EE-414 Speech Processing Lab

Lab-4 - 180020002

Aim:

- To understand the time and frequency domain characteristics of voiced and unvoiced speech.
- To perform the voiced/unvoiced/silence classification of speech.

Theory:

Voiced speech:

If the input excitation is nearly periodic impulse sequence, then the corresponding speech looks visually nearly periodic and is termed as **voiced speech**. During the production of voiced speech, the air exhaling out of lungs through the trachea is interrupted periodically by the vibrating vocal folds. Due to this, the glottal wave is generated that excites the speech production system resulting in the voiced speech. When we look at the speech signal waveform, if it looks nearly periodic in nature, then it can be marked as voiced speech.

Unvoiced speech:

If the excitation is random noise-like, then the resulting speech will also be random noise-like without any periodic nature and is termed as **Unvoiced Speech**. During the production of unvoiced speech, the air exhaling out of lungs through the trachea is not interrupted by the vibrating vocal folds. However, starting from glottis, somewhere along the length of vocal tract, total or partial closure occurs which results in obstructing air flow completely or narrowly. This modification of airflow results in stop or frication excitation and excites the vocal tract system to produce unvoiced speech. The unvoiced speech will not have any periodic nature. This will be the main distinction between voiced and unvoiced speech.

Silence Region:

The speech production process involves generating voiced and unvoiced speech in succession, separated by what is called **silence region**. During silence region, there is no excitation supplied to the vocal tract and hence no speech output. However, silence is an integral part of speech signal. Without the presence of silence region between voiced and unvoiced speech, the speech will not intelligible. Further, the duration of silence along with other voiced or unvoiced speech is also an indicator of certain category of sounds.

Procedure:

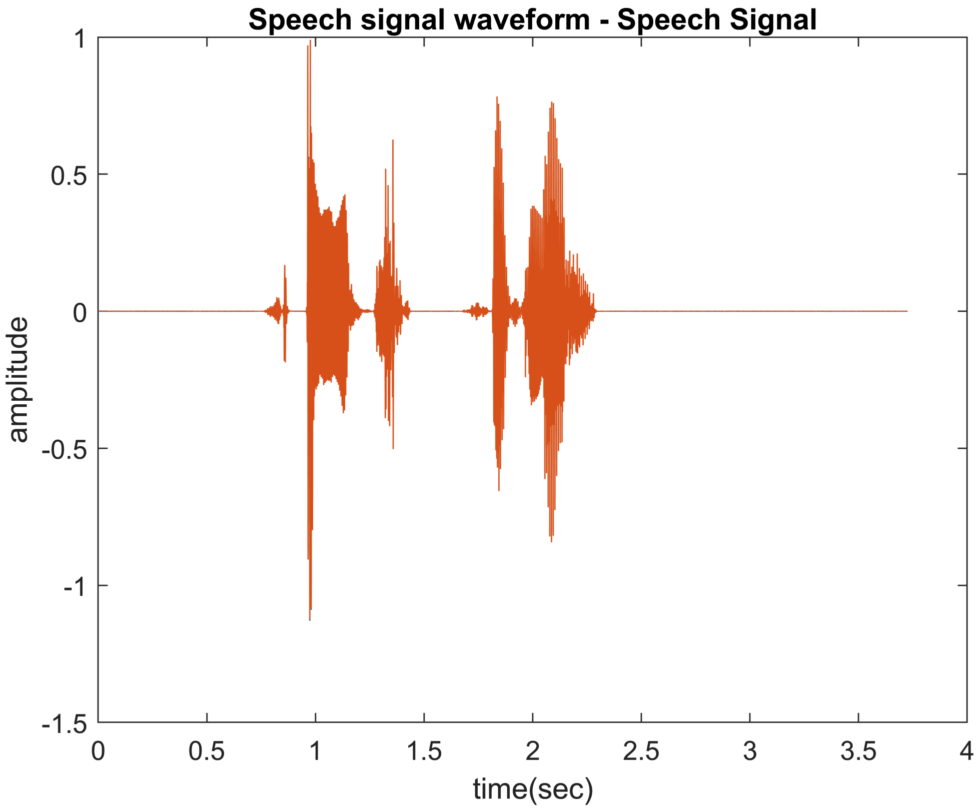
Problem statement-A

A. Record the phrase “Speech signal” and plot the time waveform. Use 16kHz and 16 bits/sample as the sampling frequency and bit resolution respectively.

we can use audacity software to record the phrase and convert into sampling frequency of 16kHz and bit resolution as 16bits/sample. Then here we plot the waveforms.

```
%Matlab program to load and plot waveform of speech signal stored in
% wav file format
%file name is speech_signal.wav and full path is given
```

```
[y,fs]=audioread('speech_signal.wav');

%normalising the signal amplitudes to be in -1 to 1
y=y./(1.01*abs(max(y)));
%plotting waveform of the speech signal
t = 0 : 1 / fs : (length(y) - 1) / fs;
plot(t, y);
xlabel('time(sec)');
ylabel('amplitude');
title('Speech signal waveform - Speech Signal');
```



Problem statement-B

B. Examine "s", "ch", any one vowel, any one nasal from A as follows. Take one segment of 25 ms duration at the centre of the sound. Compute and plot the Autocorrelation function, and comment on the periodicity of the sounds. Compare the autocorrelation plots for various sounds and comment on how autocorrelation can be used for classifying the sounds as voiced and unvoiced.

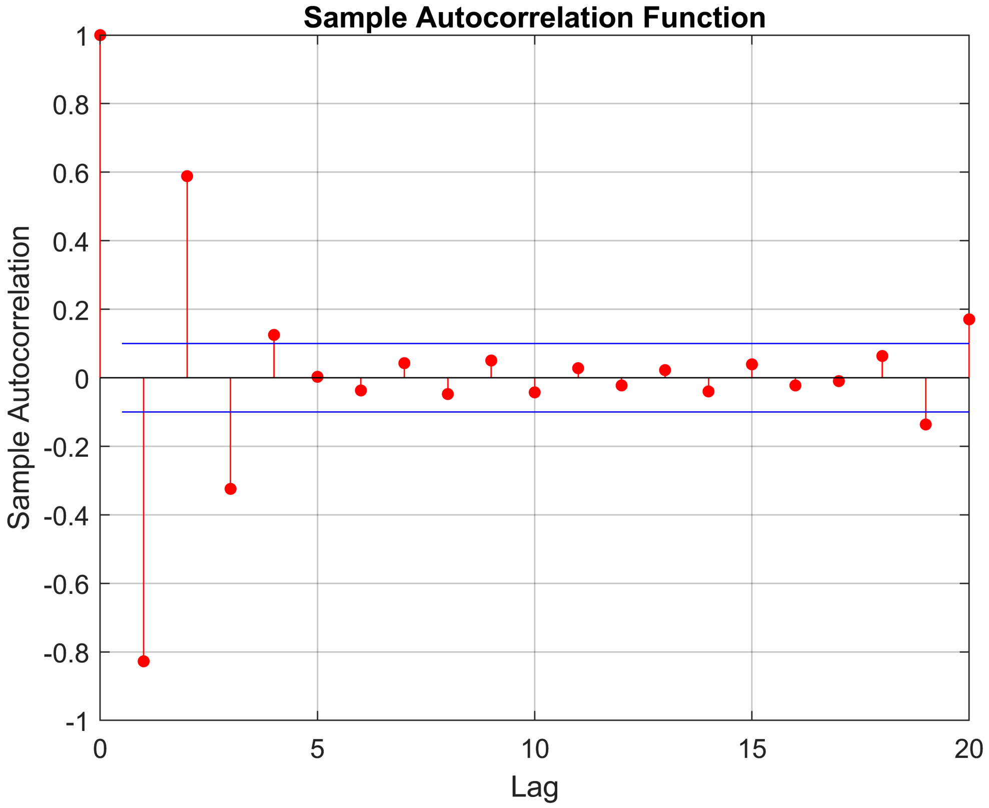
We will use the wavesurfer for analysing the plots. We can note down the 25ms duration of the segment at the centre of the sound. The time-stamps for each sound is obtained from wavesurfer.

```
% time indexes of sounds with 25ms duration at the centre
%/s/
y_s = y(ceil(0.28*fs) : floor(0.305*fs));
%/ch/
y_ch = y(ceil(0.791*fs) : floor(0.816*fs));
% vowel-/i/
y_i = y(ceil(1.2995*fs) : floor(1.3245*fs));
```

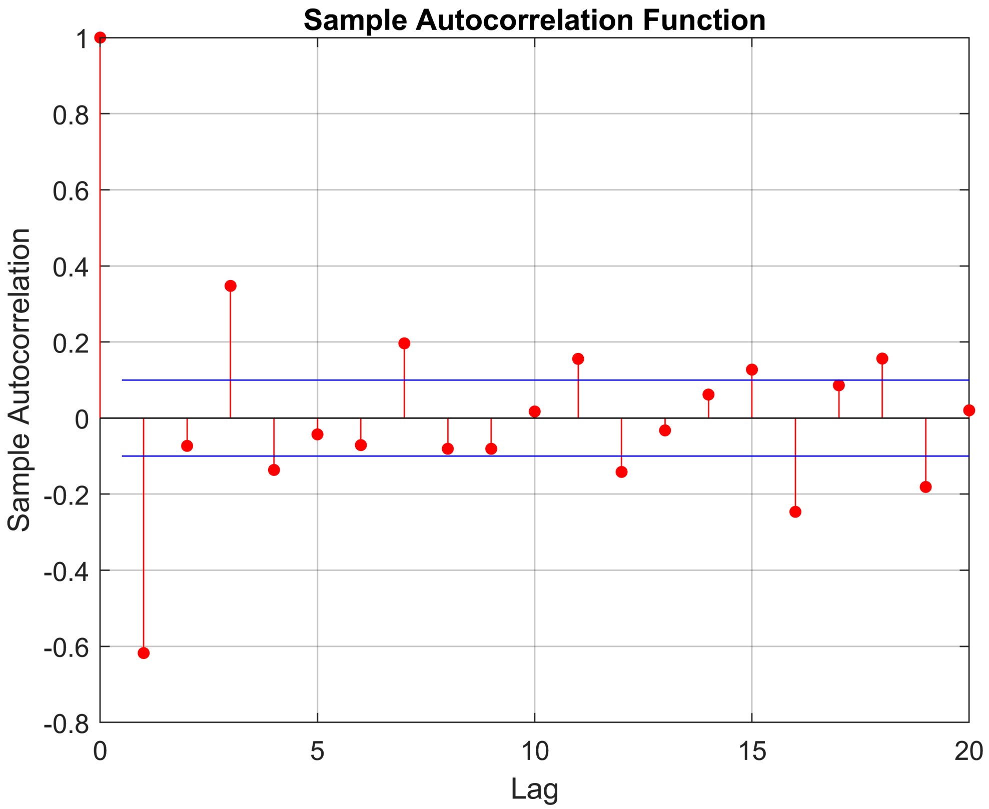
```
% nasal-/n/
y_n = y(ceil(1.4755*fs) : floor(1.5005*fs));
% silence-/p/
y_p= y(ceil(0.362*fs) : floor(0.387*fs));
```

Computing and plotting the Autocorrelation function for every speech sounds

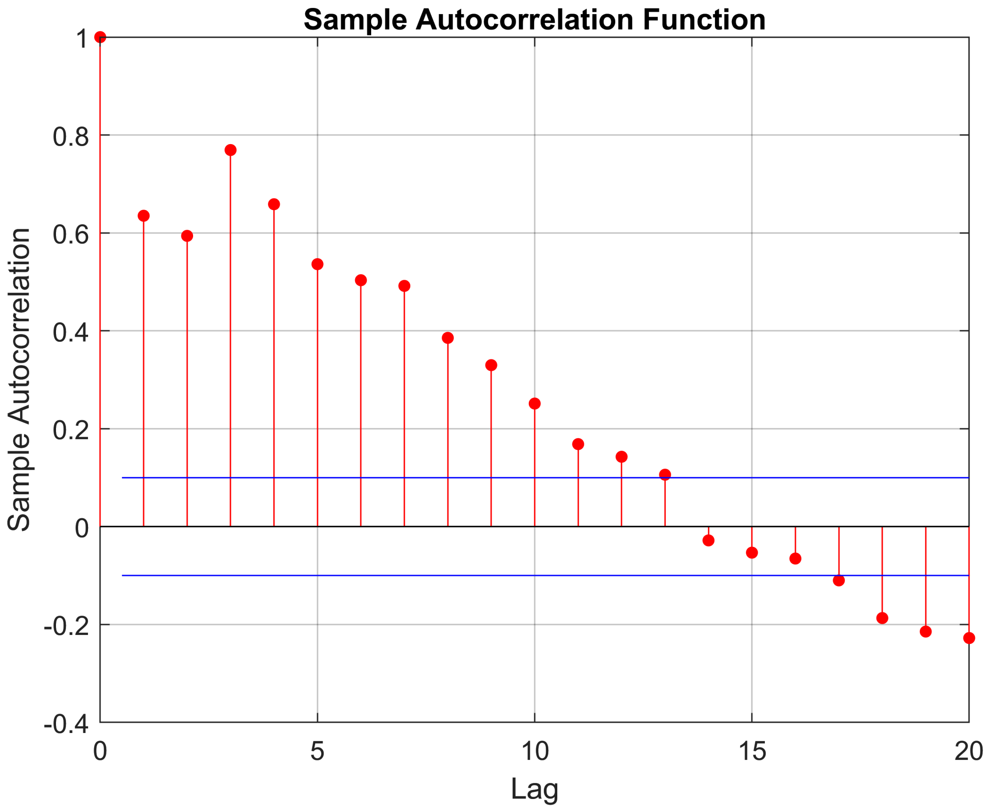
```
% /s/
autocorr(y_s);
```



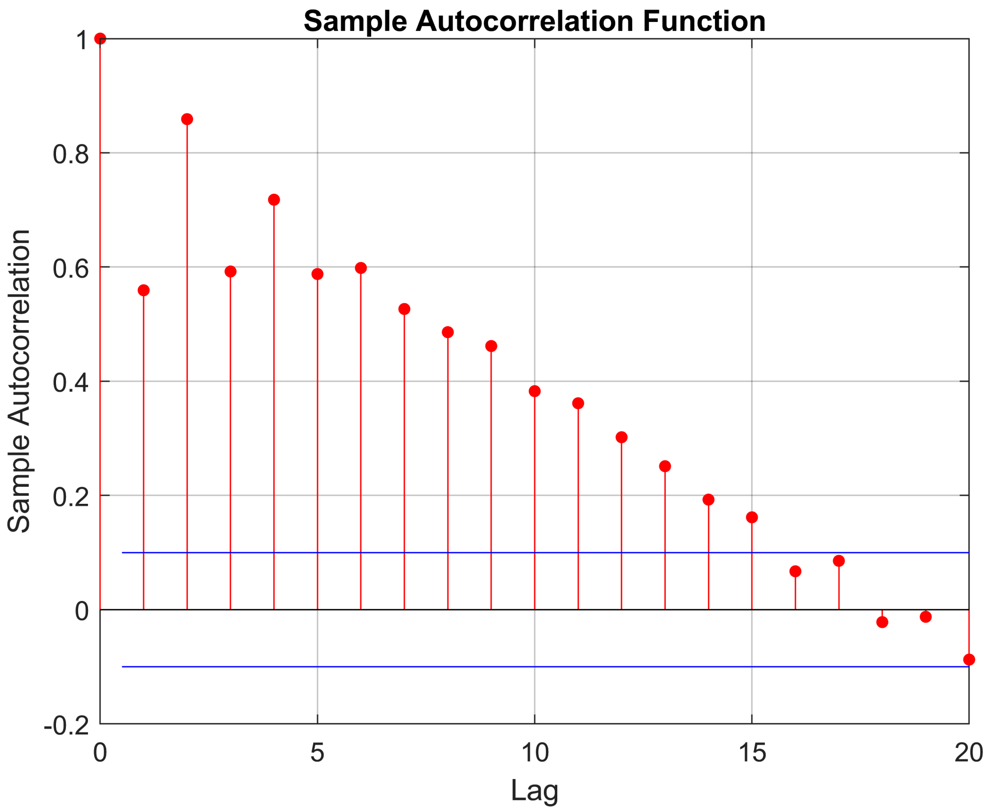
```
%/ch/
autocorr(y_ch);
```



```
%/i/
autocorr(y_i);
```



```
%/n/
autocorr(y_n);
```



Observations:

When we observe the plots for /s/ and /ch/, we did not notice any stronger peaks from the beginning of the graph. And the sound waveforms are not periodic

Whereas in case of /i/ and /n/, we observe stronger peaks from the beginning of the graph. And sound waveforms are periodic.

With Autocorrelation function, voiced sounds have higher significant peaks in the graph

This shows that /s/, /ch/ are unvoiced sounds, /i/,/n/ are voiced sounds.

Problem statement-D

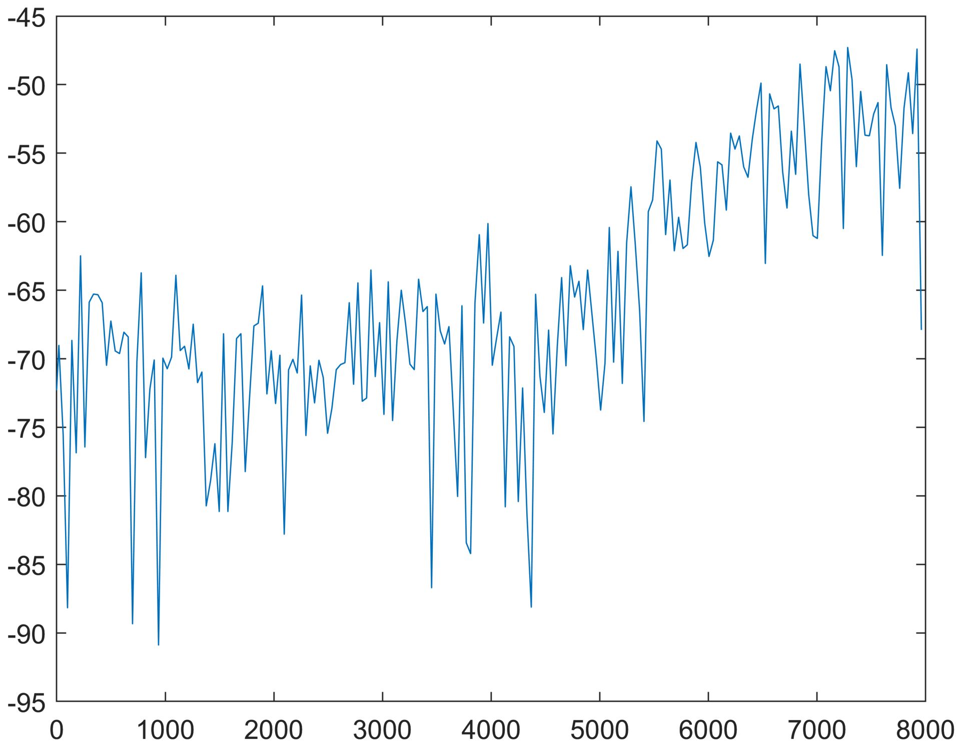
D. Plot the magnitude spectrum (with magnitude in log scale) of the 4 speech sounds. Comment/explain how the visual inspection of the spectrum can be used to classify the sound as voiced or unvoiced.

Now we plot the magnitude spectrum plots of the speech signal. We use the same method as specified to compute the N-point DFT.

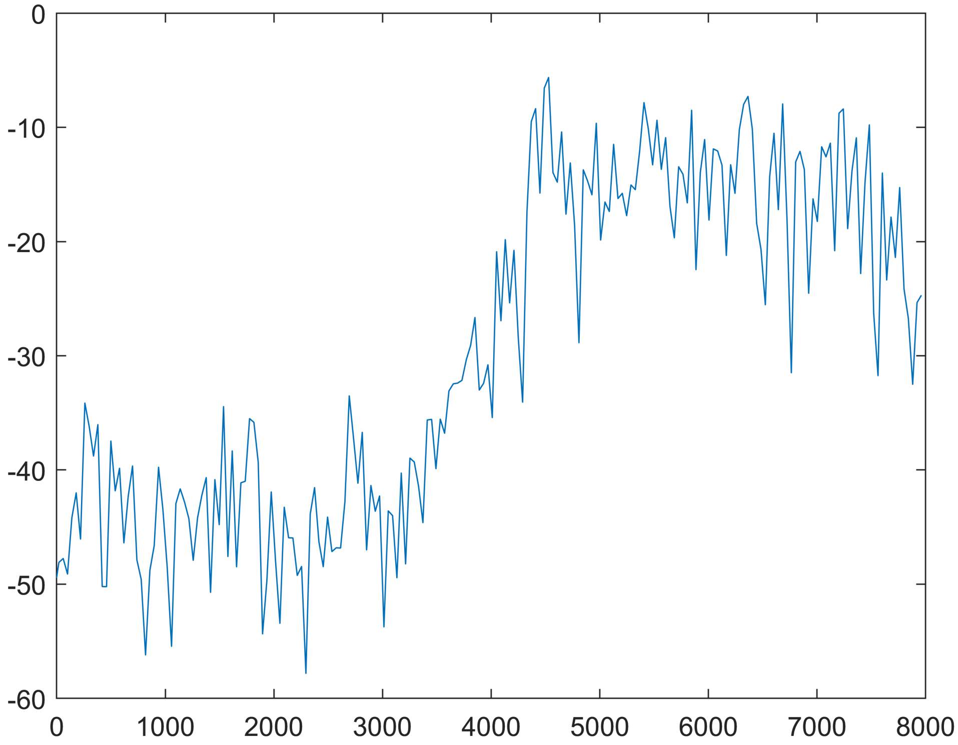
```
Y_s = fftshift(fft(y_s));
Y_ch = fftshift(fft(y_ch));
Y_i = fftshift(fft(y_i));
Y_n = fftshift(fft(y_n));
```

We have obtained the N-point DFTs of all the sounds. Now we plot the frequency spectrum for positive frequencies.

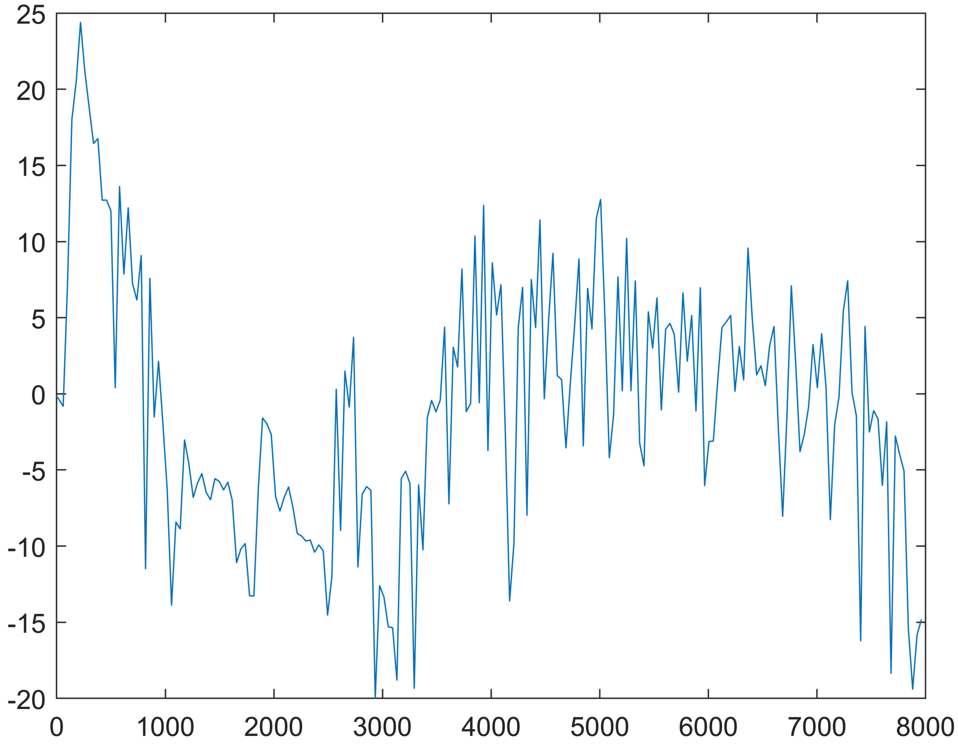
```
F_s = -fs/2 : fs/length(Y_s) : fs/2 - fs/length(Y_s);
F_ch = -fs/2 : fs/length(Y_ch) : fs/2 - fs/length(Y_ch);
F_i = -fs/2 : fs/length(Y_i) : fs/2 - fs/length(Y_i);
F_n = -fs/2 : fs/length(Y_n) : fs/2 - fs/length(Y_n);

% Plots
% /s/
plot(F_s, 20*log10(abs(Y_s)));
xlim([0, fs/2]);
```



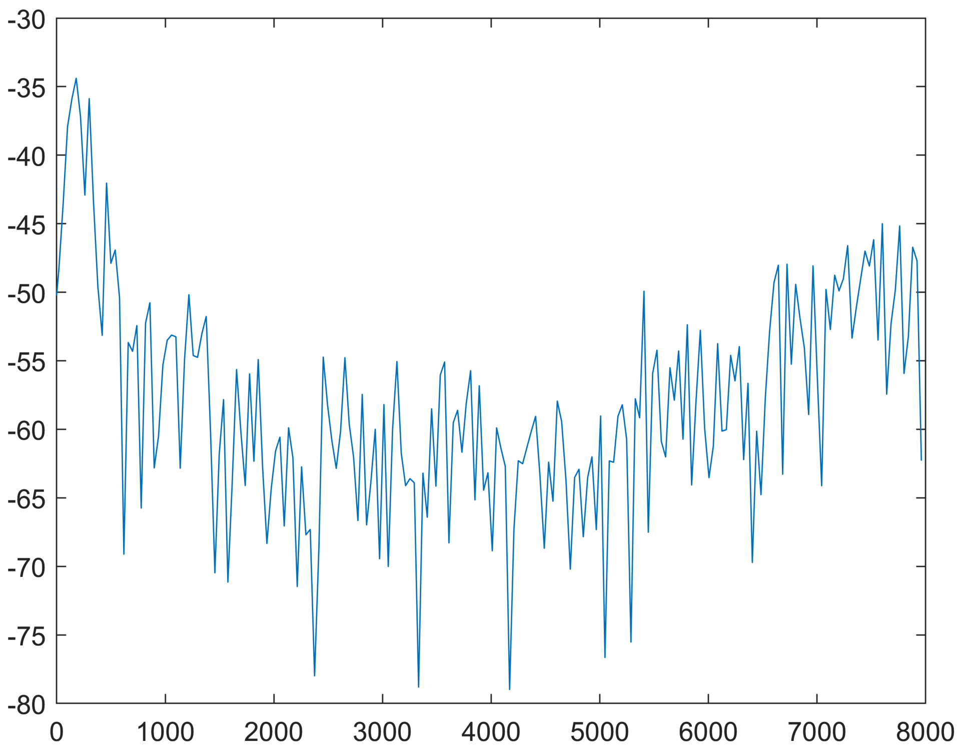
```
% /ch/
plot(F_ch, 20*log10(abs(Y_ch)));
xlim([0, fs/2]);
```



```
% /i/
plot(F_i, 20*log10(abs(Y_i)));
xlim([0, fs/2]);
```



```
% /n/
plot(F_n, 20*log10(abs(Y_n)));
xlim([0, fs/2]);
```



Comments:

By observing the spectrum of the sounds, we observe that

In case of /s/, /ch/ we dont see any presence of harmonic structure as the behaviour is not getting repeated in regular intervals.The spectrum will have more energy, typically, in the high frequency region. The spectrum will also have an upward trend starting from zero frequency and moving upwards.

In case of /i/, /n/, we observethe presence of harmonic structure as the behaviour is getting repeated in regular intervals. Also, the spectrum has more energy in the low frequency region. The spectrum will also have a downward trend starting from zero frequency and moving upwards.

Problem statement-C

C. Consider the 4 speech sounds mentioned in B and one silence segment. For each of these 5 audio segments, compute and plot Short Term Zero-Crossing rate and the Short Term Energy as a function of frame index for all the frames in the sound. Use 25 msec and 10msec as frame_size and frame_shift respectively. Comment on how you would use these time-domain features for classifying the sounds as voiced or unvoiced or silence.

```
%/s/
short_term_energy(y_s,fs,25,10);
```

```
Error using plot
Vectors must be the same length.
```

```
Error in sp_4lab>short_term_energy (line 95)
plot(t,e);
```

```
zero_crossing_rate(y_s,fs,25,10);

% /ch/
short_term_energy(y_ch,fs,25,10);
zero_crossing_rate(y_ch,fs,25,10);
%/i/
short_term_energy(y_i,fs,25,10);
zero_crossing_rate(y_i,fs,25,10);
% /n/
short_term_energy(y_n,fs,25,10);
zero_crossing_rate(y_n,fs,25,10);
% /p/
short_term_energy(y_p,fs,25,10);
zero_crossing_rate(y_p,fs,25,10);
```

Observations:

In the case of sounds like /s/, /ch/ we observe lower Short time Energy, and higher Short Term Zero Crossing rate. Therefore this sounds must correspond to unvoiced sounds.

In the case of sounds like /i/, /n/ we observe larger Short time Energy and lower Short time Zero Crossing rate. Therefore this sounds must correspond to voiced sounds.

In the case of silence: /p/

As this has lowest energy compared to the voiced, unvoiced sounds, Also relatively more number of Zero crossings compared to unvoiced sounds.

```
function[e]=short_term_energy(speechsignal,fs,Frame_size,Frame_shift)
y=speechsignal;
Frame_size= Frame_size/1000;
Frame_shift= Frame_shift/1000 ;
window_length=Frame_size*fs;
sample_shift=Frame_shift*fs;
sum=0 ;energy=[];
w=rectwin(window_length);
jj=1 ;

for i=1:(floor((length(y))/sample_shift)-ceil(window_length/sample_shift))
    for j=(((i-1)*sample_shift)+1):(((i-1)*sample_shift)+window_length)
        y(j)=y(j)*w(jj) ;
        jj=jj+1 ;
        yy=y(j)*y(j);
        sum=sum+yy;
    end
    energy=[energy sum] ;
    sum=0 ; jj=1 ;
end
w=0;
```

```
e=energy;
t = 0 : 1 / fs : (length(y) - 1) / fs;
plot(t,e);
disp(length(e));
return ;
end
function[z]=zero_crossing_rate(speechsignal,fs,Frame_size,Frame_shift)
zcr=[];
y=speechsignal;
Frame_size= Frame_size/1000;
Frame_shift= Frame_shift/1000 ;
window_length=Frame_size*fs;
sample_shift=Frame_shift*fs;
sum=0 ;energy=0 ;
w=rectwin(window_length);
jj=1 ;

for i=1:(floor((length(y))/sample_shift)-ceil(window_length/sample_shift))
    y(((i-1)*sample_shift)+1)=y(((i-1)*sample_shift)+1)*w(jj);
    jj=jj+1 ;
    for j=(((i-1)*sample_shift)+2):(((i-1)*sample_shift)+window_length)
        y(j)=y(j)*w(jj) ; jj=jj+1 ;
        yy=y(j)*y(j-1);
        if(yy<0)
            sum=sum+1;
        end
    end
    zcr=[zcr sum/(2*window_length)];
    sum=0 ; jj=1 ;
end
w=0;
z=zcr;
t = 0 : 1 / fs : (length(y) - 1) / fs;
plot(t,z)
end
```