

Speech Processing

Lab-4 Report

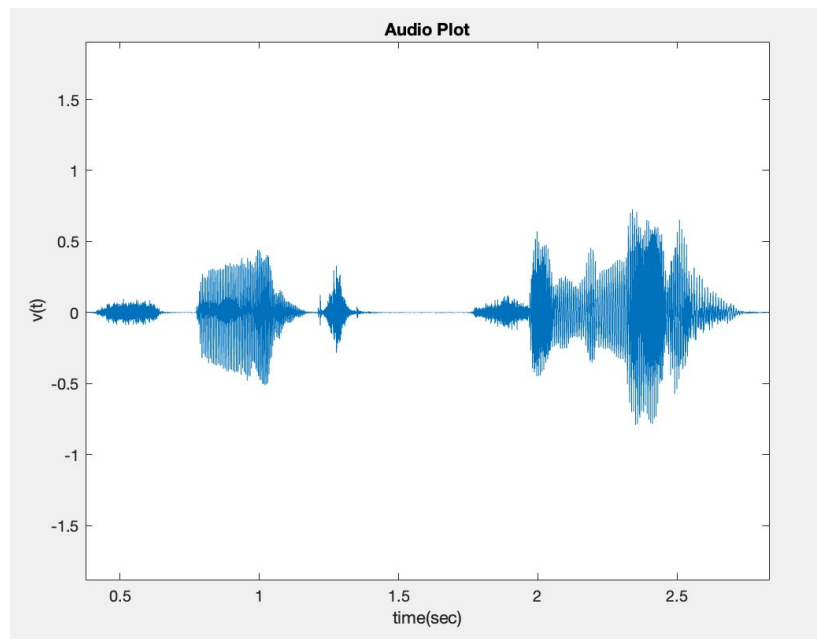
Rajat Tyagi 180020029

Part A :

Recording the phrase “Speech Signal”:

Recording specifications : $F_s = 16$ kHz and Bit resolution = 16.

Recorded audio plot:



Part B :

Selected sounds in the speech signal : ‘s’ , ‘ch’ , ‘n’ and ‘p’.

Theory :

As the name suggests, autocorrelation is the correlation of the signal with a shifted version of itself.

Autocorrelation is a useful property for analysis of the speech signal in time domain. Correlation is defined as the mutual relationship between or connection between two entities. In a periodic signal if we autocorrelate a signal with it's T (T is the period of the signal) shifted version of itself, we see that the autocorrelation value is high, as the two quantities that are being correlated are equal., hence Autocorrelation is used for determining the time period of the periodic segments of a speech signal.

Autocorrelation is given by :

$$r_k = \sum_{i=1}^{N-k-1} s_i s_{i+k}$$

Where 'k' is the shift, N is the length of the frame and r_k is the autocorrelation for the given shift 'k'.

Procedure :

- **Step 1 :**
Extract the middle 25ms part of the given sounds.
- **Step 2 :**
Pass this extracted array to the AutoCorr() function. Working of this function is explained in the code section.
- **Step 3 :**
Plot the autocorrelated array.

Code :

Autocorr() function.

```
function autocorr = AutoCorr(y)
    autocorr = [];

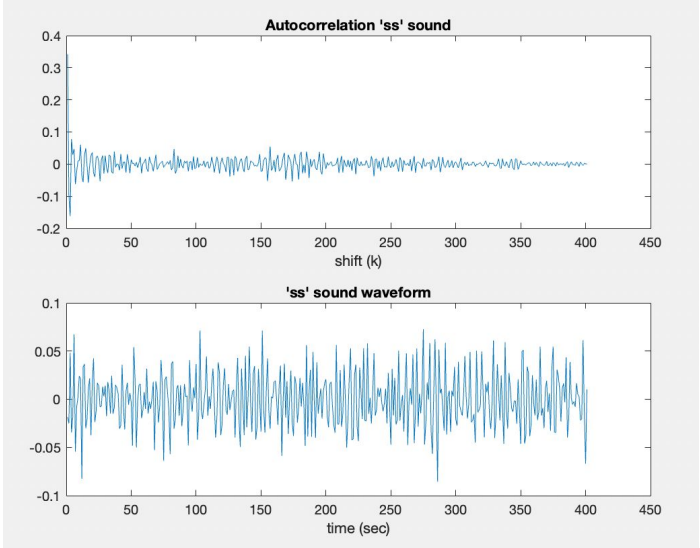
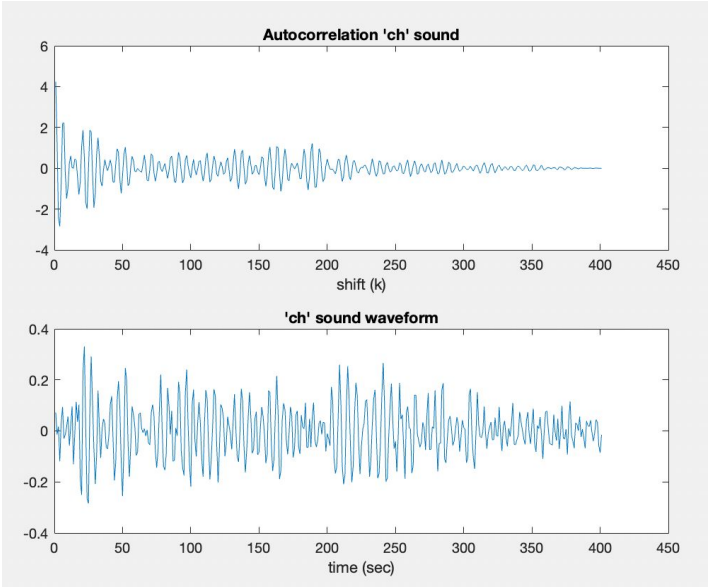
    for shift = 0 : length(y) - 1

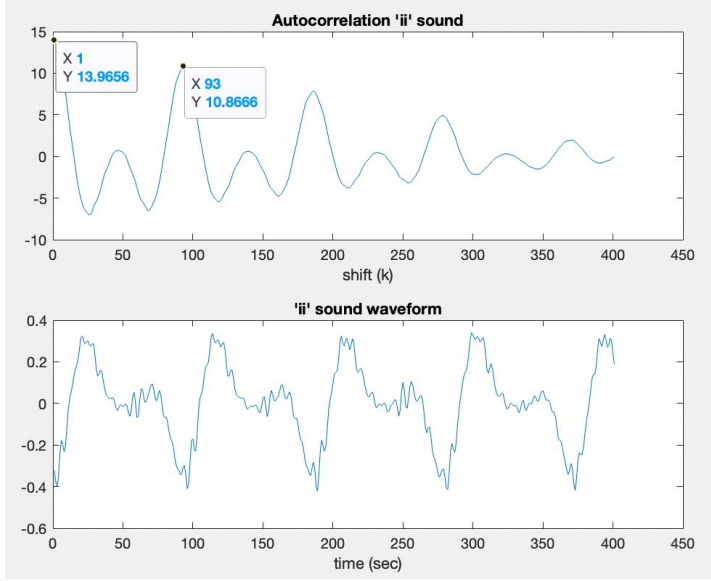
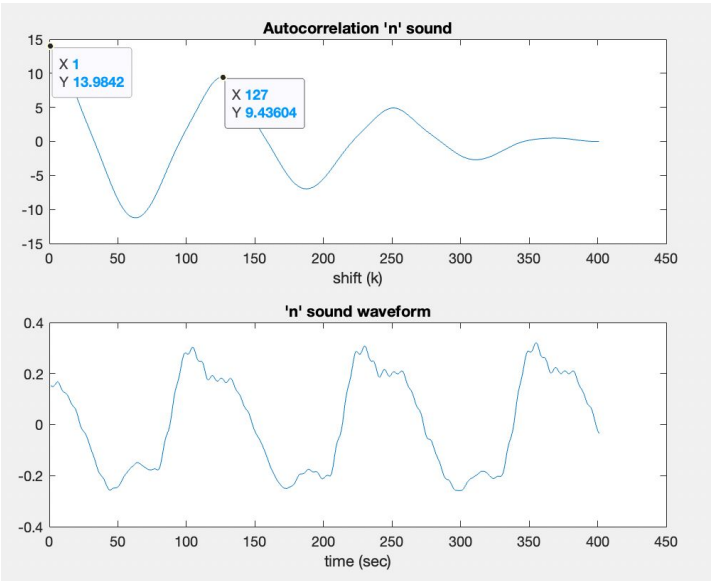
        buf = 0;
        for i = 1:length(y) - shift
            buf = buf + y(i)*y(i+shift);
        end
        autocorr = [autocorr buf];

    end
end
```

This function has two loops, first one iterates the shift and second one iterates the element of the array.

Plots

Sounds	Autocorrelation plot w.r.t shift (k)
ss	 <p>The figure for the 'ss' sound consists of two vertically stacked plots. The top plot, titled 'Autocorrelation 'ss' sound', shows the autocorrelation function with the y-axis ranging from -0.2 to 0.4 and the x-axis (shift in k) ranging from 0 to 450. The curve starts at approximately 0.35 at shift 0 and decays rapidly towards zero by shift 50, remaining near zero for the rest of the range. The bottom plot, titled ''ss' sound waveform', shows the sound waveform with the y-axis ranging from -0.1 to 0.1 and the x-axis (time in sec) ranging from 0 to 450. The waveform is a complex, noisy signal fluctuating around zero.</p>
ch	 <p>The figure for the 'ch' sound consists of two vertically stacked plots. The top plot, titled 'Autocorrelation 'ch' sound', shows the autocorrelation function with the y-axis ranging from -4 to 6 and the x-axis (shift in k) ranging from 0 to 450. The curve starts at approximately 5.5 at shift 0 and decays towards zero, with some oscillations, reaching near zero by shift 300. The bottom plot, titled ''ch' sound waveform', shows the sound waveform with the y-axis ranging from -0.4 to 0.4 and the x-axis (time in sec) ranging from 0 to 450. The waveform is a complex, noisy signal fluctuating around zero.</p>

ii	
n	

Observations:

In the autocorrelation plots of:

1. ss sound : We observe that the autocorrelation is almost like noise with no distinct peaks with some significant amplitude.
2. ch sound :In the autocorrelation function for ch sound we see a few peaks but they are not that significant.
3. ii sound : we can see that the peaks in the autocorrelation function at integral multiples of the period.

4. n sound : Similar to ii sound here also we observe peaks at integer multiples of the period.

Inference :

By looking at the observations we can design a method to determine the periodicity (and inturn voiced nature) of any signal by using the autocorrelation function.

The method can be as follows:

- 1) Find the peaks in the autocorrelation function.
- 2) If any of these peaks is above a certain threshold we can assume that shift to be the period of the signal and hence conclude that the sound contained in the signal is voiced.
- 3) Usually this threshold is set to be $0.3 r_o$ (r_o being the energy of the given segment of the signal).

ii sound : we can see that the first peak is at shift = 93 samples and has a value greater than $0.3 r_o$ hence we can conclude that the ii sound is voiced. Period = shift/ F_s = $93/16000 = 5.8125$ ms.

n sound : we can see that the first peak is at shift = 127 samples and has a value greater than $0.3 r_o$ hence we can conclude that the n sound is voiced. Period = shift/ F_s = $127/16000 = 7.9375$ ms.

Part C:

Theory:

Zero crossing rate is the number of times zero axis is crossed per frame.

The fricatives such as 'ss' have high ZCR as the source of excitation in these cases is noise. Whereas on the other hand ZCR for voiced sounds is higher because they are periodic. This property can be exploited to classify a certain segment of sound as voiced or unvoiced.

Short Time energy is the energy of the samples of audio in a given frame. We know that the voiced sounds are periodic and have higher amplitude, hence they have a higher energy whereas the unvoiced sounds have lower energies, this property can also be exploited to classify a certain segment of sound as voiced or unvoiced, if the energy of a frame is greater than a threshold it can be classified as a voiced sound otherwise unvoiced.

Procedure:

- **Step 1 :**
Extract the given sounds into an array.
- **Step 2 :**
Find the frame size and frame shift in terms of number of samples.
Frame size = 25ms = $25\text{ms} \times 16\text{kHz} = 400$ samples
Frame shift = 10ms = $10\text{ms} \times 16\text{kHz} = 160$ samples.
- **Step 3 :**
Pass the sound arrays and frame size and shift into the ZCR() and STE() functions to get the Zero Crossing Rate and Short Time Energy plots.

Code:

This part of the lab uses to functions

i) ZCR() :

```
function zcr = ZCR(y,frameSize,frameShift,text)

%number of windows
numWindows = floor((length(y) - frameSize)/frameShift) + 1;

%Array to store zcr
zcr = [];

%iterating the windows
for i = 1 : numWindows

    %determining the start and end of the given window
    winstart = 1 + frameShift*(i-1);
    winend = winstart + frameSize;

    %buffer variable
    buf = 0;
```

```

    %iterating from starting of window to end of window
    for j = wstart : winend - 1

        %incrementing if two consecutive elements have opposite sign
        if (y(j)*y(j+1) <= 0)

            buf = buf + 1;

        end

    end

    %division by frame size to compute rate
    buf = buf/frameSize;

    %appending calculated zcr for the current frame.
    zcr = [zcr buf];

end

%plotting
plot(zcr);
title("Zero Crossing Rate of " + text);
xlabel("Frame Number");
ylabel("ZCR");

end

```

This function takes the audio array, frame size and frame shift as inputs. First it determines the number of windows possible for the given input y. Then it iterates from one window to another. In each Window it counts the number of times the plot crosses the x axis, to calculate the number of crossings we check the sign of the product of two continuous samples of the signal, if it is +ve there is no crossing whereas if it is 0 or -ve that means there is a crossing.

ii) STE() :

```

function ste = STE(y,frameSize,frameShift,text)

    %number of windows
    numWindows = floor((length(y) - frameSize)/frameShift) + 1;

    %Array to store zcr
    ste = [];

    %iterating the windows
    for i = 1 : numWindows

```

```

    %determining the start and end of a given window
    winstart = 1 + frameShift*(i-1);
    winend = winstart + frameSize;

    %buffer variable
    buf = 0;

    %iterating from starting of window to end of window
    for j = winstart : winend - 1

        buf = buf + y(j)*y(j);

    end

    %appending calculated ste for the current frame.
    ste = [ste buf];

end

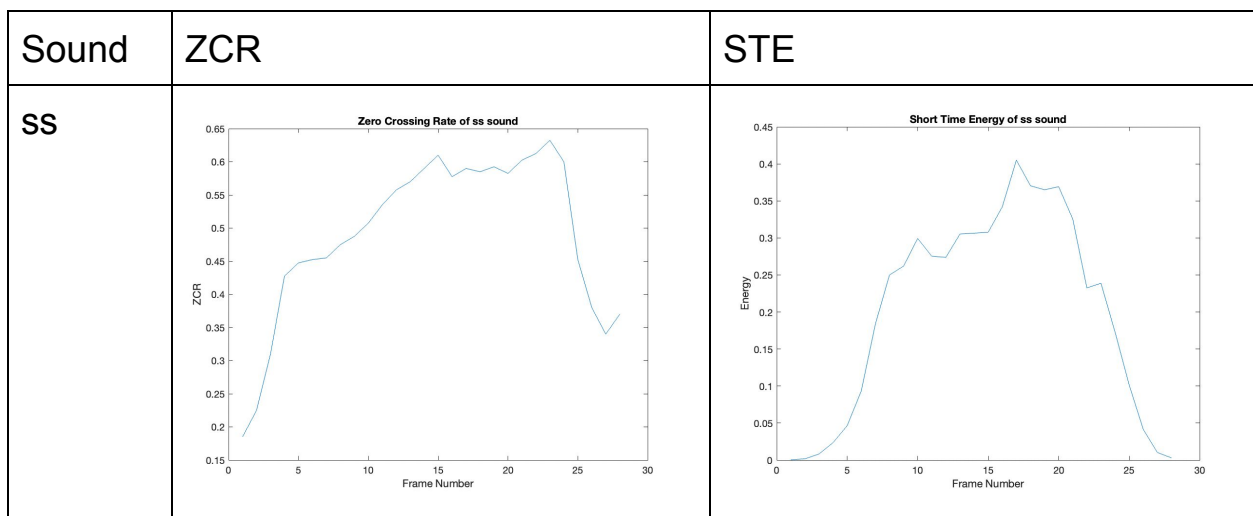
%plotting
plot(ste);
title("Short Time Energy of " + text);
xlabel("Frame Number");
ylabel("Energy");
end

```

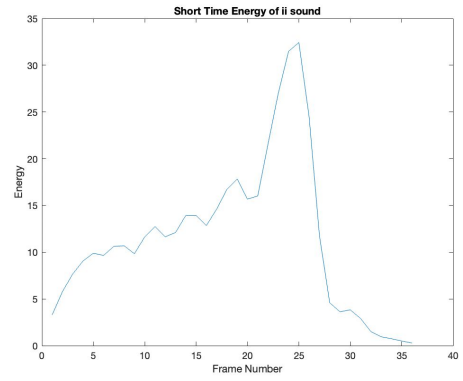
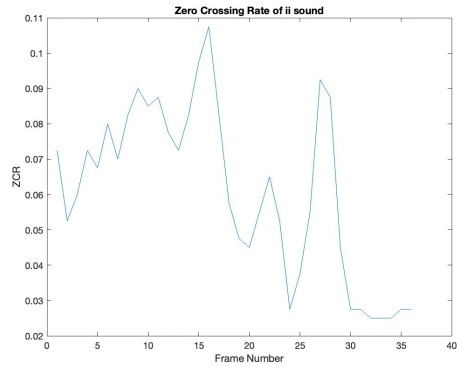
This function also takes the audio array, frame size and frame shift as inputs and determines the number of windows possible for the given input y, then it iterates from one window to another.

In each Window sums up the amplitude squared of each term to get the total energy present in the frame.

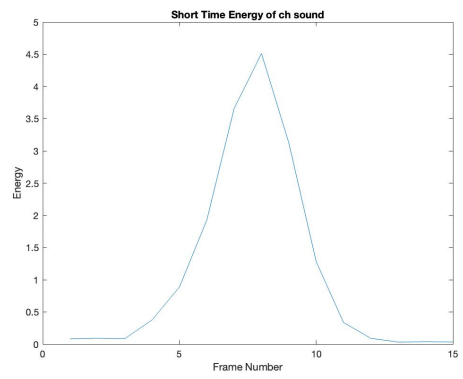
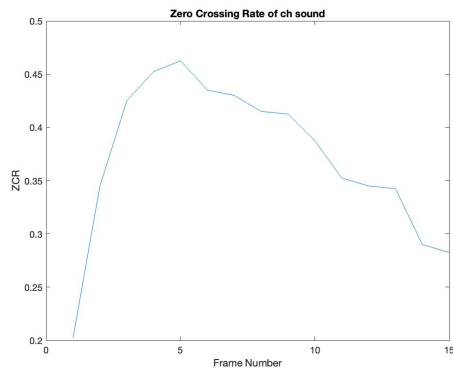
Plots:



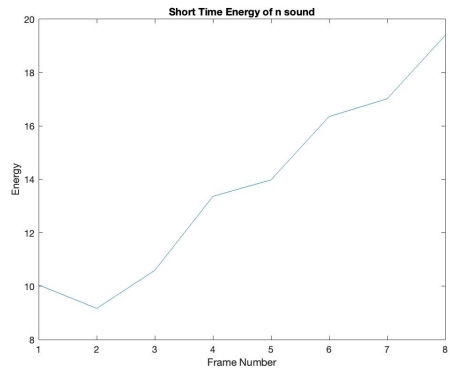
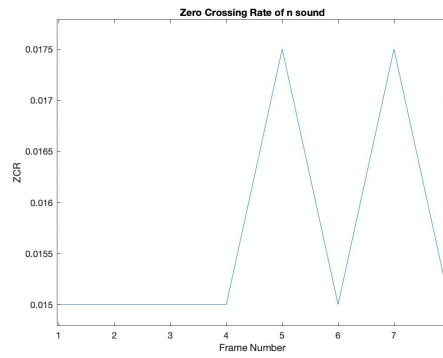
ii



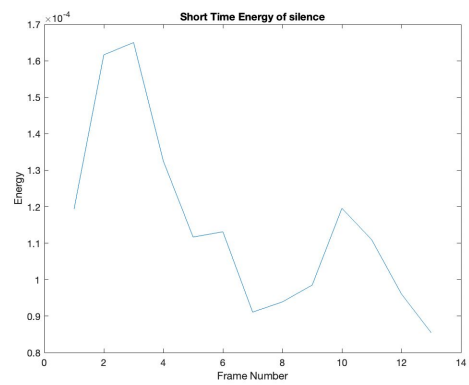
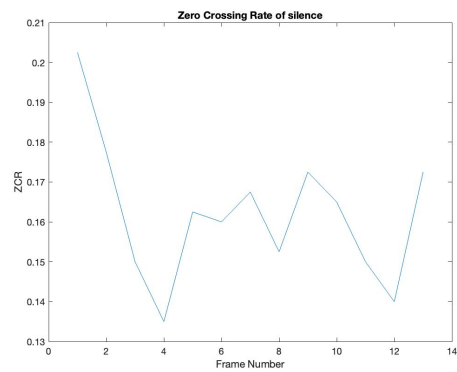
ch



n



silence



Observations:

ZCR for the:

1. ss sound : it is higher compared to the other sounds. It is around 0.6. Its high zcr is intuitive because of the frication present in 'ss'.
2. ii sound : it has a very low zcr (below 0.1), this is due to the periodic nature of the ii sound.
3. ch sound : it has a higher zcr i.e around 0.45, this is due to the frications present in the 'ch' sound.
4. n sound : it has the lowest zcr, this is evident because of 'n' sound's periodic nature.
5. Silence : it's zcr is higher than 'ii' and 'n' sounds, but is lesser than 'ch' and 'ss'.

Short time energy for,

1. ss sound : The STE for 'ss' sound is very less lesser than 02, because it has a lower amplitude.
2. ii sound : ii sound has the highest STE around 10-30, because of its high amplitudes.
3. ch sound : it has higher energy than 'ss' but it is lower than ii sound below 4.5.
4. n sound : n sound also has higher energy around 10 - 20.
5. Silence : The energy of silence is so small that it can be assumed to be zero, it is in the order of 10^{-4} .

Inference:

For ZCR:

We can find two thresholds (alpha and beta), such that all the sounds having ZCR greater than alpha can be categorised as unvoiced, whereas all the sounds having ZCR lower than beta can be categorised as voiced.

For Short Time Energy:

We had a significant zcr in case of silence, but STE is almost 0 for silence, hence silence can easily be detected via STE,

Additionally we can also set a threshold such that all the sounds above this threshold can be categorized as voiced sounds.

Part D :

Theory:

Voiced sounds:

The sounds which are nearly periodic in nature can be classified as voiced sounds. Hence they have fixed peaks in their frequency domain and virtually no energy in other frequencies.

Unvoiced sounds:

The sounds whose time domain waveform looks like random noise, as they are generated from random noise as the source of excitation. We know random noise contains all the frequencies, hence these unvoiced sounds have energy at higher frequencies as well.

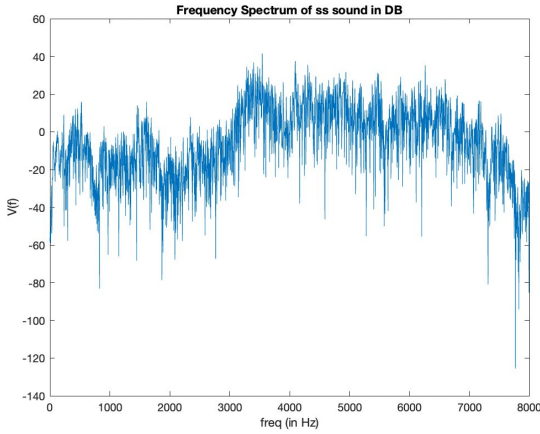
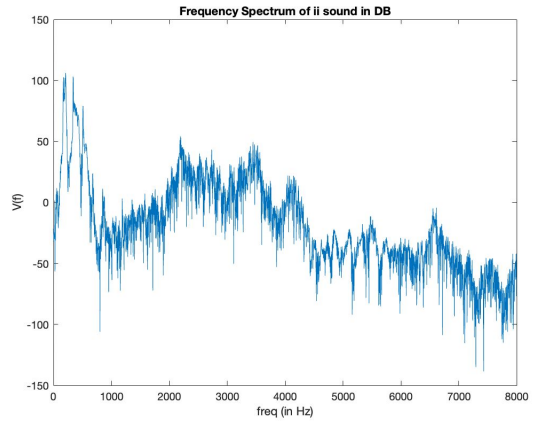
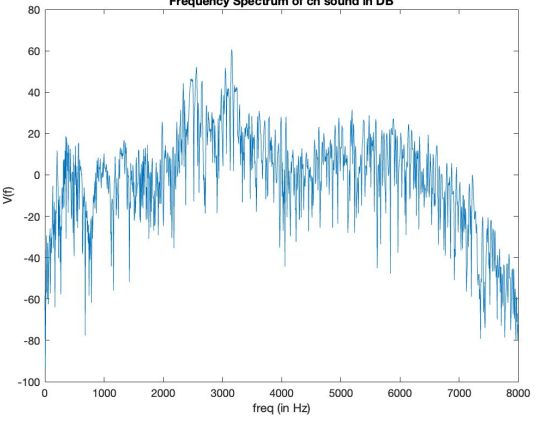
Procedure:

- **Step 1 :**

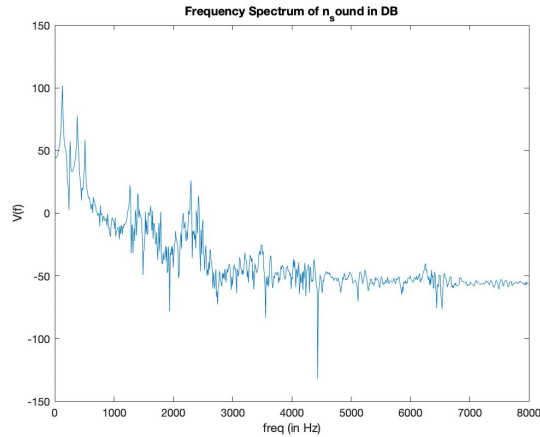
Pass the extracted sounds to the `speechfft()` function. To get the magnitude spectrum of the input sound in DB.

Code is the same which was used in the previous two labs hence not explained here.

Plots:

sound	Frequency Spectrum
ss	 <p>The plot shows the frequency spectrum of the 'ss' sound. The x-axis is labeled 'freq (in Hz)' and ranges from 0 to 8000. The y-axis is labeled 'V(f)' and ranges from -140 to 60. The spectrum is highly noisy, with a general level between -20 and 20 dB across most frequencies, and a slight dip below -80 dB at the highest frequencies.</p>
ii	 <p>The plot shows the frequency spectrum of the 'ii' sound. The x-axis is labeled 'freq (in Hz)' and ranges from 0 to 8000. The y-axis is labeled 'V(f)' and ranges from -150 to 150. The spectrum shows a very high peak near 0 Hz (around 100 dB), followed by a sharp drop and then a noisy spectrum fluctuating between -50 and 50 dB for the remainder of the frequency range.</p>
ch	 <p>The plot shows the frequency spectrum of the 'ch' sound. The x-axis is labeled 'freq (in Hz)' and ranges from 0 to 8000. The y-axis is labeled 'V(f)' and ranges from -100 to 80. The spectrum is noisy, with a notable peak around 3000-4000 Hz reaching approximately 60 dB, and a gradual decline in energy towards the 8000 Hz mark.</p>

n



Observations:

We observe that the 'ch' and 'ss' sounds have significant energy in the higher frequencies, whereas ii and n sound have almost no power in higher frequencies instead they have prominent peaks at lower frequencies.

Inference and Conclusion:

As we know from the theory of voiced and unvoiced speech, the voiced components have all their energies in lower frequencies and unvoiced components have significant energy in higher frequencies we can conclude that 'ch' and 'ss' are unvoiced whereas 'ii' and 'n' are voiced sounds.