an acoustic signal produced from a speech production system. From our understanding of signals and systems, the system characteristics depend on the design of the system. For the case of linear time invariant system, this is completely characterized in terms of its impulse response. However, the nature of response depends on the type of input excitation to the system. For instance, we have impulse response, step response, sinusoidal response and so on for a given system. Each of these output responses are used to understand the behavior of the system under different conditions. A similar phenomenon happens in the production of speech also. Based on the input excitation phenomenon, the speech production can be broadly categorized into three activities. The first case where the input excitation is nearly periodic in nature, the second case where the input excitation is random noise-like in nature and third case where there is no excitation to the system.

NOTCH FILTER:

• A notch filter is used to eliminate a specific frequency from a given signal

TYPES OF NOTCH FILTERS:

FILTERS are mainly of two types in notches

1) IIR notch filter (INFINITE IMPULSE RESPONSE NOTCH FILTER)

2) FIR notch filter (FINITE IMPULSE RESPONSE NOTCH FILTER)

In this project we are using both IIR notch filters

IIR NOTCH FILTER:

• It is a filter that depends upon the present and past outputs and present input

• The general equation of an IIR filter can be expressed as follows:

$$y[n]=\sum_{k=0}^{N}b[k]x[n-k]+\sum_{k=1}^{M}a[k]y[n-k]$$

ak and bk are the filter coefficients.

There are 8 main tasks in our project

Task1: Record yourself saying `yes' and `no' and create a wav files. The recordings should be at 8000 samples per second.Using MATLAB, you can form one vector of 4000 samples (half second)for the `yes'. You should form a second vector also of 4000 samples of the`no'. Plot the two speech signals

Task2: Extract a 50 millisecond segment of voiced speech from your `yes'signal. You should select the segment during the `e' sound of `yes'.The segment should be roughly periodic.Compute and display the spectrum (DTFT) of your 50 millisecondspeech segment. The frequency units should be in Hertz and should be computed according the sampling rate (8000 samples/second).Plot the spectrum on a linear scale and on a log scale. (For the logscale, use 20log10(|Xf|)).Based on your spectrum, what is the fundamental frequency present?Can you recognize the harmonics in the spectrum?

Task3: It can be observed that the spectrum of a short segment of speech signal contains roughly equally-spaced peaks. Now it is desired to eliminate one of the prominent peaks.Select the frequency (in cycles/second) of the peak to be eliminated from the spectrum computed.

Design a second order digital notch filter using the following transfer function

$$H[z] = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

that has zeros at the selected frequencies and corresponding poles. The zeros should be 2jfneπ± and the poles should be where ‘r’ is slightly less than 1, and fn is normalized frequency (cycles/sample);this should be a number between zero and one half.Verify different values of ‘r’.

Task4: Repeat Task2 for your `no' signal.Based on the spectra that you compute, what is the pitch frequency of your speech? (The pitch frequency is the fundamental frequency of the quasi-periodic voiced speech signal.)

Task5: Plot the pole-zero diagram of your filter (use the Matlab command zplane) in Matlab. Verify that the poles and zero match where they were designed to be.

Task6: Plot the frequency response magnitude of your filter |[Hf]| versus physical frequency (the frequency axis should go from zero to half thesampling frequency). You can use the Matlab command freqz tocompute the frequency response. Verify that the frequency responsehas a null at the intended frequency.

Task7: Plot the impulse response h[n]of your filter. You can create an impulse signal and then use the Matlab command filter to apply yourfilter to the impulse signal.

Task8: Apply your filter to your speech signal (use Matlab command filter). Extracta short segment of your speech signal before and after filtering. Plot both the original speech waveform x[n]and filtered speech waveform y[n](you might try to plot them on the same axis). Also plot the difference between these two waveforms d[n] = y(n) -x[n]

What do you expect the signal d[n]to look like?

For the short segment you extract from the original and filtered speech waveforms, compute and plot the spectrum. Were you able to eliminate the spectral peak that you intended to?

CHAPTER 2:

Task 1: Recording 'yes' and 'No' speech signal

Matlab Code:

```
% For Recording 'yes' speech signal
r = audiorecorder(8000,16,1);
disp('Start speaking.')
recordblocking(r,1);
disp('End of Recording.');
play(r); %yes
yes = getaudiodata(r);
 % For Recording 'no' speech signal
g = audiorecorder(8000,16,1);
disp('Start speaking.')
recordblocking(g,1);
disp('End of Recording.');
play(g); %no
no = getaudiodata(g);
 yes1 = yes(1:4000);
yes2 = yes(4001:8000);
 no1 = no(1:4000);
no2 = no(4001:8000);
 % Plot Of 'yes' Speech Signal In Frequency Domain
figure(1); plot(yes); title('Yes Speech Signal (8000Samples/sec)'); xlabel('Samples-----
>');ylabel('Magnitude');
 % Plot Of 'yes1' Speech Signal In Frequency Domain
figure(2); plot(yes1); title('One Vector Of 4000 Samples (half second) Of Yes');
xlabel('Samples----->');ylabel('Magnitude');
 % Plot Of 'yes2' Speech Signal In Frequency Domain
```

```
figure(3); plot(yes2); title('Second Vector Of 4000 Samples (half second) Of Yes');
xlabel('Samples----->');ylabel('Magnitude');
%Plot Of ‘yes’ Speech Signal In Time Domain
n = size(yes);
Fs = 8000;
ts = 1/Fs;
tmax = (n-1)*ts;
t = 0:ts:tmax;
figure(4); plot(t,yes); title('Yes Speech Signal (8000Samples/sec)'); xlabel('Time In
Seconds----->');ylabel('Magnitude');
 %Plot Of ‘yes1’ Speech Signal In Time Domain
ts1 = 1/Fs;
tmax1 = ((n/2)-1)*ts1;
t1 = 0:ts1:tmax1;
figure(5); plot(t1,yes1);title('One Vector Of 4000 Samples (half second) Of Yes');
xlabel('Time In Seconds----->');ylabel('Magnitude');
 %Plot Of ‘yes2’ Speech Signal In Time Domain
ts2 = 1/Fs;
tmax2 = ((n/2)-1)*ts2;
t2 = 0:ts2:tmax2;
figure(6); plot(t2,yes2);title('Second Vector Of 4000 Samples (half second) Of Yes');
xlabel('Time In Seconds----->');ylabel('Magnitude');
 % Plot Of ‘no’ Speech Signal In Frequency Domain
figure(7); plot(no); title('No Speech Signal (8000Samples/sec)'); xlabel('Samples-----
>');ylabel('Magnitude');
 % Plot Of ‘no1’ Speech Signal In Frequency Domain
figure(8); plot(no1); title('One Vector Of 4000 Samples (half second) Of No');
xlabel('Samples----->');ylabel('Magnitude');
 % Plot Of ‘no2’ Speech Signal In Frequency Domain
figure(9); plot(no2); title('Second Vector Of 4000 Samples (half second) Of No');
xlabel('Samples----->');ylabel('Magnitude');
 %Plot Of ‘no’ Speech Signal In Time Domain
```

```
n = size(no);
Fs = 8000;
ts = 1/Fs;
tmax = (n-1)*ts;
t = 0:ts:tmax;
figure(10);plot(t,no); title('No Speech Signal (8000Samples/sec)'); xlabel('Time In Seconds----->');ylabel('Magnitude');
 %Plot Of ‘no1’ Speech Signal In Time Domain
ts1 = 1/Fs;
tmax1 = ((n/2)-1)*ts1;
t1 = 0:ts1:tmax1;
figure(11); plot(t1,no1);title('One Vector Of 4000 Samples (half second) Of No');
xlabel('Time In Seconds----->');ylabel('Magnitude');
 %Plot Of ‘no2’ Speech Signal In Time Domain
ts2 = 1/Fs;
tmax2 = ((n/2)-1)*ts2;
t2 = 0:ts2:tmax2;
figure(12); plot(t2,no2);title('Second Vector Of 4000 Samples (half second) Of No');
xlabel('Time In Seconds----->');ylabel('Magnitude');

% all data is stored as task1.mat and task4.mat
```

Result:

Start speaking.

End of Recording.

Start speaking.

End of Recording

Figure1:

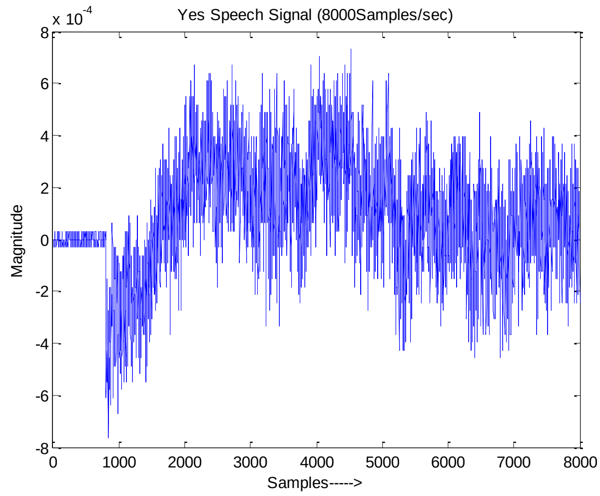


Figure2:

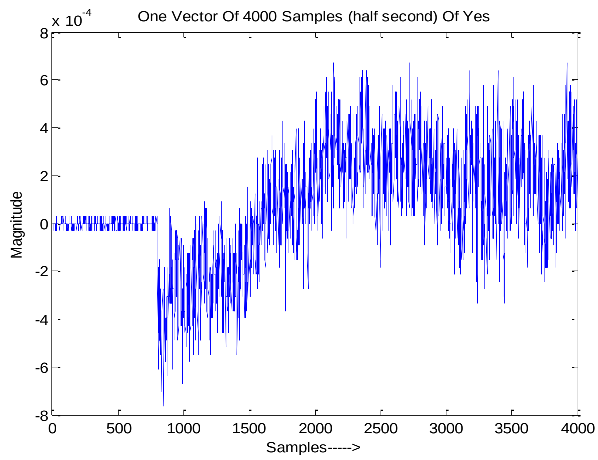


Figure3:

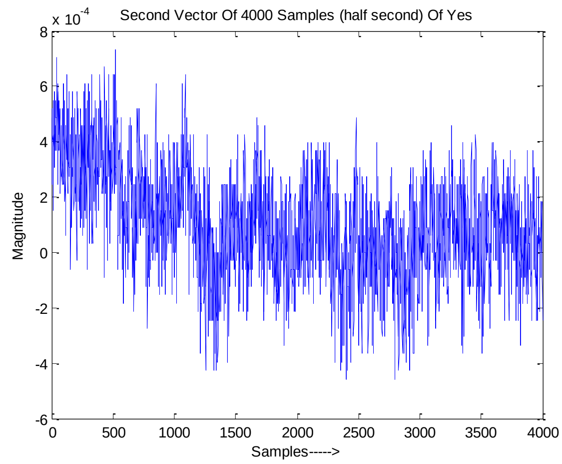


Figure4:

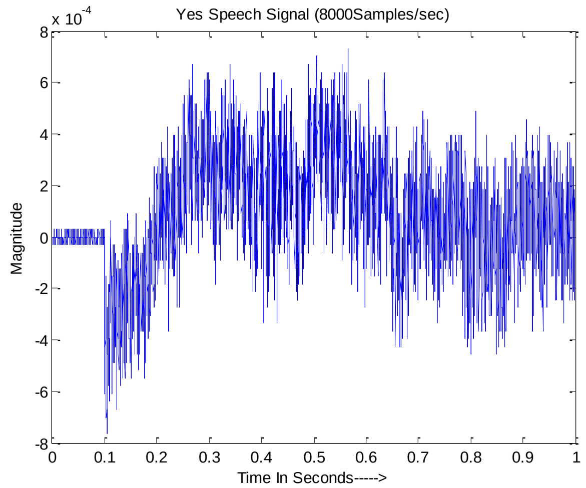


Figure5:

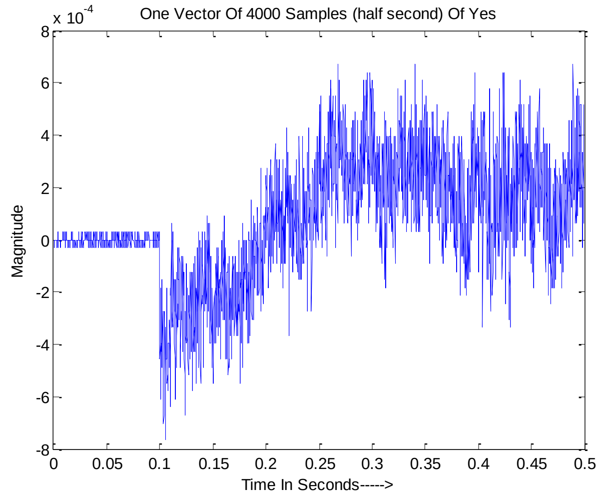


Figure6:

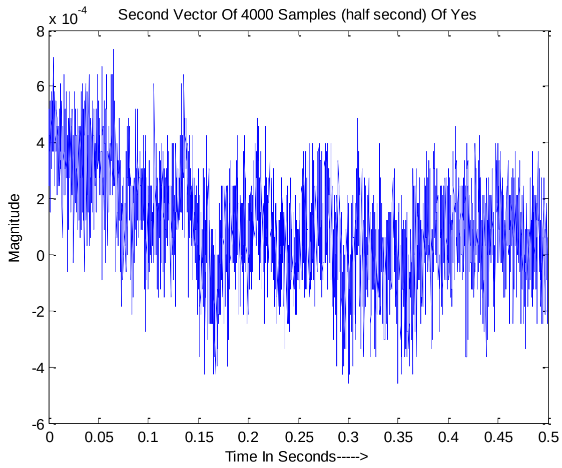


Figure7:

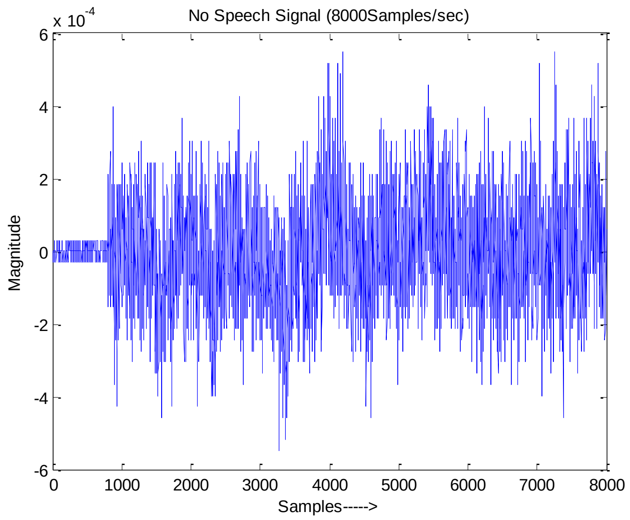


Figure8:

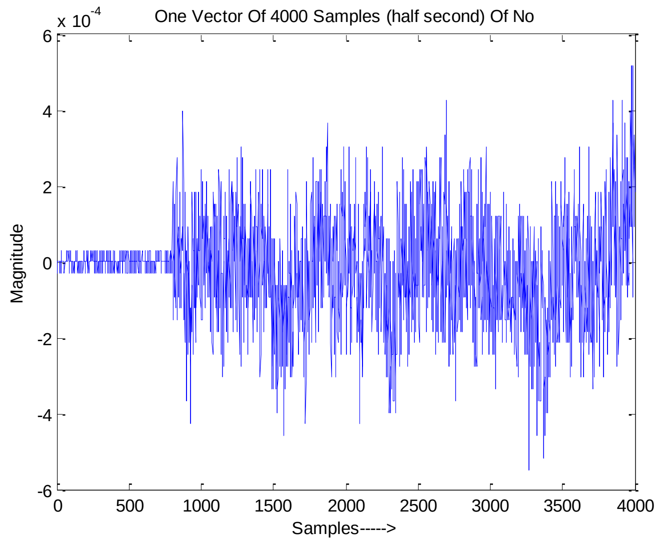


Figure9:

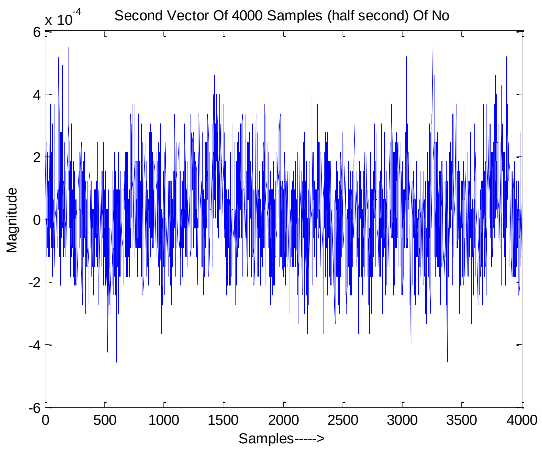


Figure10:

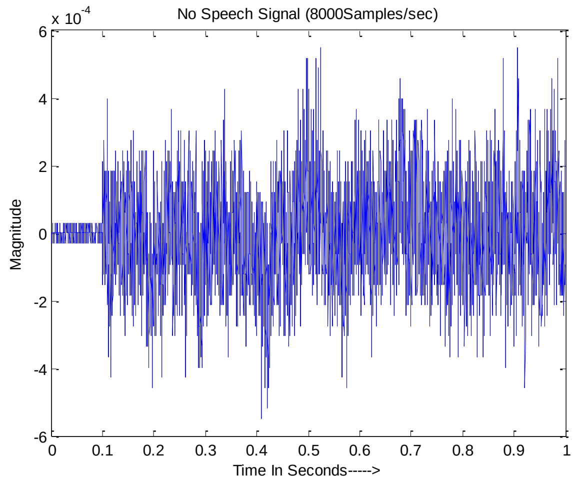


Figure11:

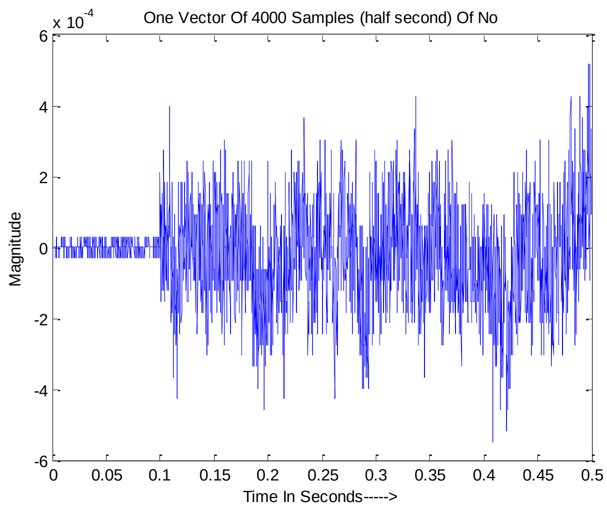
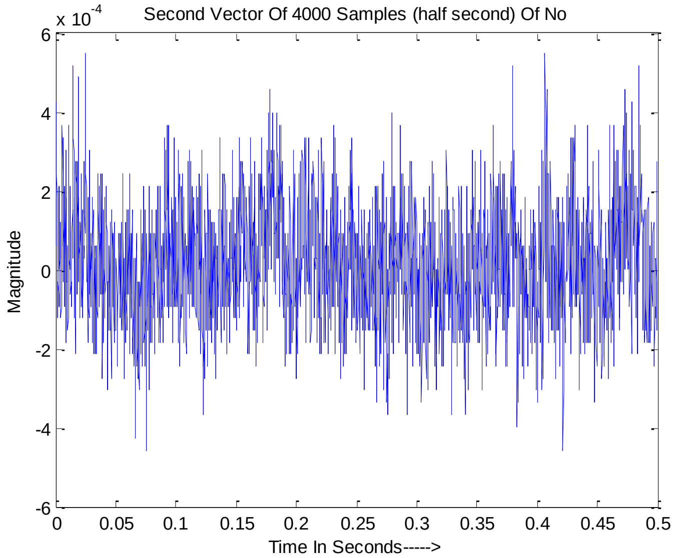


Figure12:



Task2:

```
load task1.mat;
sound(yes);
sound(yes1);
sound(yes2);
 %While Listening To The Above Sounds, It Is Observed That the 'e' segment of 'yes'
speech signal is available in the vector "yes1"
% Extracting 50ms 'e' segment from 'yes' speech signal from "yes1"
 max_value = max(abs(yes1));
yes1 = yes1/max_value;
yes1 = yes1(Fs*0.35:Fs*0.40);
 %Fs = 8000
% Fs*0.35 = 2800 , Fs*0.40 = 3200
% 0.40 – 0.35 = 0.05 = 50ms
% (400 samples of yes1 from 2800 to 3200)
 sound(yes1);
ext = 0.35:1/Fs:0.40; % 0.40 – 0.35 = 0.05 = 50ms
plot(ext,yes1); title('50ms Segment of 'e' in yes Which is roughly Periodic');
xlabel('Time In Seconds --------->'); ylabel('Magnitude----->');
dtftyes1 = abs(fft(yes1));
```

```
%figure();plot(dtftyes1);
%title('Linear Magnitude Spectrum Of 'e' Segment'); %xlabel('Frequency In Hz');
ylabel('Magnitude');
dtftyes1 = dtftyes1(1:ceil((length(dtftyes1)/2)));
tt = linspace(1/Fs,Fs/2,length(dtftyes1));
figure(2);plot(tt,dtftyes1);
title('Linear Magnitude Spectrum Of 'e' Segment'); grid on; xlabel('Frequency In Hz');
ylabel('Magnitude');
dtftyes1log=20*log10(dtftyes1);
figure(3);plot(tt,dtftyes1log);
title('Log Magnitude Spectrum Of 'e' Segmnet (dB Scale)'); xlabel('Frequency In Hz');
ylabel('Magnitude'); grid on;
 %Fundamental Frequency 780Hz
% Harmonics Recognized
```

Result:

Figure1:

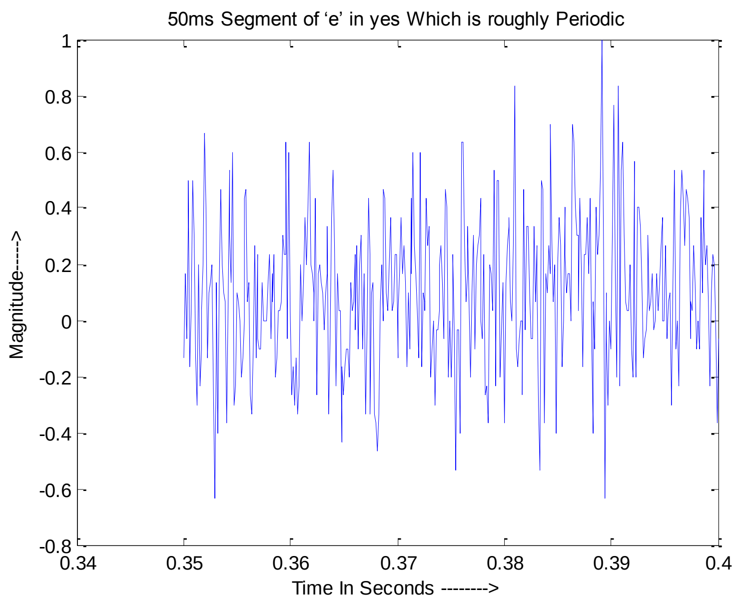


Figure2:

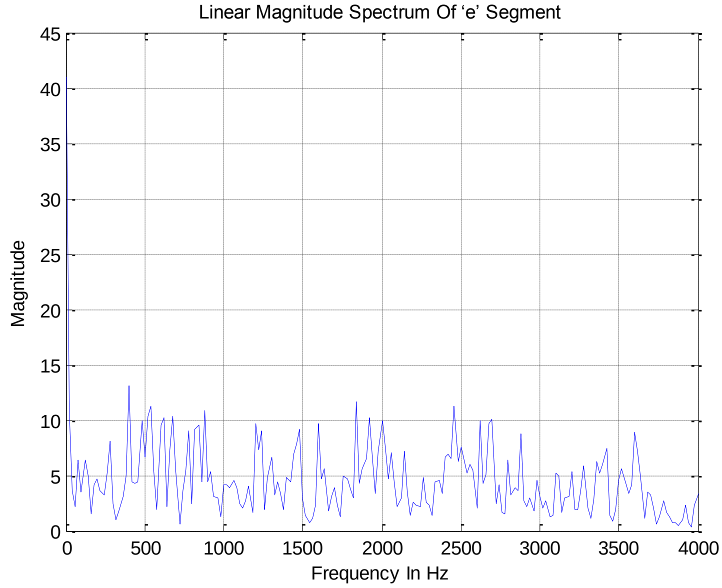
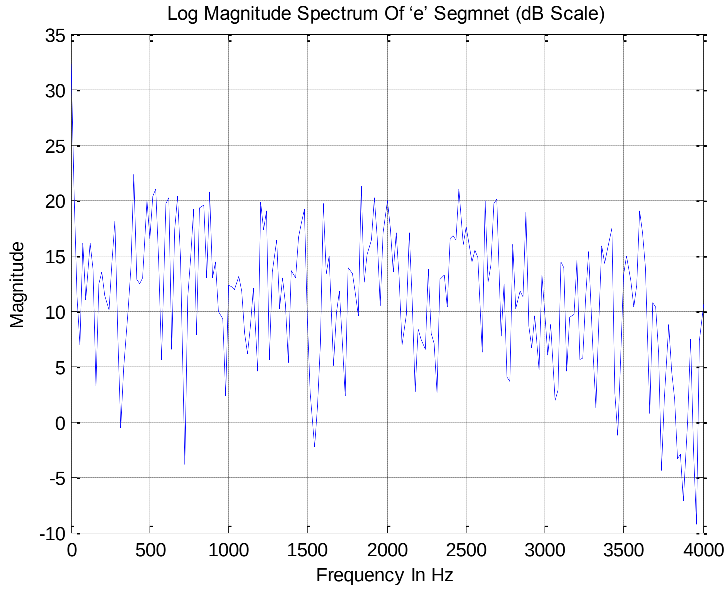


Figure3:



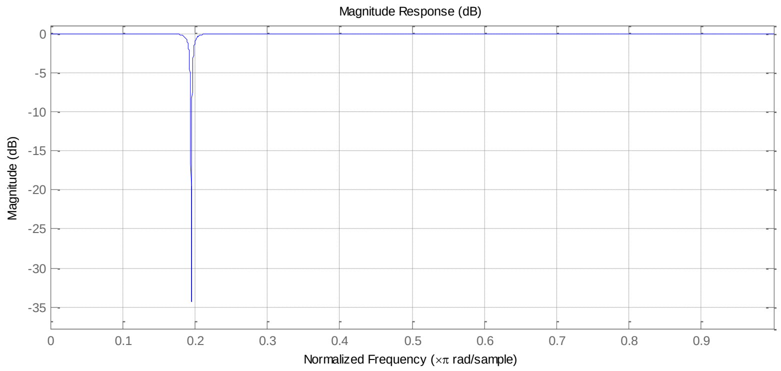
Task3:

```
%Making Use Of In-built Second Order IIR NOTCH FILTER
% Observed Fudamental Frquency = 780Hz

 Fs = 8000;
w = 780/(Fs/2);
b = w/35; % Q factor = 35
```

```
[num,den] = iirnotch(w,b);
fvtool(num,den);
```

Result:



Task4:

```
load task4.mat;
 sound(no);
sound(no1);
sound(no2);
 %While Listening To The Above Sounds, It Is Observed That the 'o' segment of 'no'
speech signal is available in the vector "no2"
% Extracting 50ms 'o' segment from 'no' speech signal from "no2"
 Fs = 8000;
max_value = max(abs(no2));
no2 = no2/max_value;
no2 = no2(Fs*0.25:Fs*0.30);
 %Fs = 8000
% Fs*0.25 = 2000 , Fs*0.30 = 2400
% 0.30 – 0.25 = 0.05 = 50ms
% (400 samples of yes1 from 2000 to 2400)
 sound(no2);
ext = 0.25:1/Fs:0.30; % 0.30 – 0.25 = 0.05 = 50ms
figure(1); plot(ext,no2); title('50ms Segment of 'o' in no Which is roughly Periodic');
xlabel('Time In Seconds -------->'); ylabel('Magnitude----->');
```

```
dtftno1 = abs(fft(no2));
%figure(2); plot(dtftno1);
%title('Linear Magnitude Spectrum Of 'o' Segment'); %xlabel('Frequency In Hz');
ylabel('Magnitude');
dtftno1 = dtftno1(1:ceil((length(dtftno1)/2)));
tt = linspace(1/Fs,Fs/2,length(dtftno1));
figure(2);plot(tt,dtftno1);
title('Linear Magnitude Spectrum Of 'o' Segment'); grid on; xlabel('Frequency In Hz');
ylabel('Magnitude');
dtftno1log=20*log10(dtftno1);
figure(3);plot(tt,dtftno1log);
title('Log Magnitude Spectrum Of 'o' Segmnet (dB Scale)'); xlabel('Frequency In Hz');
ylabel('Magnitude'); grid on;
 %Fundamental Frequency 780Hz
% Harmonics Recognized
```

Result:

Figure1:

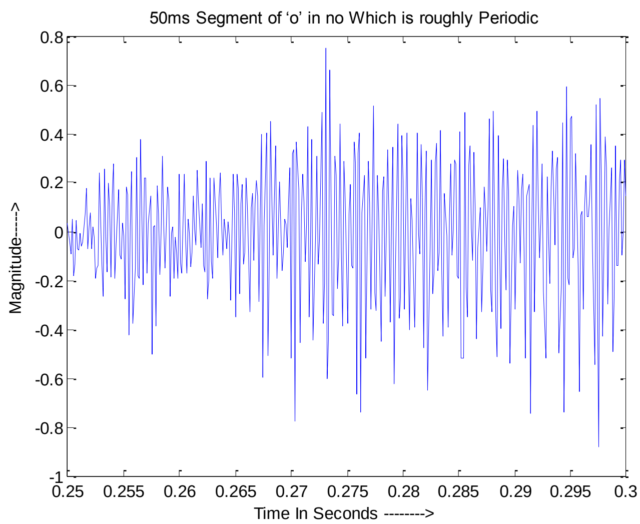


Figure2:

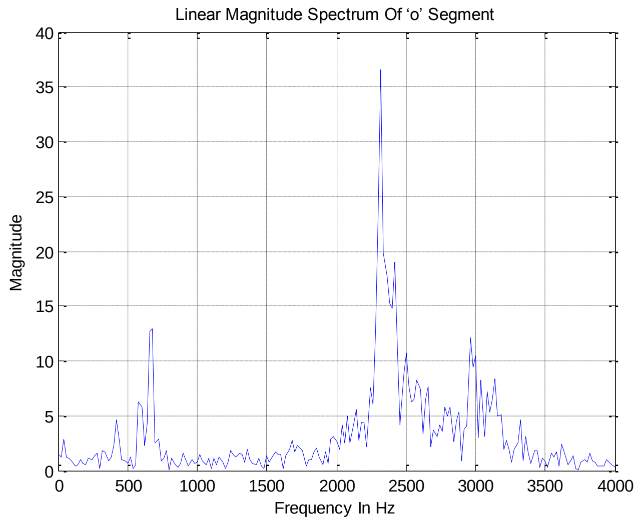
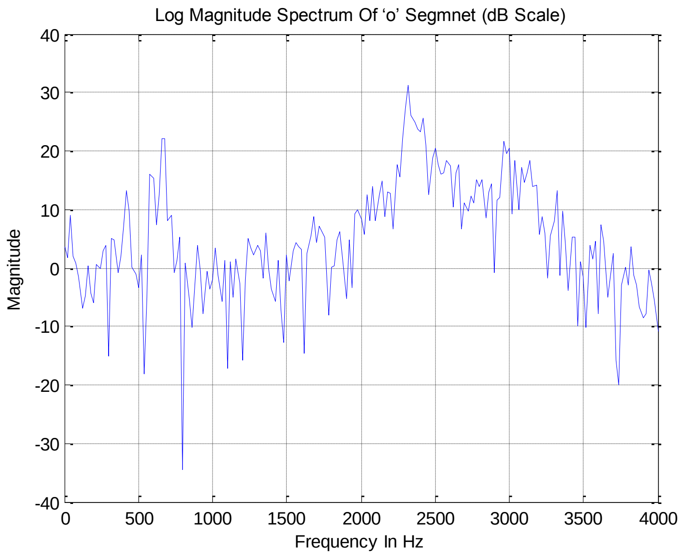


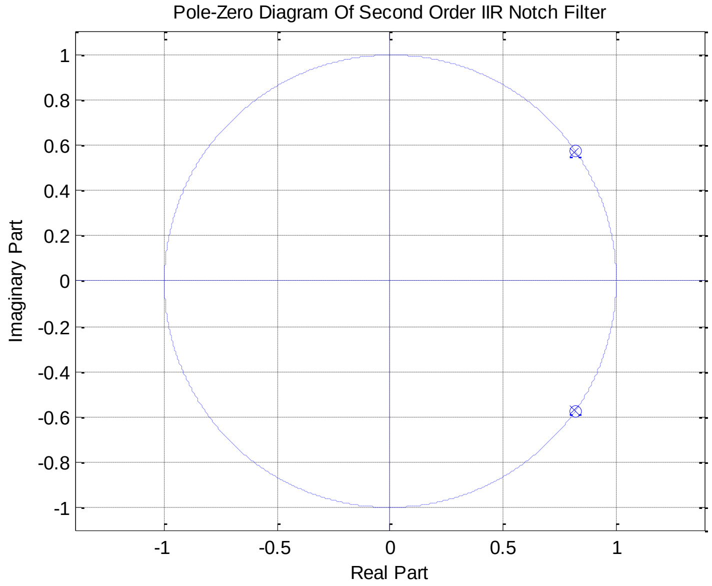
Figure3:



Task5:

```
load task3.mat;
%Pole-Zero Diagram Of Notch Filter
zeroes = roots(num);
poles = roots(den);
zplane(num,den);
title('Pole-Zero Diagram Of Second Order IIR Notch Filter'); grid on;
```

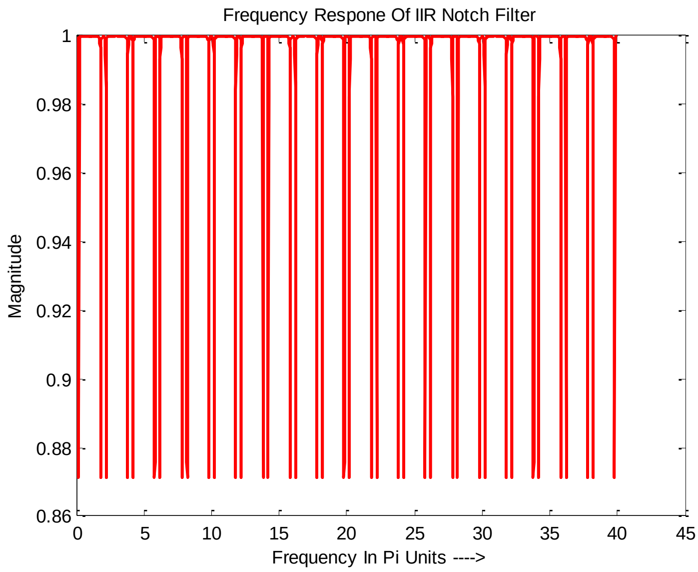
Result:



Task6:

```
%Task 6:
 load task6.mat;
 k = 0:4000;
w = (pi/100)*k;
X = freqz(num,den,w);
magX = abs(X);
plot(w/pi,magX,'r','LineWidth',1.5);
title('Frequency Respone Of IIR Notch Filter');
xlabel('Frequency In Pi Units ---->'); ylabel('Magnitude');
[xs,fs]=audioread('silence.wma');
whos;
subplot(4,1,1);plot(xs(1:2205));
subplot(4,1,2);plot(xs(2206:2206+2205));
subplot(4,1,3);plot(xs(2206+2206:2206+2206+2205));
subplot(4,1,4);plot(xs);
```

Result:



Task7:

```
load task7.mat;
[h,t] = impz(num,den,100);
figure(1); plot(t,h,'b'); title('Impulse Response Of IIR Notch Filter');
figure(2); stem(t,h,'r','fill','LineWidth',1.5);
title('Impulse Response Of IIR Notch Filter');
grid on;
impsig = filter(num,den,h);
%plot(t,impsig); title('Impulse Signal After Filtering');
%xlabel('Time In Seconds'); ylabel('Magnitude');
```

Result:

Figure1:

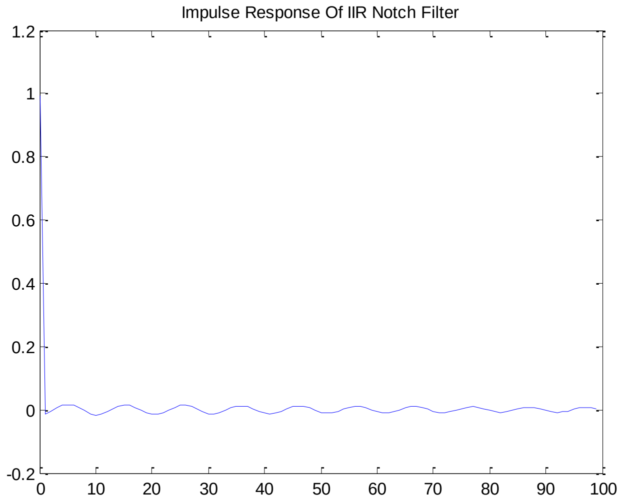
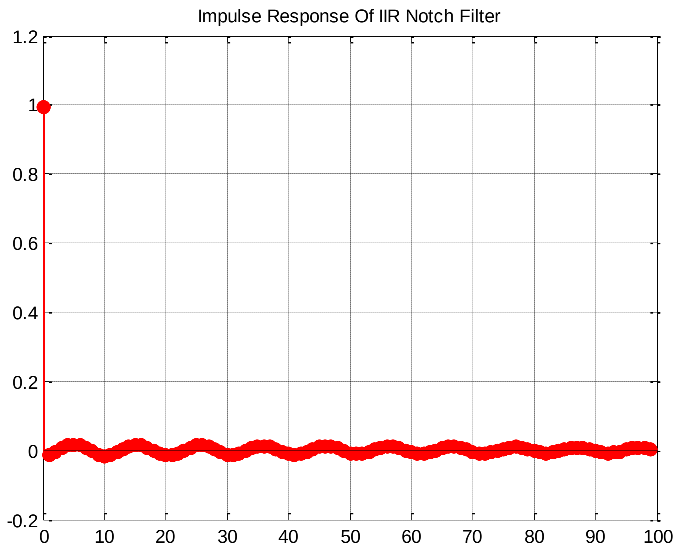


Figure2:



Task8:

```
%Task 8:

load task3.mat;

load task1.mat;
```

```
%applying the notch filter to the speech signal
yesnew = filter(num,den,yes); %yes speech signal
nonew = filter(num,den,no);  %no  speech signal
Fs
yesshort = yes(Fs*0.35:Fs*0.40);
yesnewshort = yesnew(Fs*0.35:Fs*0.40);
tshort = 0.35:1/Fs:0.40;
figure(1);plot(tshort,yesshort);title('50ms Short Segment Before Filtering');
xlabel('Time In Seconds');ylabel('Magnitude');
figure(2);plot(tshort,yesnewshort); title('50ms Short Segment After Filtering');
xlabel('Time In Seconds');ylabel('Magnitude');
 difference = abs(yesnewshort-yesshort);
figure(3);plot(tshort,difference);title('Difference Between The Two Waveforms');
xlabel('Time In Seconds');ylabel('Magnitude');
```

Result:

Figure1:

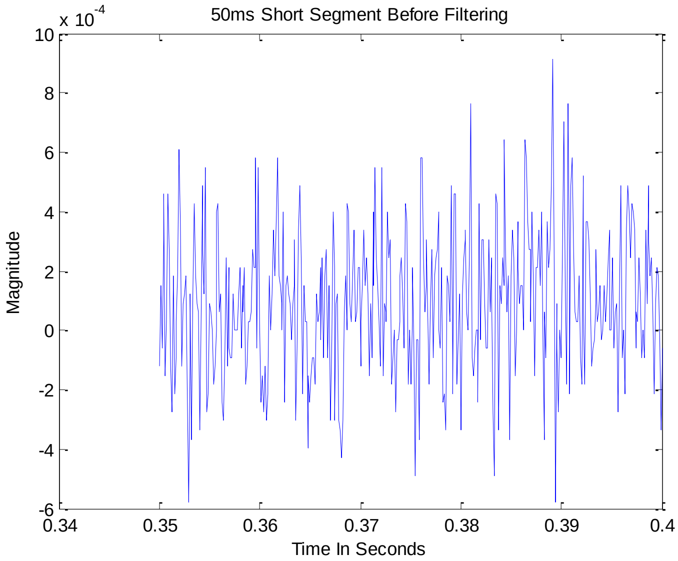


Figure2:

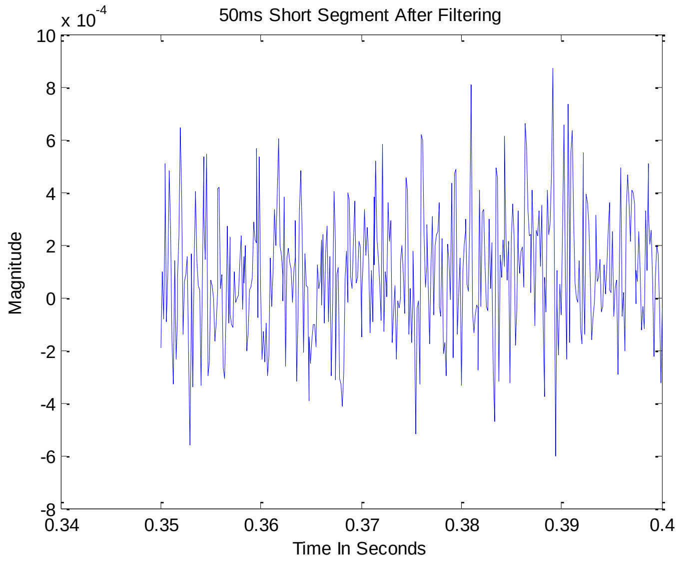
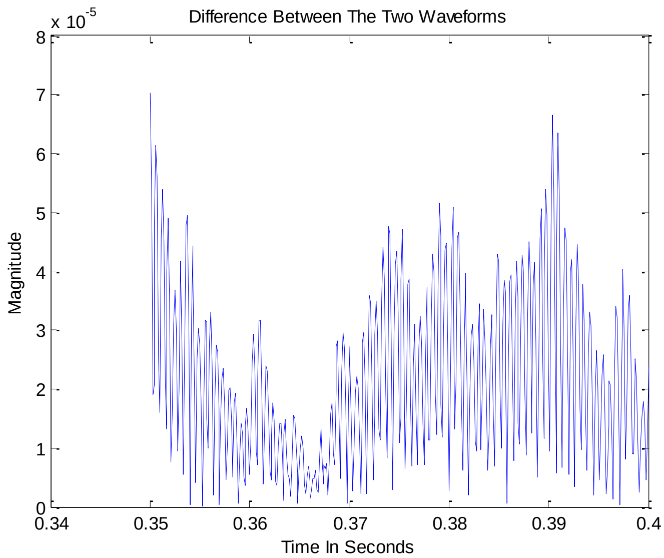


Figure3:



CHAPTER 3:

References:

1) Agarwal, A., Jain, A. and Prakash, N., 2010. Word Boundary Detection in Continuous Speech based on Suprasegmental Features for Hindi Language. 2nd International Conference on Signal Processing Systems, vol. 2, pp. V2-591-V2-594.

2) Anusuya, M. A. and Katti, S. K., 2009. Speech Recognition by machine: A review. International journal of Computer science and information security, vol. 6, no. 3, pp. 181-205.