EE-414 Speech Processing Lab

Lab-5

Aim

- **To study different sound units present in majority of Indian languages.**
- **To understand the production mechanism of each sound unit.**
- **To learn the time domain and frequency domain characteristics of different sound units.**

Introduction:

Speech signals are composed of different sound units which differ from language to language. It is important to get a feel of how these sound units are produced, by analysing its time, spectral properties and spectrogram. There are different categories of speech sounds such as vowels(short vowels, long vowels, diphthongs) and consonants(short consonants, nasals, semi-vowels, fricatives, affricates). Speech signals are formed due to the combinations of different sound units of a particular language, hence it is important to study these sound units individually. The sounds produced are either voiced(when there is no interruption of air flow from glottis to lips by voiced excitation) or unvoiced (where there is narrow constriction/obstruction in the air flow by unvoiced/voiced excitation).

A.

Short vowels, Long vowels and Diphthongs

1. Record the sounds of any one short vowel sound, long vowel sound and a diphthong (Also, record the two sounds present in the diphthong).
2. Plot the time domain waveform, magnitude spectrum and the spectrogram for each of the above sounds.
3. Inspect each of the above plots and write your observations comparing them.

Theory:

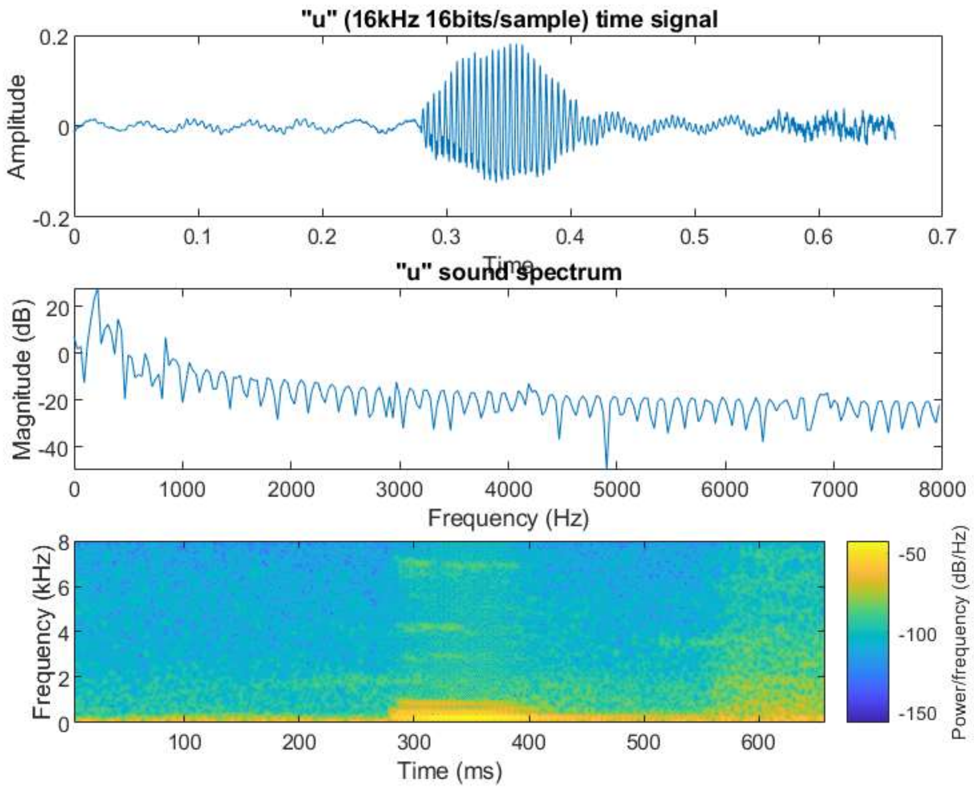
Short vowels, long vowels and diphtongs are the categories of classification for vowels. Short vowels and long vowels are produced in a similar manner however the duration of long vowels is much longer (about twice as long) as compared to short vowels. Diphtongs are combinations of different vowels, where there is a change from one vowel to another vowel during the production of sound (i.e. the vocal tract changes as the two sounds are produced)

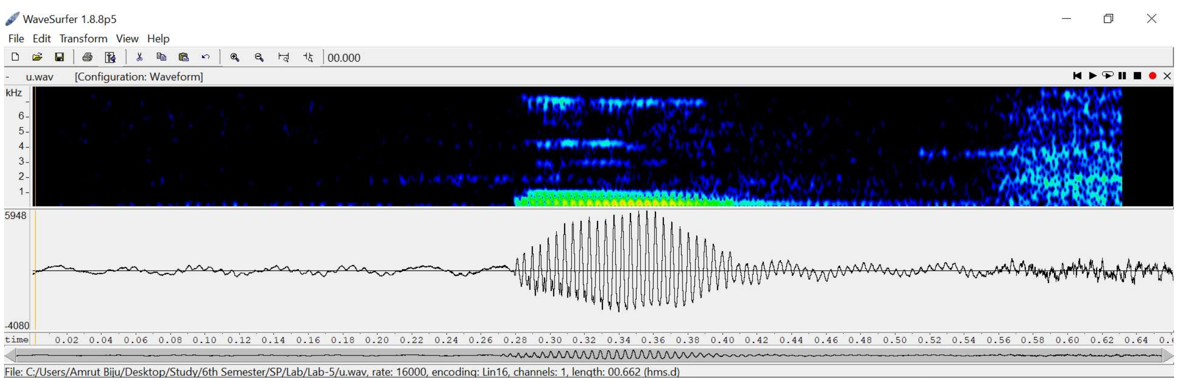
Procedure:

We are analysing "u" (short vowel), "uu" (long vowel) and "ou" (diphthong). We record these sounds via the software *Audacity,* with 16 bits/sample and 16 kHz sampling frequency. We then plot time domain signal using user-defined function *plot_time_signal()* and frequency spectrum using user-defined function *plot_freq_spectrum().* We use inbuilt matlab function *spectrogram()* to plot the spectrogram. We then analyse the plots and make some observations.

Code and Plots:

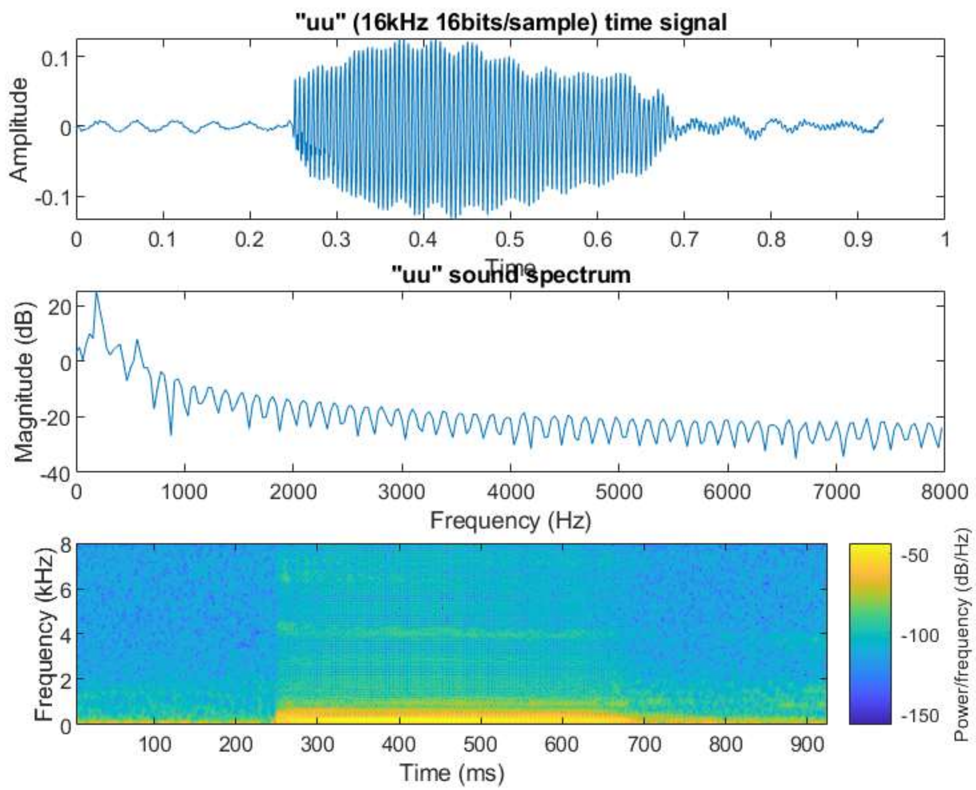
```
close all; clc; clear all;

[u_sound,fs]=audioread('u.wav');

[uu_sound,fs]=audioread('uu.wav');

[ou_sound,fs]=audioread('ou.wav');

[a_sound,fs]=audioread('a.wav');

% "u"
subplot(311);
plot_time_signal(u_sound, fs, '"u" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(u_sound((0.32*fs):(int64((0.32+0.025)*fs))), fs, '"u" sound spectrum');
subplot(313);
spectrogram(u_sound, 128, 125, 512, fs, 'yaxis');
```

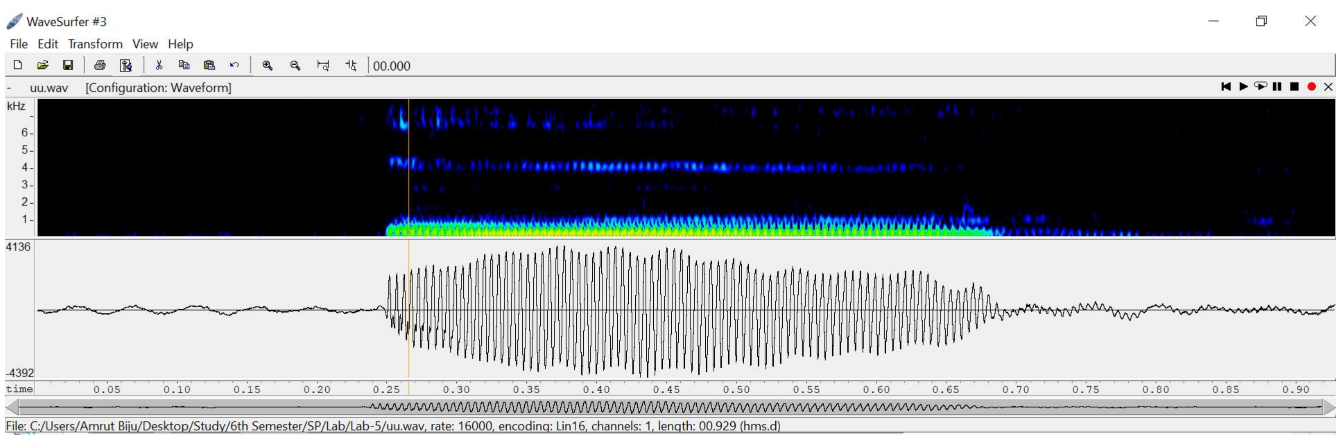




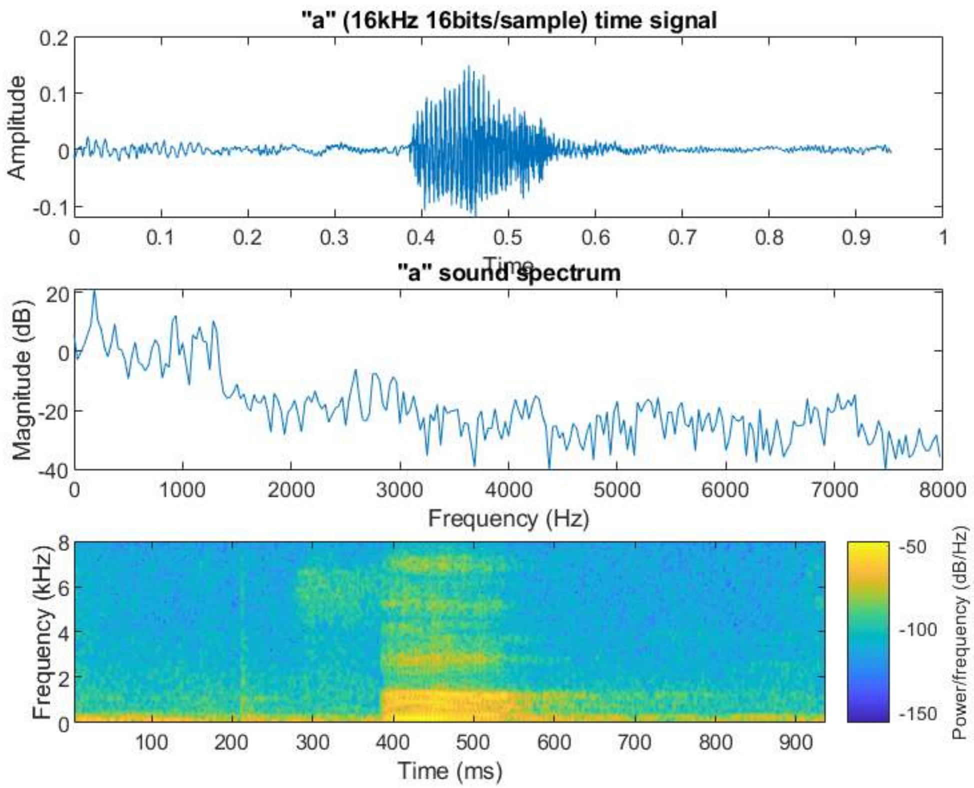

```
% "uu"
close hidden
subplot(311);
plot_time_signal(uu_sound, fs, '"uu" (16kHz 16bits/sample) time signal');
subplot(312);
```

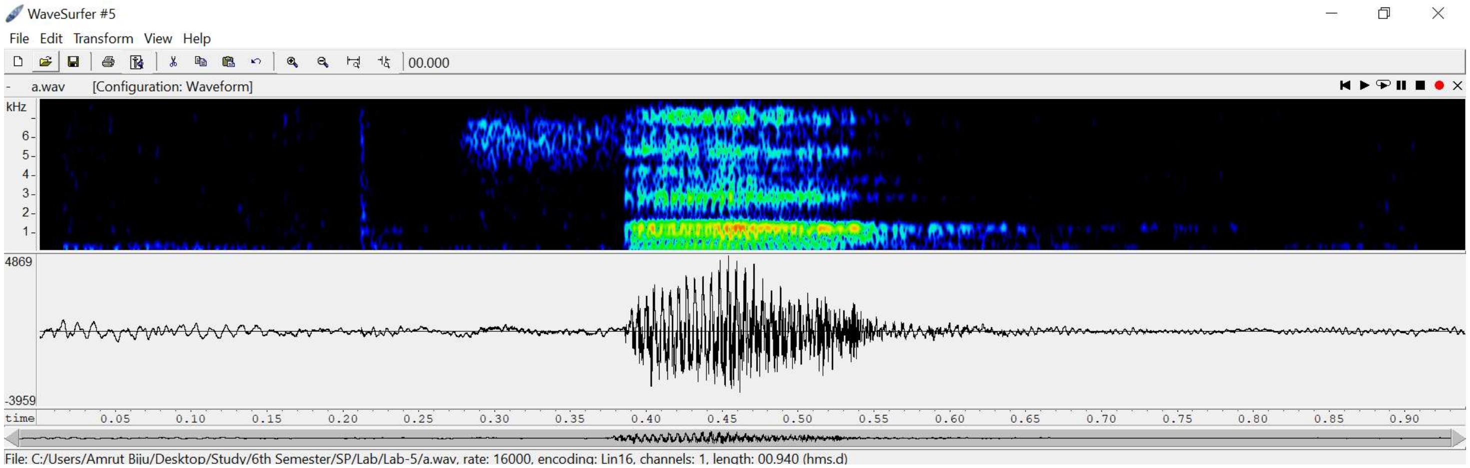
```
plot_freq_spectrum(uu_sound((0.30*fs):(int64((0.30+0.025)*fs))), fs, '"uu" sound spectrum');
subplot(313);
spectrogram(uu_sound, 128, 125, 512, fs, 'yaxis');
```



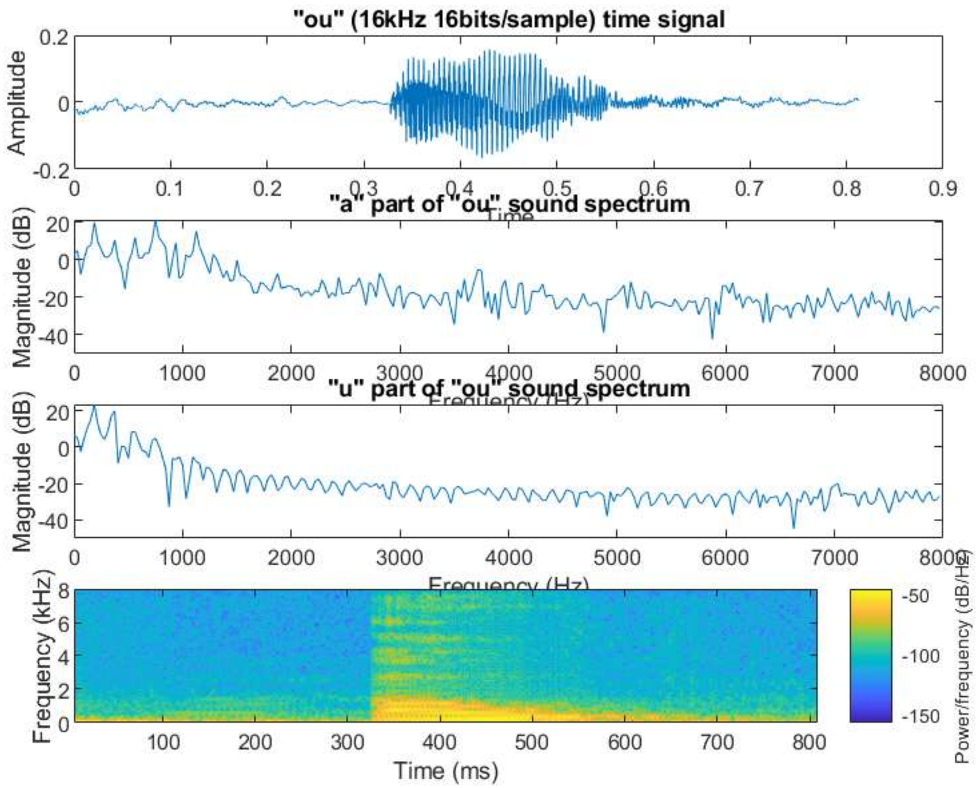


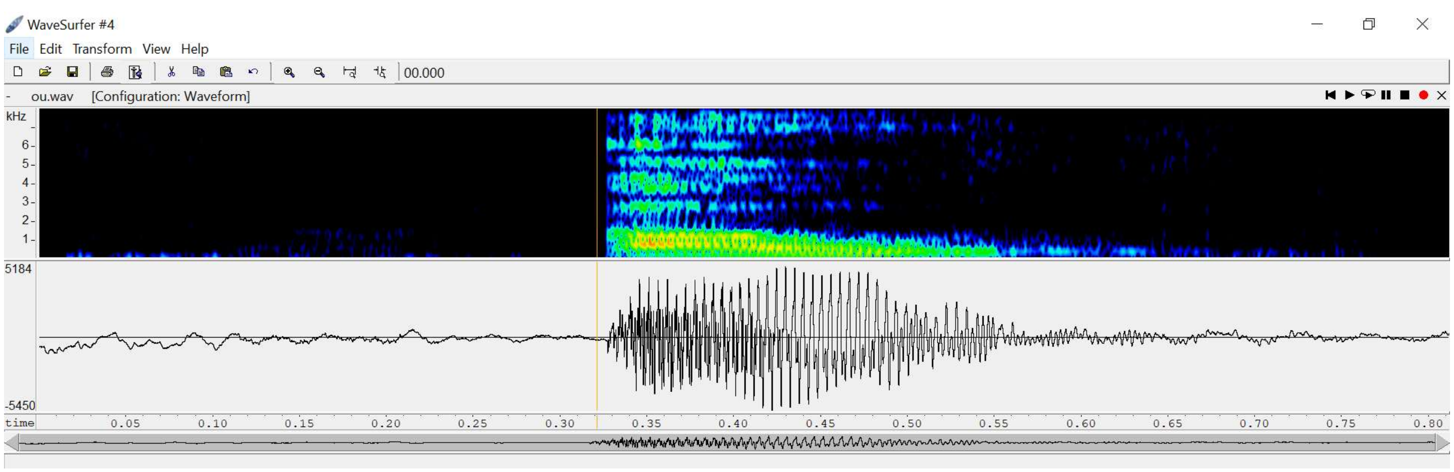
```
% "a" (for comparison, for diphtong part)
close hidden
subplot(311);
plot_time_signal(a_sound, fs, '"a" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(a_sound((0.40*fs):(int64((0.40+0.025)*fs))), fs, '"a" sound spectrum');
subplot(313);
spectrogram(a_sound, 128, 125, 512, fs, 'yaxis');
```






```
% "ou"
close hidden
subplot(411);
plot_time_signal(ou_sound, fs, '"ou" (16kHz 16bits/sample) time signal');
subplot(412);
plot_freq_spectrum(ou_sound((0.35*fs):(int64((0.35+0.025)*fs))), fs, '"a" part of "ou" sound spectrum');
subplot(413);
plot_freq_spectrum(ou_sound((0.47*fs):(int64((0.47+0.025)*fs))), fs, '"u" part of "ou" sound spectrum');
subplot(414);
spectrogram(ou_sound,  128, 125, 512, fs,'yaxis');
```





Observations:

We see that "u" and "uu" have same time domain structure except that for "uu" the duration is longer. We can also see that the spectral properties are also same through frequency spectrum and spectrogram. For the diphthong "ou", we see that it has "a" and "u" sounds by seeing the spectrograms and the frequency spectrum. We see that first half has the same frequency spectrum as "a" sound and 2nd half has same frequency spectrum as "u" sound.

B.

Stop Consonants

1. Pick up any one of the POA(Position of Articulation) types and record the sounds present in the respective row for all the MOA(Manner of Articulation) types.
2. Plot the time domain waveform, the magnitude spectrum and the spectrogram for each of the above sounds.
3. Inspect the above plots and describe the various sub phonetic events that take place, their relative duration and how they vary across different kinds of MOA.

Theory:

There is stoppage of air flow in the vocal tract when stop consonant sounds are produced (hence the name stop consonant). The stop consonant are categorized by Place of Articulation (POA) and Manner of Articulation (MOA)

POA is categorized as Velar, Palatal, Alveolar, Dental and Bilabial. These categories show us where the stoppage of air flow occurs.

MOA is categorized as unvoiced unaspirated (UU), unvoiced aspirated (UA), voiced unaspirated (VU) and voiced aspirated (VA). These categories show how the excitation happens in the vocal tract (either voicing or aspiration takes place or not).

Procedure:

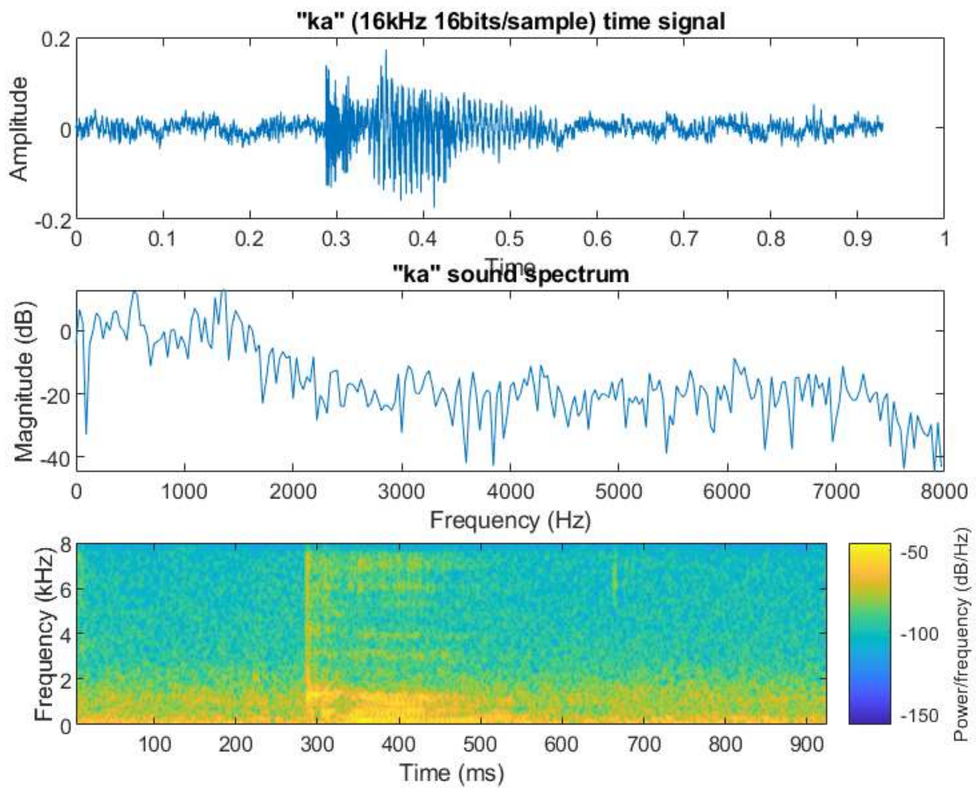
We are analysing the velar sounds i.e. Ka, Kha, Ga, Gha. We record these sounds via the software *Audacity,* with 16 bits/sample and 16 kHz sampling frequency. We then plot time domain signal using user-defined function *plot_time_signal()* and frequency spectrum using user-defined function *plot_freq_spectrum().* We use inbuilt matlab function *spectrogram()* to plot the spectrogram. We then analyse the plots and make some observations.

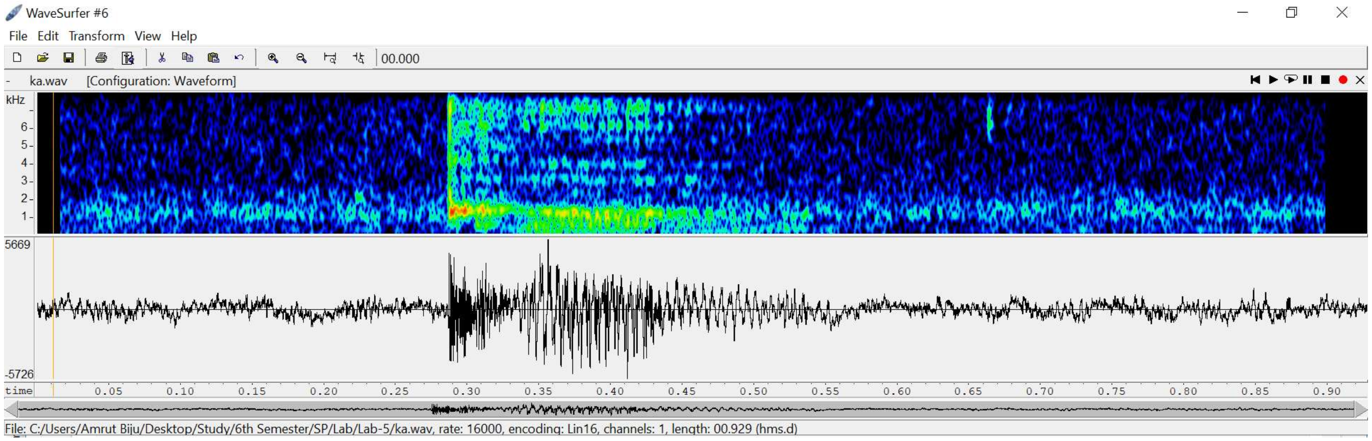
Code and Plots:

```
close all; clc; clear all;
close hidden

[ka_sound,fs]=audioread('ka.wav');

[kha_sound,fs]=audioread('kha.wav');
```

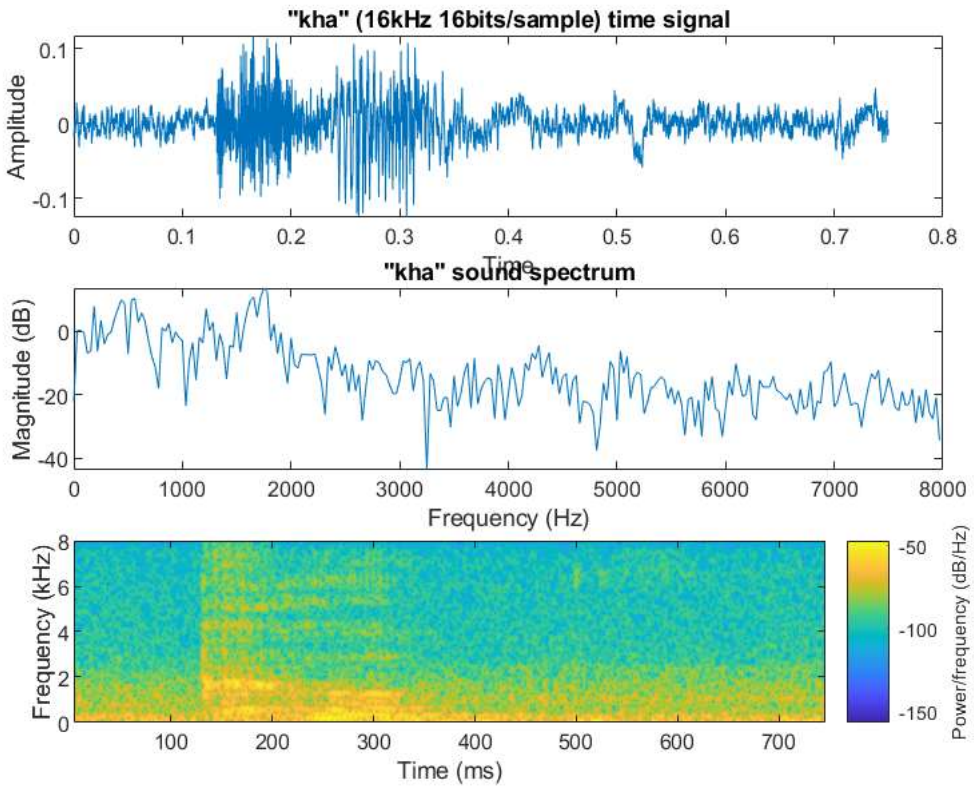
```
[ga_sound,fs]=audioread('ga.wav');

[gha_sound,fs]=audioread('gha.wav');
```

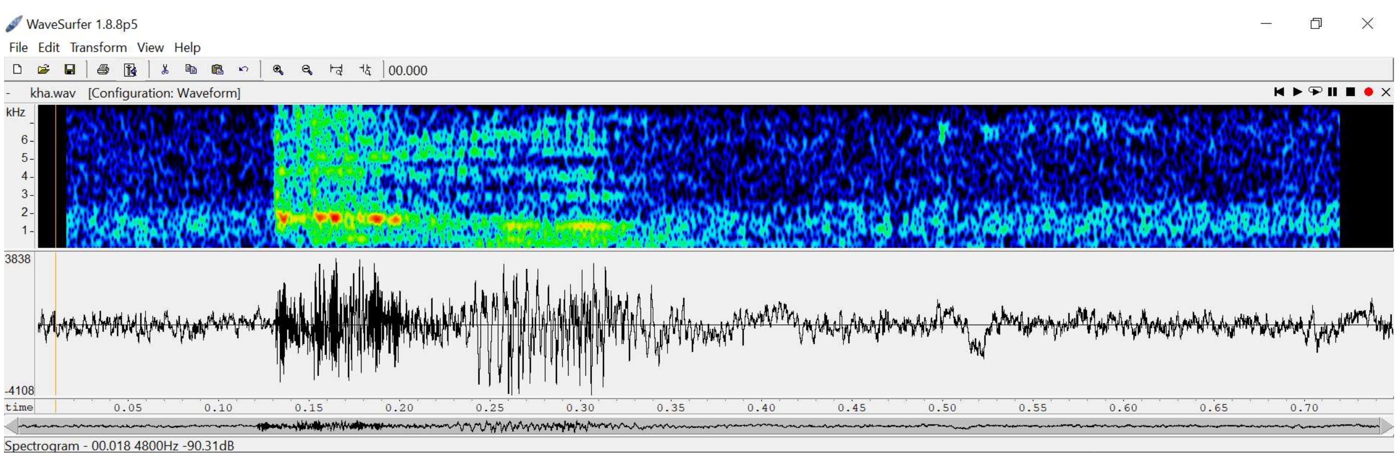
```
% /ka/
subplot(311);
plot_time_signal(ka_sound, fs, '"ka" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(ka_sound((0.3*fs):(int64((0.3+0.025)*fs))), fs, '"ka" sound spectrum');
subplot(313);
spectrogram(ka_sound, 128, 125, 512, fs, 'yaxis');
```





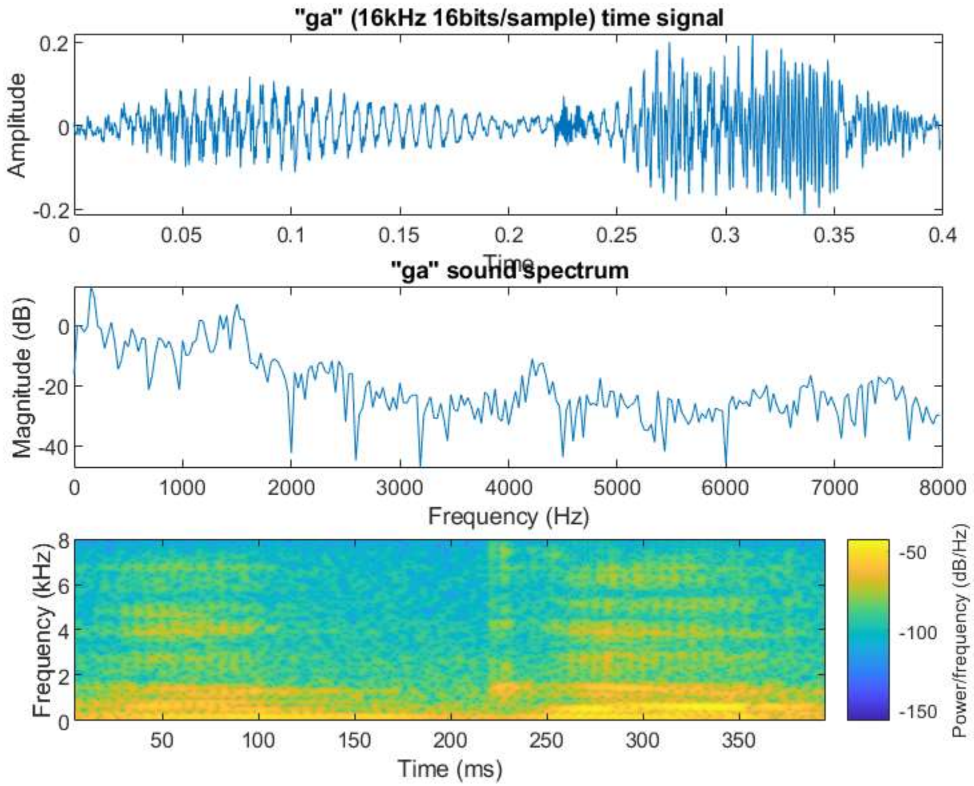

```
% /kha/
close hidden
subplot(311);
plot_time_signal(kha_sound, fs, '"kha" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(kha_sound((0.15*fs):(int64((0.15+0.025)*fs))), fs, '"kha" sound spectrum');
subplot(313);
spectrogram(kha_sound, 128, 125, 512, fs, 'yaxis');
```

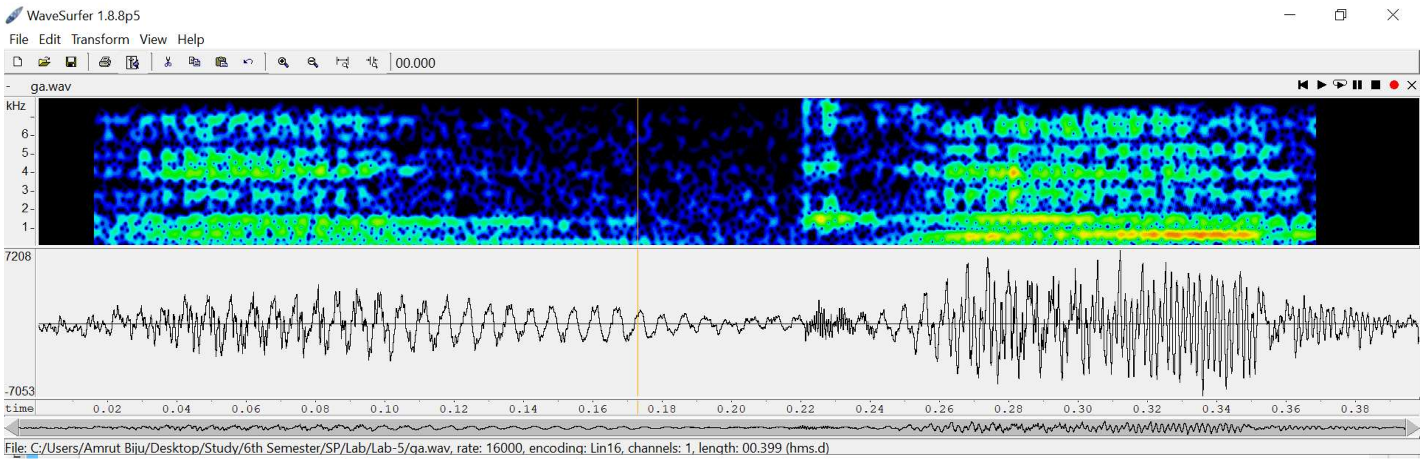




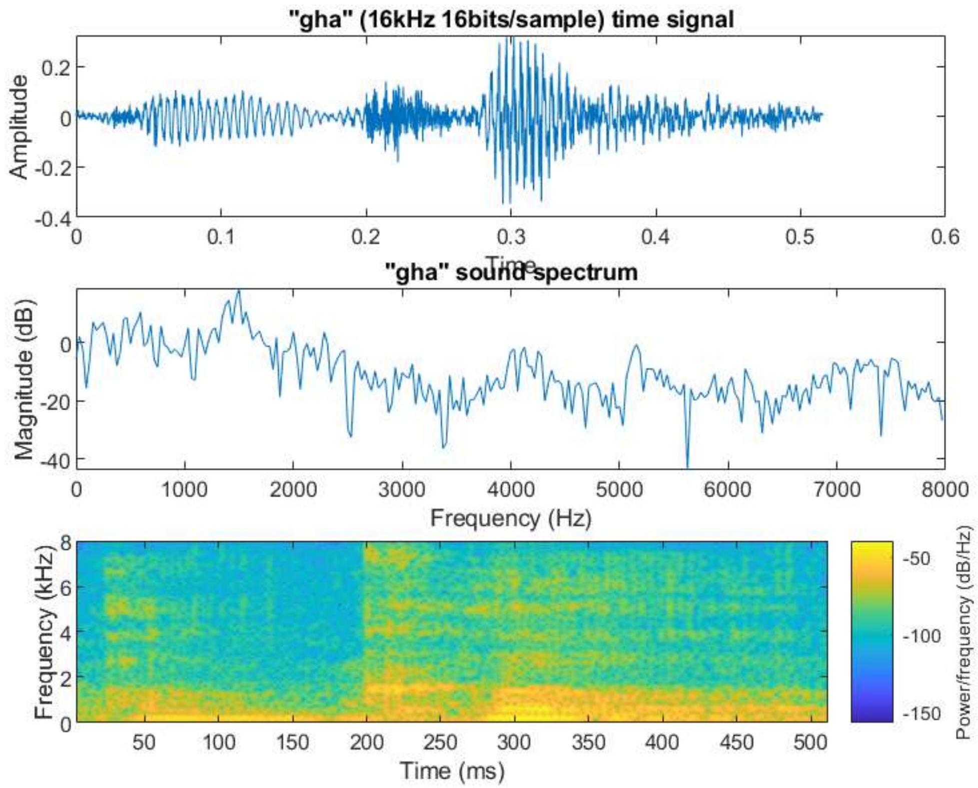

```
% /ga/
close hidden
subplot(311);
```

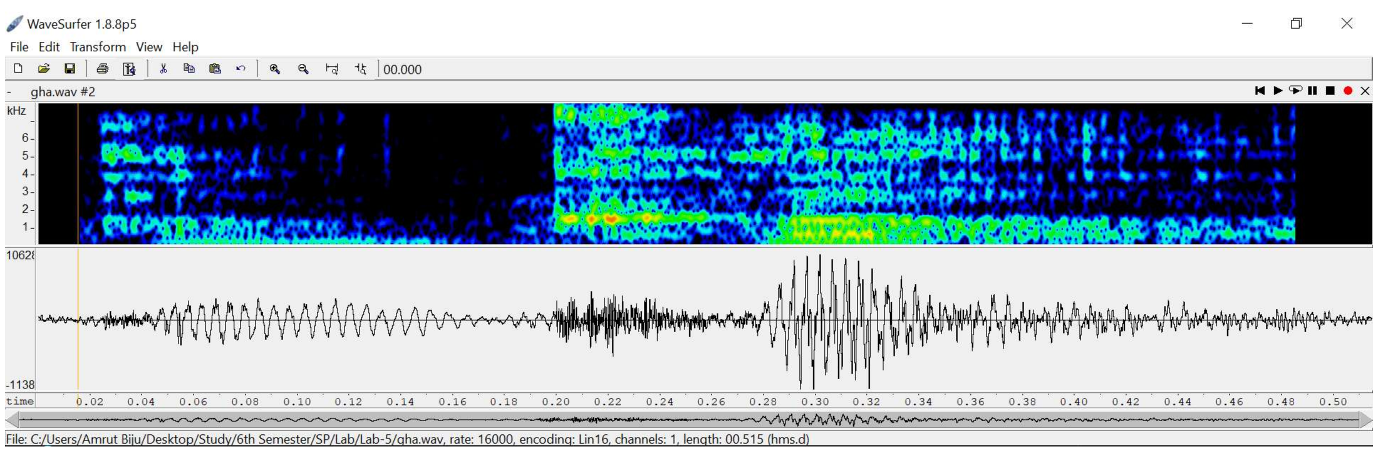
```
plot_time_signal(ga_sound, fs, '"ga" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(ga_sound((0.225*fs):(int64((0.225+0.025)*fs))), fs, '"ga" sound spectrum');
subplot(313);
spectrogram(ga_sound, 128, 125, 512, fs, 'yaxis');
```






```
% /gha/
close hidden
subplot(311);
plot_time_signal(gha_sound, fs, '"gha" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(gha_sound((0.21*fs):(int64((0.21+0.025)*fs))), fs, '"gha" sound spectrum');
subplot(313);
spectrogram(gha_sound, 128, 125, 512, fs, 'yaxis');
```





Observations:

Ka: (UU) There is no periodicity (from time domain plot). We can see that there is a closure 1st then a burst is followed (wide range of frequencies with considerable energy) .

Kha: (UA) There is no periodicity (from time domain plot). We can see an additional event along with closure and burst, which is aspiration (a lot of frequency components with considerable present). The duration of "kh" is longer than "k".

Ga: (VU) There is periodicity (from time domain plot). We see a lesser range of frequency components covered as compared to UU sounds (as here it is voiced). Here also we see a closure and burst but with lesser frequencies as compared to UU sounds. Amplitude is much lower than unvoiced here.

Gha: (VA) There is periodicity (from time domain plot). We see a lesser range of frequency components covered as compared to UA sounds (as here it is voiced). Here also we see a closure, burst and aspiration but with lesser frequencies as compared to UA sounds. The duration of "gh" is longer as compared to "g". Amplitude is much lower than unvoiced.

C.

Nasals

1. Record the sounds of any two nasal sounds and plot their time domain waveform, the magnitude spectrum and the spectrogram.
2. Inspect the above plots and write your observations. Also, comment on how they compare to vowel sounds.

Theory:

Nasal sounds are produced primarily by flow of through the nasal cavity. These sounds are similar to vowels, and are voiced in nature. The nasal sounds present are : ng, nj, nx, n, m. These are continuant sounds.

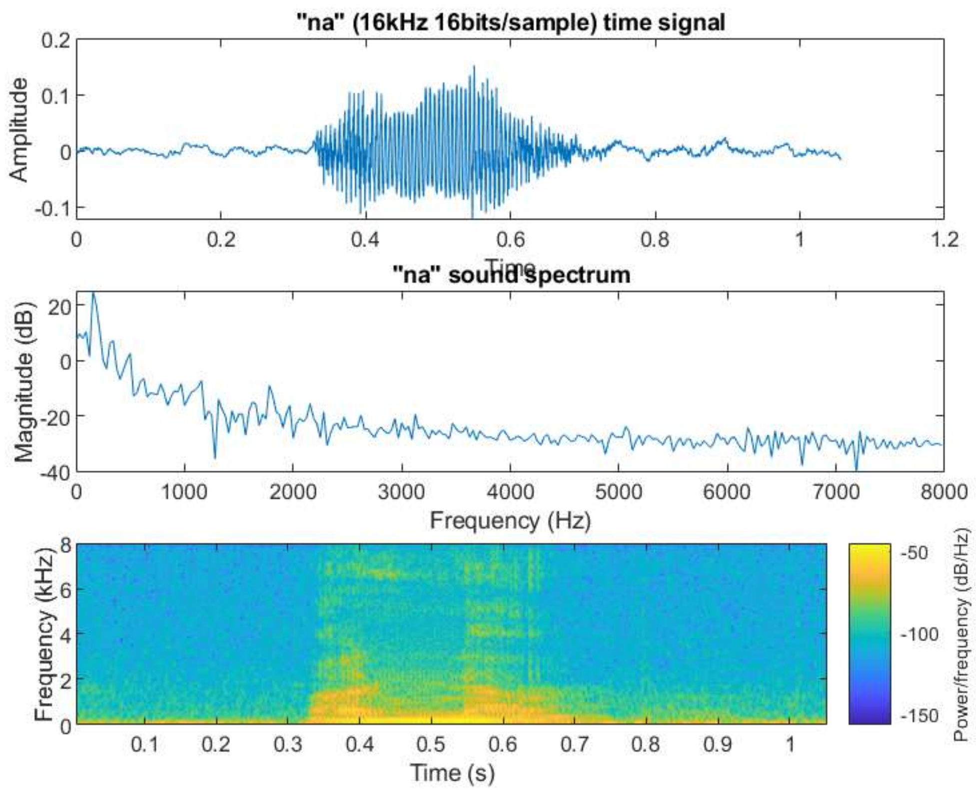
Procedure:

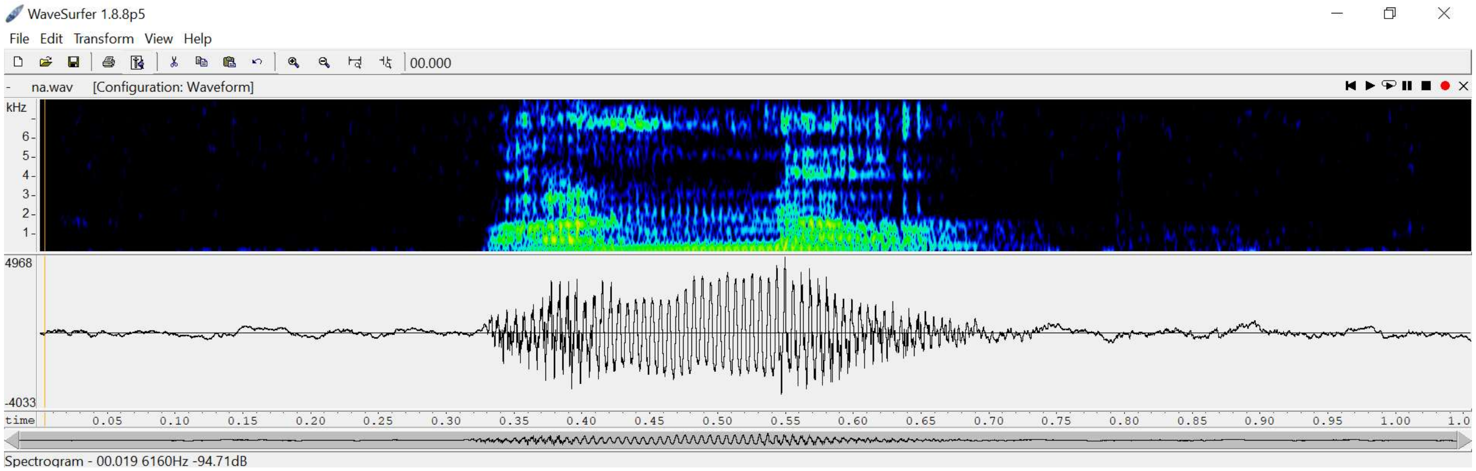
We are analysing the nasal sounds "Na" and "Ma". We record these sounds via the software WaveSurfer, with 16 bits/sample and 16 kHz sampling frequency. We then plot time domain signal using user-defined function *plot_time_signal()* and frequency spectrum using user-defined function *plot_freq_spectrum().* We use inbuilt matlab function *spectrogram()* to plot the spectrogram. We then analyse the plots and make some observations.

Code and Plots:

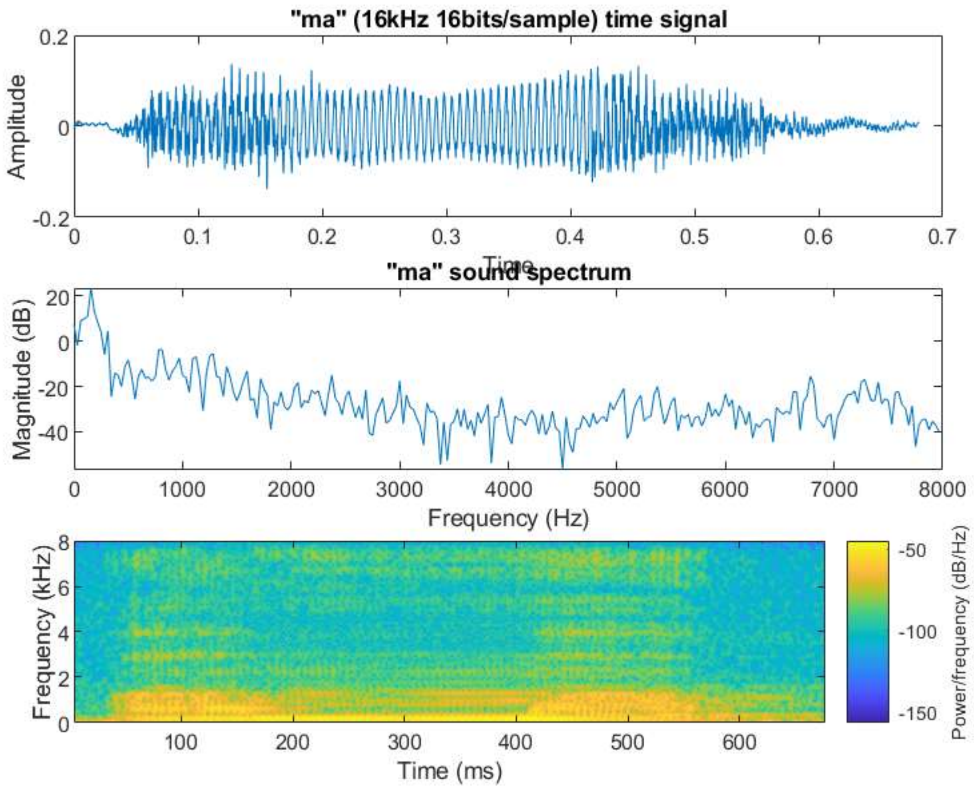
```
close all; clc; clear all;
close hidden

[na_sound,fs]=audioread('na.wav');

[ma_sound,fs]=audioread('ma.wav');
```

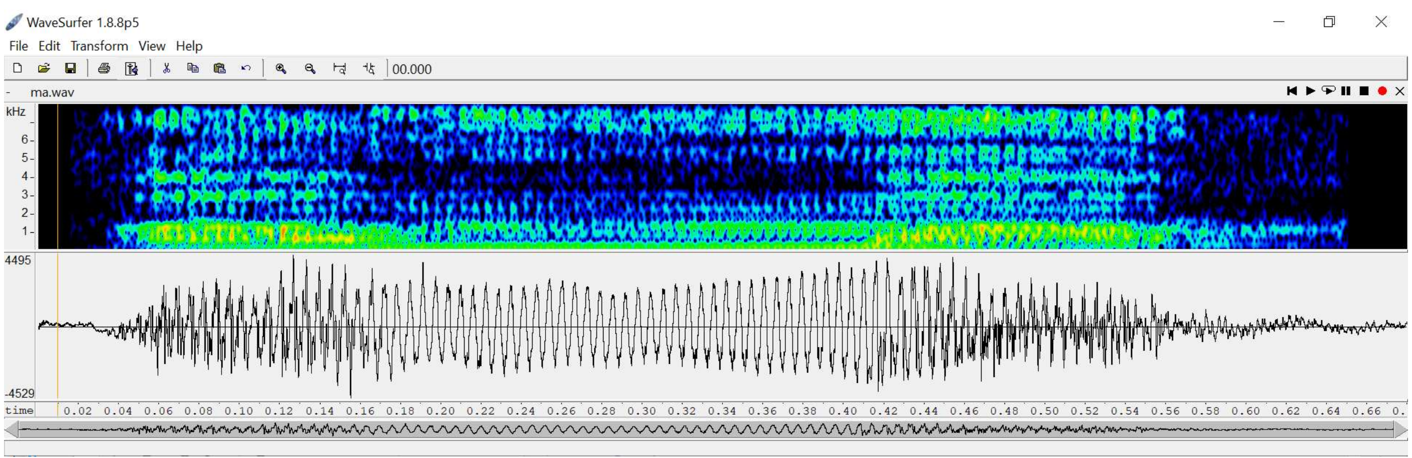
```
% /na/
subplot(311);
plot_time_signal(na_sound, fs, '"na" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(na_sound((0.5*fs):(int64((0.5+0.025)*fs))), fs, '"na" sound spectrum');
subplot(313);
spectrogram(na_sound, 128, 125, 512, fs, 'yaxis');
```






```
% /ma/
close hidden
subplot(311);
plot_time_signal(ma_sound, fs, '"ma" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(ma_sound((0.24*fs):(int64((0.24+0.025)*fs))), fs, '"ma" sound spectrum');
subplot(313);
spectrogram(ma_sound, 128, 125, 512, fs, 'yaxis');
```





Observations:

We observe anti-resonances (from the spectrogram) taking place for nasal sounds, meaning some part of the sound produced is getting absorbed along the path of air flow. Also we see that the waveform is periodic, implying that it is a voiced sound. The sound produced is due to the nasal cavity mainly. They are similar to vowels but these sounds have lesser formant energy as compared to vowels.

D.

Semi-Vowels

1. Record the sounds of any two semi-vowels and plot their time domain waveform, the magnitude spectrum and the spectrogram.
2. Inspect the above plots and write your observations. Comment on how these vary from the vowel sounds.

Theory:

Semi-vowels are similar to vowels but there is some obstruction in the passage of air flow. These are somewhat periodic and have lower energy as compared to vowels. There are 4 semi-vowels : y, r, l, w.

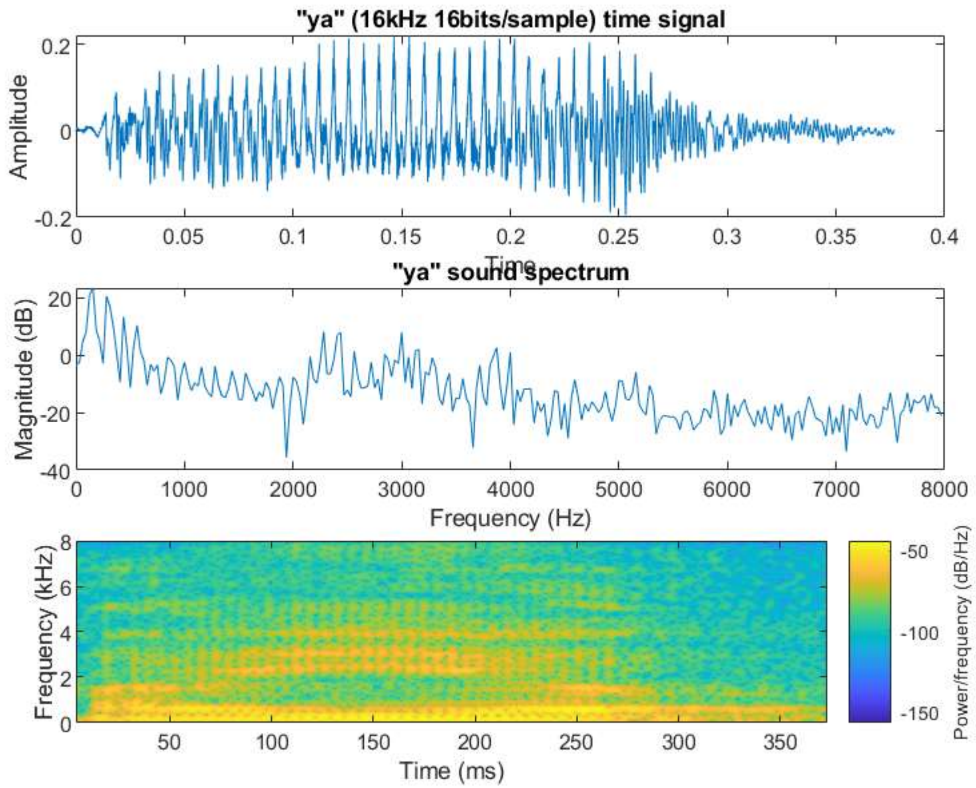
Procedure:

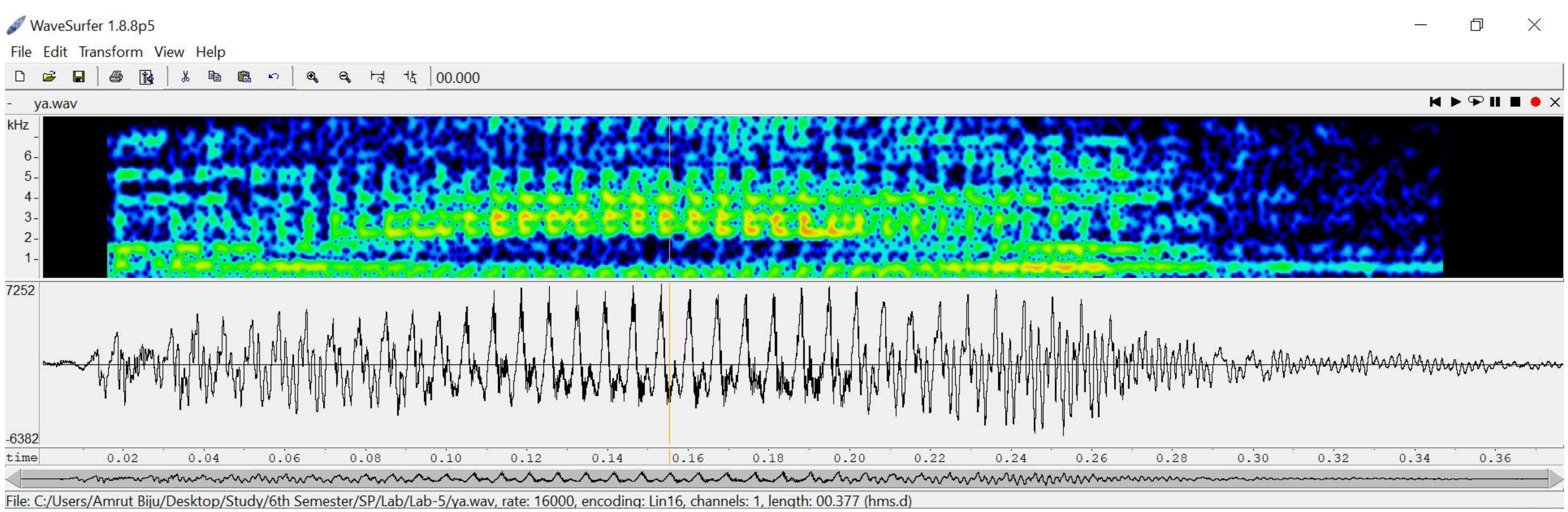
We are analysing the semi-vowel sounds "ya" and "ra". We record these sounds via the software *Audacity,* with 16 bits/sample and 16 kHz sampling frequency. We then plot time domain signal using user-defined function *plot_time_signal()* and frequency spectrum using user-defined function *plot_freq_spectrum().* We use inbuilt matlab function *spectrogram()* to plot the spectrogram. We then analyse the plots and make some observations.

Code and Plots:

```
close all; clc; clear all;
close hidden

[ya_sound,fs]=audioread('ya.wav');

[ra_sound,fs]=audioread('ra.wav');
```

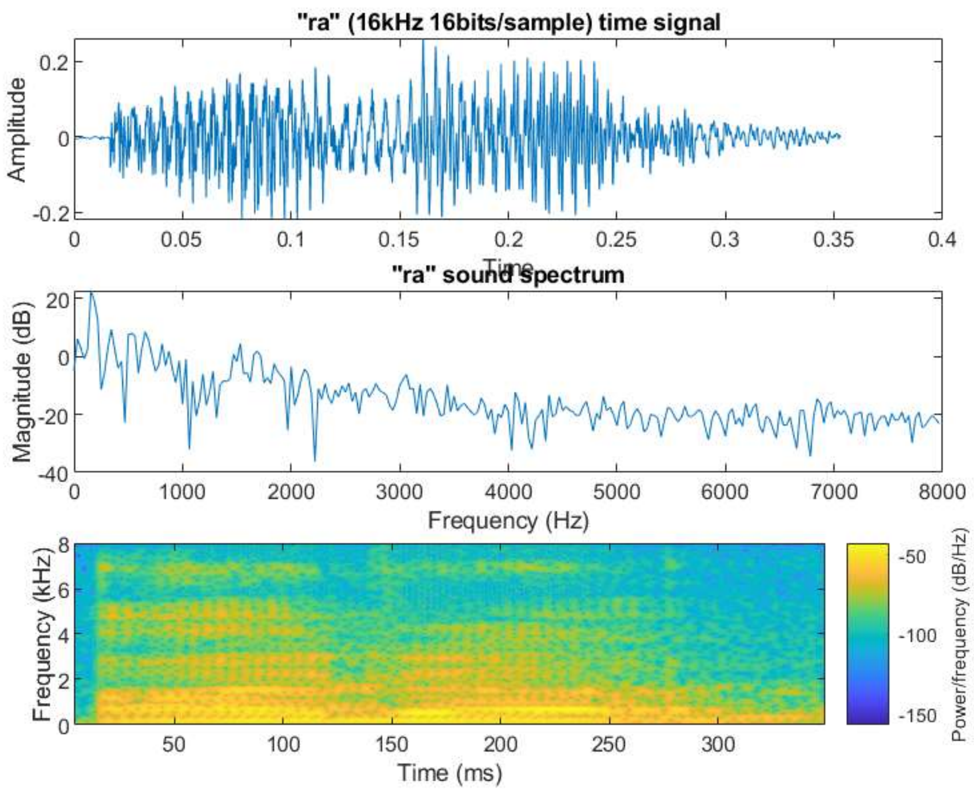
```
% /ya/
subplot(311);
plot_time_signal(ya_sound, fs, '"ya" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(ya_sound((0.16*fs):(int64((0.16+0.025)*fs))), fs, '"ya" sound spectrum');
subplot(313);
spectrogram(ya_sound, 128, 125, 512, fs, 'yaxis');
```

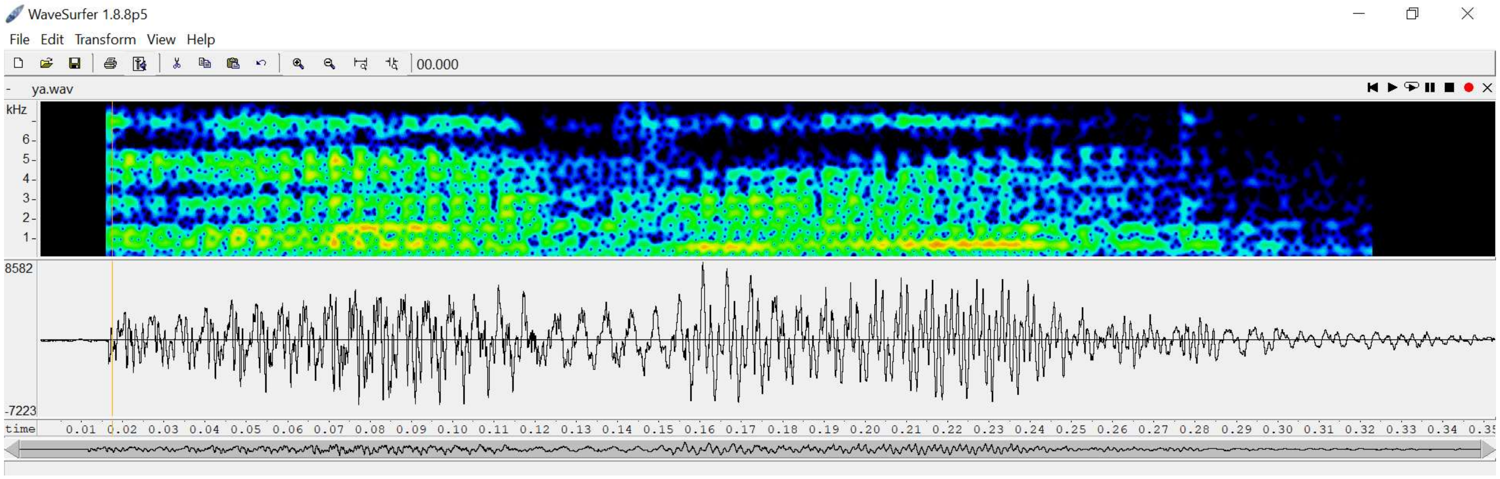

"ya" (16kHz 16bits/sample) time signal
Amplitude
Time
"ya" sound spectrum
Magnitude (dB)
Frequency (Hz)
Frequency (kHz)
Time (ms)
Power/frequency (dB/Hz)


WaveSurfer 1.8.8p5
File Edit Transform View Help
00.000
ya.wav
kHz
time
File: C:/Users/Amrut Biju/Desktop/Study/6th Semester/SP/Lab/Lab-5/ya.wav, rate: 16000, encoding: Lin16, channels: 1, length: 00.377 (hms.d)

```
% /ra/
```

```
close hidden
subplot(311);
plot_time_signal(ra_sound, fs, '"ra" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(ra_sound((0.13*fs):(int64((0.13+0.025)*fs))), fs, '"ra" sound spectrum');
subplot(313);
spectrogram(ra_sound, 128, 125, 512, fs, 'yaxis');
```





Observations:

We observe from the waveforms that these are somewhat periodic. Also we observe lesser energy as compared to vowels (from the spectrogram).

/ya/ : we see that amplitude is larger than /a/, there are higher frequencies observed.

/ra/ : we see that amplitude is smaller than /a/, there are lower frequencies observed.

These both sounds are aspirated, whereas /y/ and /l/ are unaspirated.

E.

Fricatives

1. Pick up any two fricatives having different positions of constrictions. Record these sounds and plot the time-domain waveform, the magnitude spectrum and the spectrogram.
2. Inspect the above plots and write your observations.

Theory:

Fricatives are sounds that are produced due to a narrow constriction in the passage of air flow (through this narrow constriction, air rushes out).

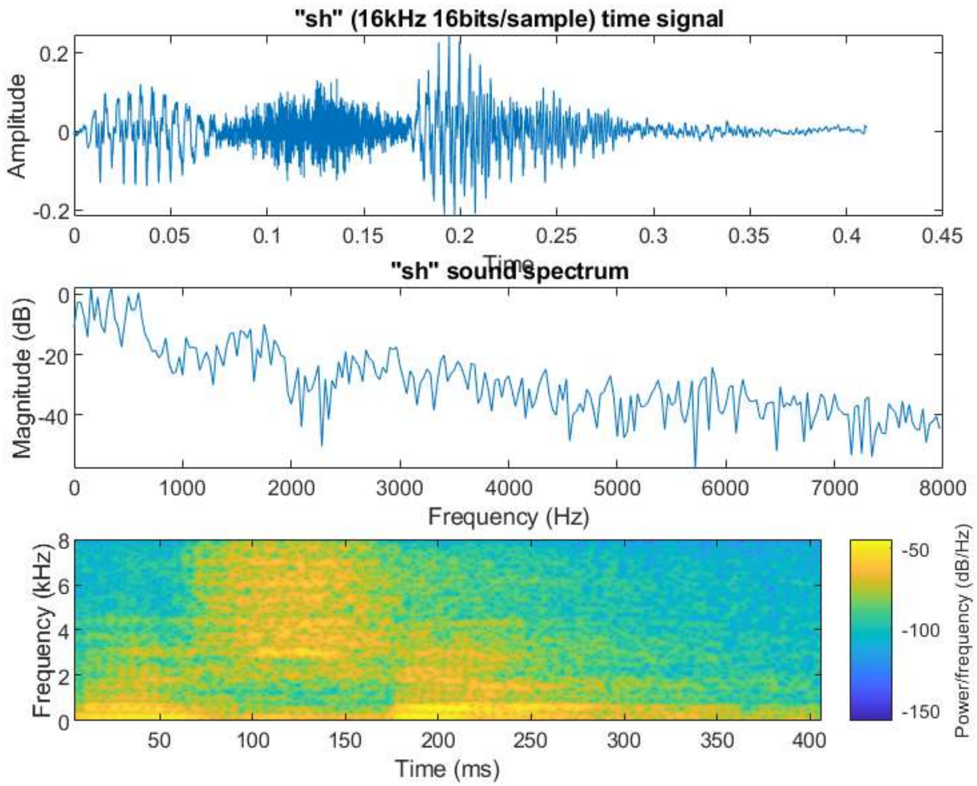
Procedure:

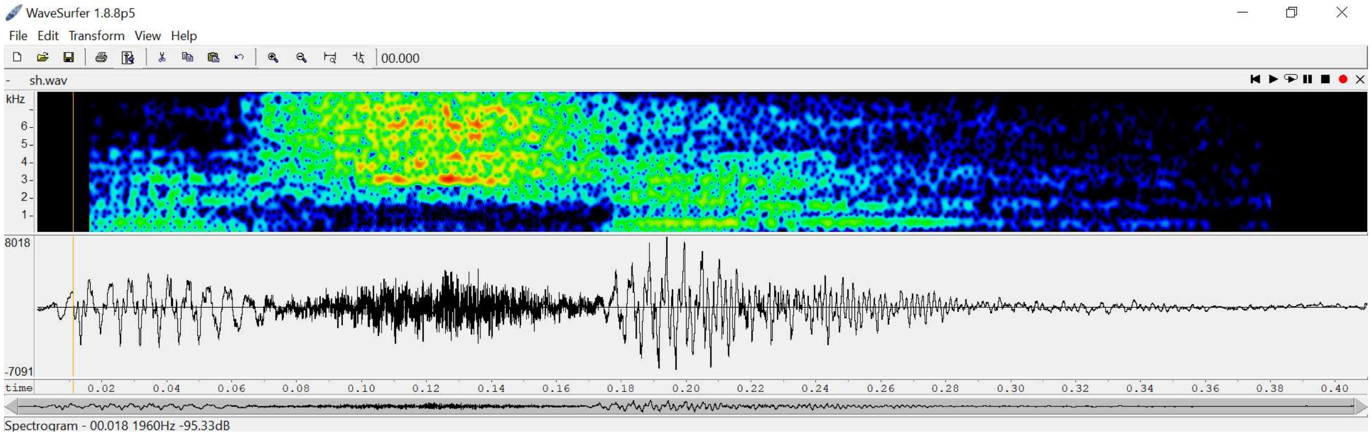
We are analysing the fricative sounds "sh" and "s". We record these sounds via the software *Audacity,* with 16 bits/sample and 16 kHz sampling frequency. We then plot time domain signal using user-defined function *plot_time_signal()* and frequency spectrum using user-defined function *plot_freq_spectrum().* We use inbuilt matlab function *spectrogram()* to plot the spectrogram. We then analyse the plots and make some observations.

Code and Plots:

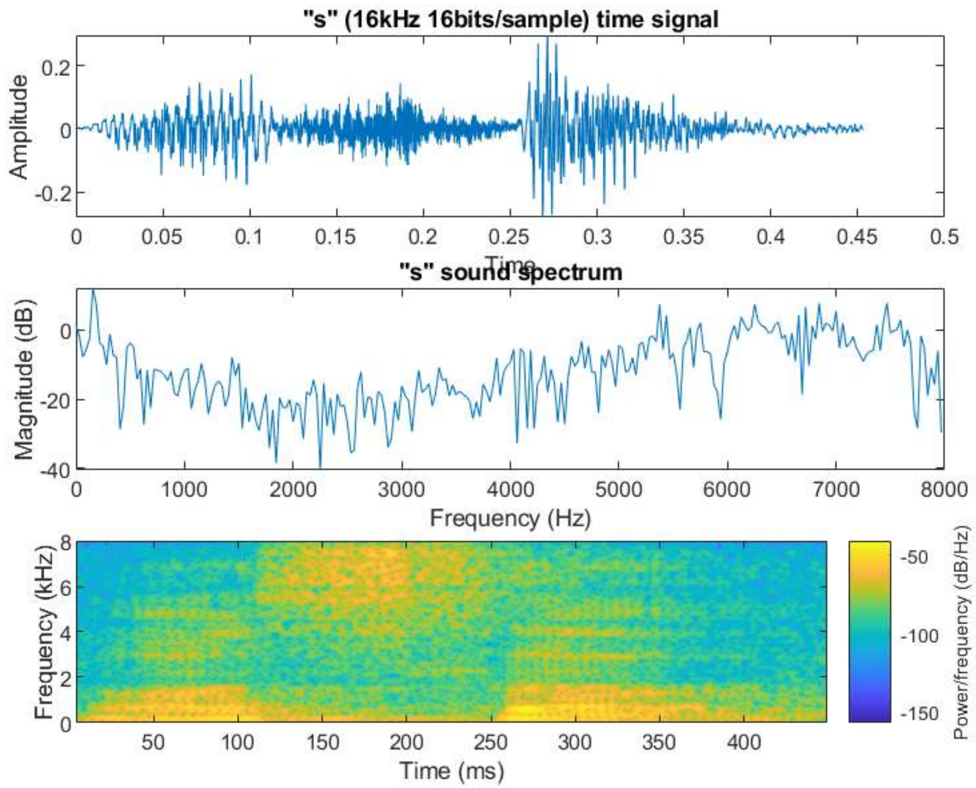
```
close all; clc; clear all;
close hidden

[sh_sound,fs]=audioread('sh.wav');

[s_sound,fs]=audioread('s.wav');
```

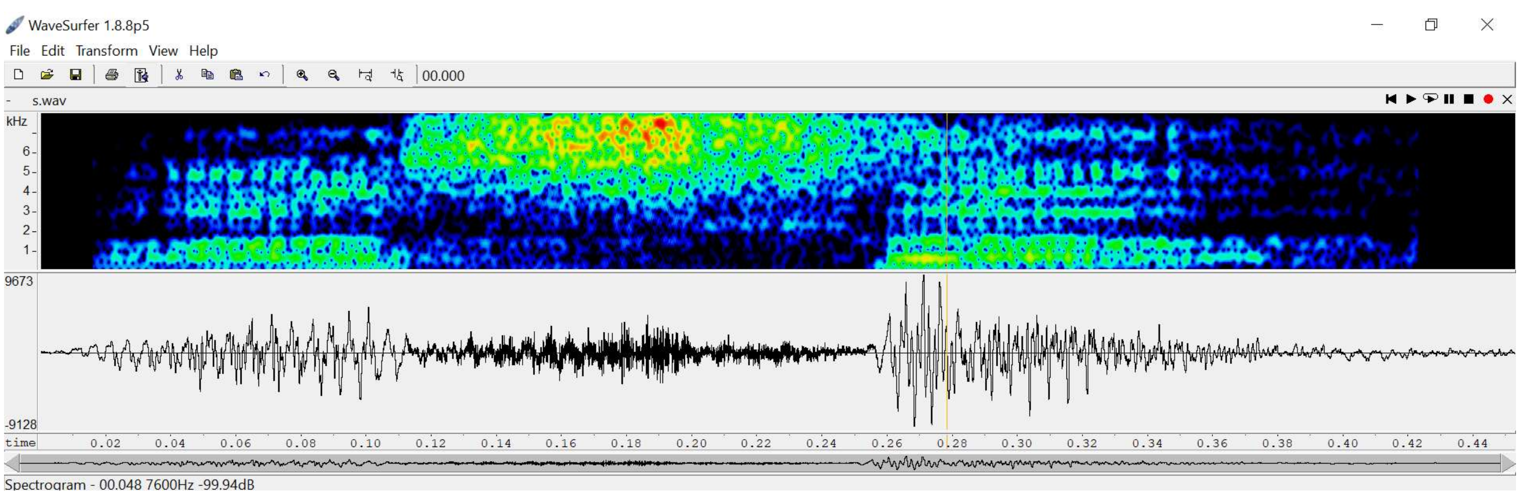
```
% /sh/
subplot(311);
plot_time_signal(sh_sound, fs, '"sh" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(sh_sound((0.3*fs):(int64((0.3+0.025)*fs))), fs, '"sh" sound spectrum');
subplot(313);
spectrogram(sh_sound, 128, 125, 512, fs, 'yaxis');
```






```
% /s/
close hidden
subplot(311);
plot_time_signal(s_sound, fs, '"s" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(s_sound((0.15*fs):(int64((0.15+0.025)*fs))), fs, '"s" sound spectrum');
subplot(313);
spectrogram(s_sound, 128, 125, 512, fs, 'yaxis');
```





Observations:

We see that fricatives occupy only large frequencies with considerable energy (from spectrogram). We also see that these are aperiodic in nature (from waveform). We see that amplitude is lesser than /a/ (voiced) (from waveform).

F.

Affricates

1. Record any one affricate sound and plot the time domain waveform, the magnitude spectrum and the spectrogram.
2. Inspect the plots and write down your observations.

Theory:

Affricates involve closure followed by frication. These are present in 2nd row of the 5x5 matrix of the stop consonants. The affricates are : c, ch, j, jh, nj. Place of articulation for these sounds are in the palatal region. 1st there is complete stoppage of air, which is then partially released to provide a narrow constriction for frication.

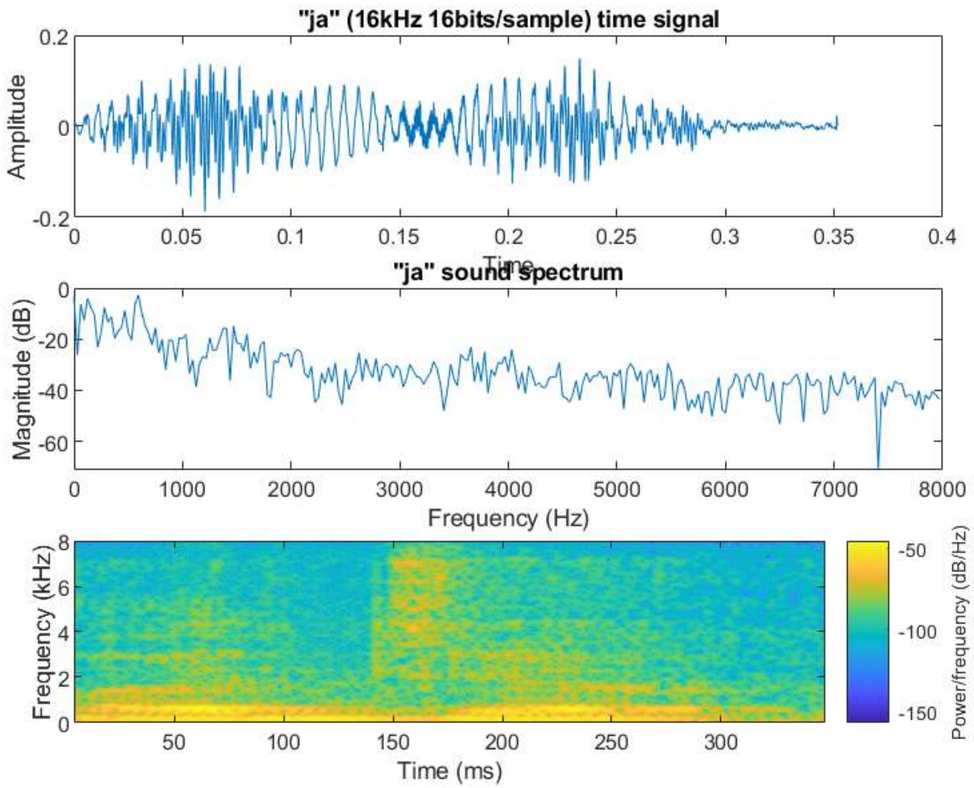
Procedure:

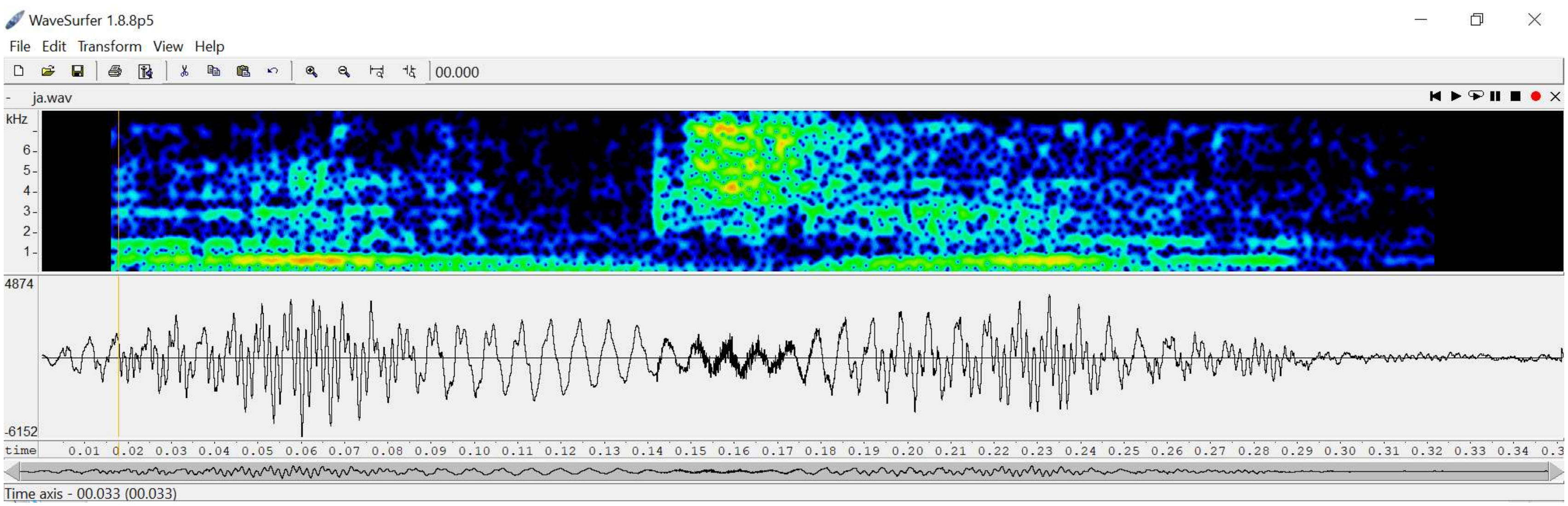
We are analysing one affricate sound "Ja". We record this sound via the software *Audacity,* with 16 bits/sample and 16 kHz sampling frequency. We then plot time domain signal using user-defined function *plot_time_signal()* and frequency spectrum using user-defined function *plot_freq_spectrum().* We use inbuilt matlab function *spectrogram()* to plot the spectrogram. We then analyse the plots and make some observations.

Code and Plots:

```
close all; clc; clear all;
close hidden

[ja_sound,fs]=audioread('ja.wav');
```

```
% /ja/
subplot(311);
plot_time_signal(ja_sound, fs, '"ja" (16kHz 16bits/sample) time signal');
subplot(312);
plot_freq_spectrum(ja_sound((0.3*fs):(int64((0.3+0.025)*fs))), fs, '"ja" sound spectrum');
subplot(313);
spectrogram(ja_sound, 128, 125, 512, fs, 'yaxis');
```


"ja" (16kHz 16bits/sample) time signal
Amplitude
Time
"ja" sound spectrum
Magnitude (dB)
Frequency (Hz)
Frequency (kHz)
Time (ms)
Power/frequency (dB/Hz)


WaveSurfer 1.8.8p5
File Edit Transform View Help
00.000
ja.wav
kHz
4874
-6152
time
Time axis - 00.033 (00.033)

Observations:

We see that there is a very small stoppage and then frication occurs (high frequency regions are occupied) (seen from spectrogram). We see 2 events happening i.e. closure and frication. We see for "ja", the amplitude is smaller than the vowel "a" amplitude (voiced, vowel).

Function Definitions

```
function plot_time_signal(y, fs, title_name)  % PLOT TIME DOMAIN PLOT FOR ANY SIGNAL
    plot((0:length(y)-1)/fs,y);
    xlabel("Time");
    ylabel("Amplitude");
    title(title_name);
end

% PLOT FREQUENCY SPECTRUM FOR ANY SIGNAL
function plot_freq_spectrum(x, fs, title_name)
    N = length(x);
    n = pow2(nextpow2(N)); % Transforming the length so that the number of samples is a power of 2.
    f = fs*[0: n-1]/n;
    X_mags = 20*log10(abs(fft(x, n)));
    N_2 = ceil(n/2);
    plot(f(1:N_2), (X_mags(1:N_2)));
    xlabel('Frequency (Hz)')
    ylabel('Magnitude (dB)');
    title(title_name);
end
```