

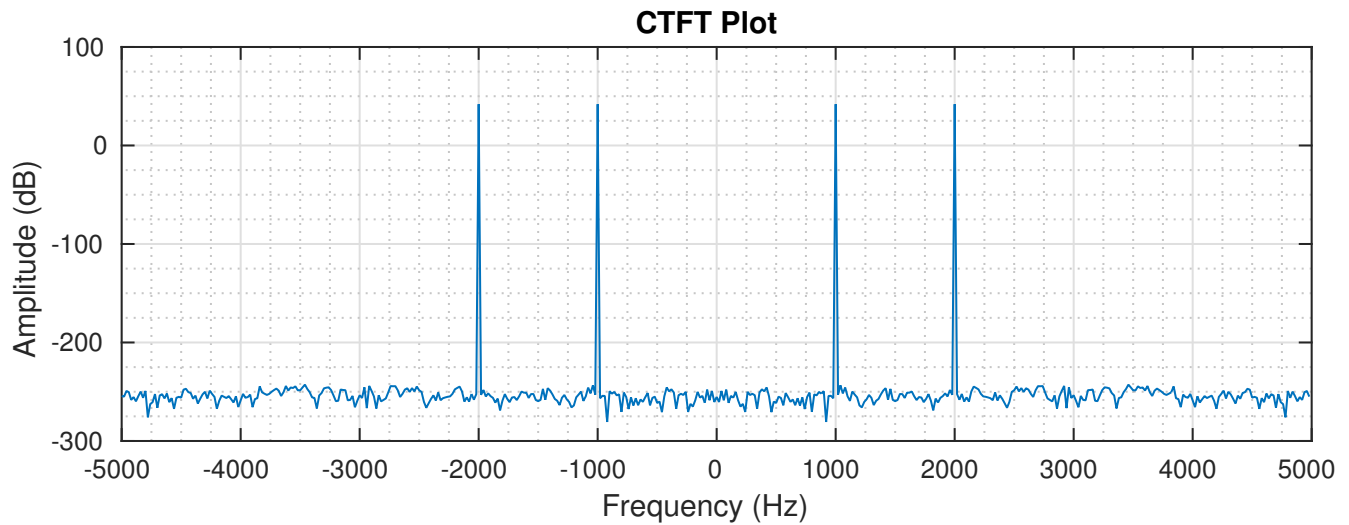
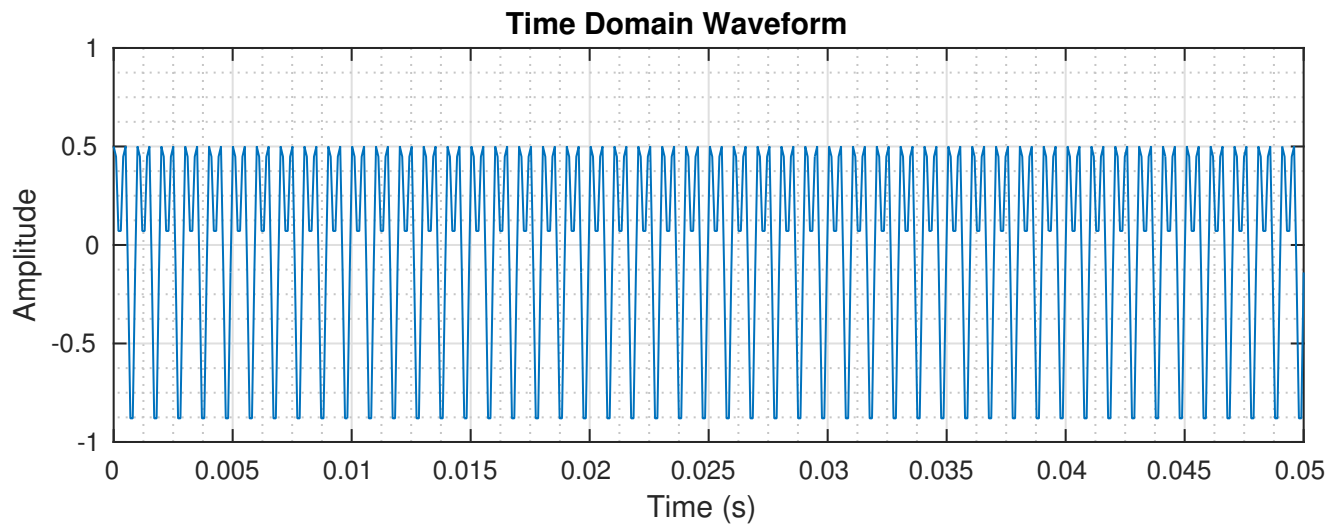
ECEN455 Lab1

Divyansha Pandey
UIN : 623007531

Task 1

Input Signal: $0.5 * \sin(2\pi(1000)t) + 0.5 * \cos(2\pi(2000)t)$

Sampling Rate: 10 kHz



Matlab code:

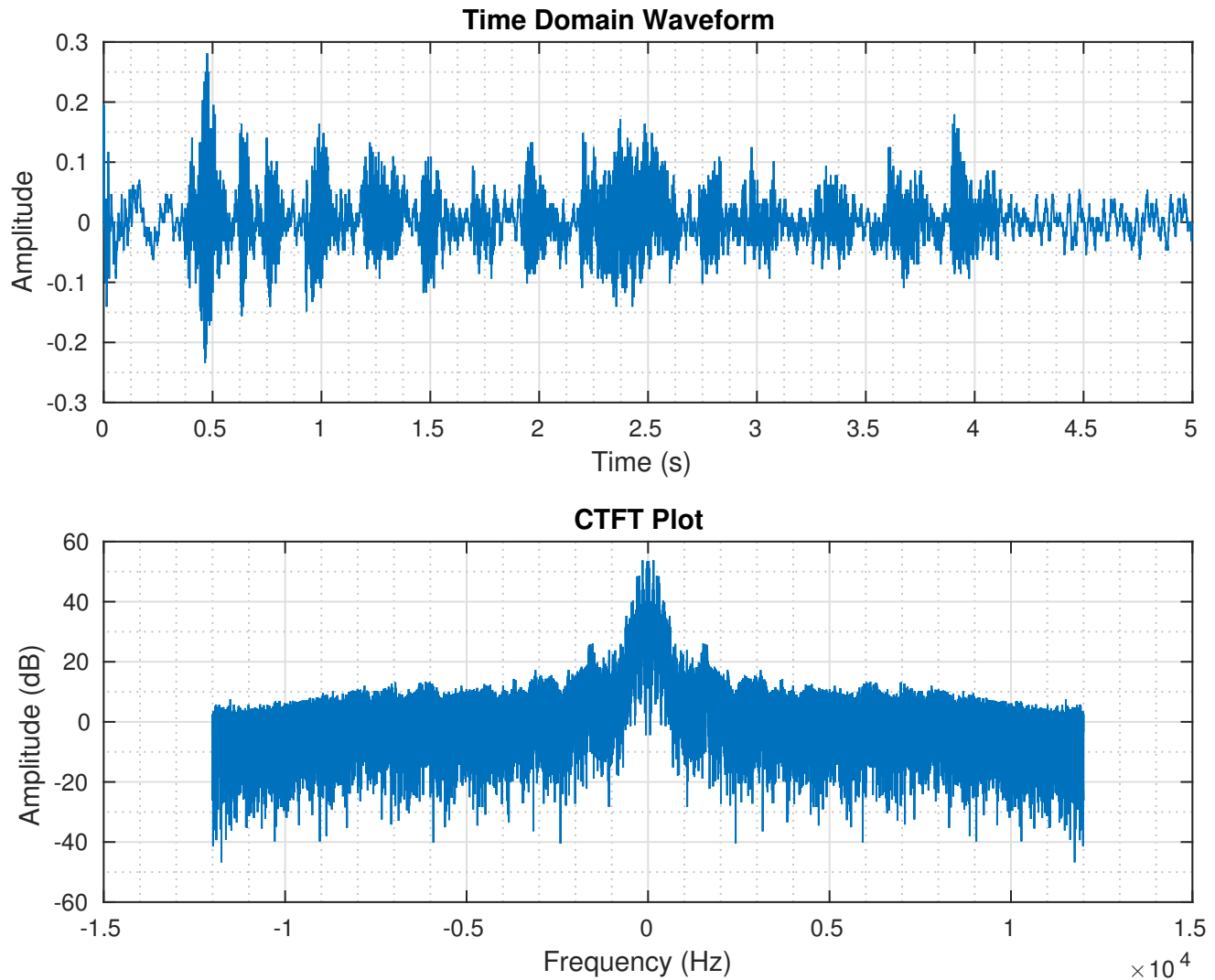
```
1 clc; clear; close all;
2 f_sig = 1000; % Tone frequency
3 fs = 10000; % Sampling frequency
4 Ts = 1/fs;
5 samp_time = 0.05; % sample 50ms of the waveform
6 for k = 1:samp_time/Ts
7     x(k) = 0.5*sin(2*pi*(k-1)*f_sig*Ts) + 0.5*cos(2*pi*(k-1)*(2*f_sig)*Ts);
8 end
9 [Xf,f] = CTFT(x,fs);
10 subplot(2,1,1)
11 plot(linspace(0,samp_time,samp_time*fs),x)
12 title('Time Domain Waveform')
13 xlabel('Time (s)')
14 ylabel('Amplitude')
15 grid on
16 grid minor
17 subplot(2,1,2)
18 plot(f,20*log10(abs(Xf)))
19 title('CTFT Plot')
20 xlabel('Frequency (Hz)')
21 ylabel('Amplitude (dB)')
22 grid on
23 grid minor
24 print('Task1','-depsc')
```

CTFT.m:

```
1 function [ Xf, f ] = CTFT( x,fs )
2 % Takes the time domain signal x sampled at fs and returns the CTFT(approximation/sampled ...
   version) signal.
3 Xf = fftshift(fft(x));
4 N = length(Xf);
5 f = linspace(-N/2,N/2 - 1, N) *fs/N;
6 end
```

Task 2

Following is the time and frequency domain plot of the raw recording.

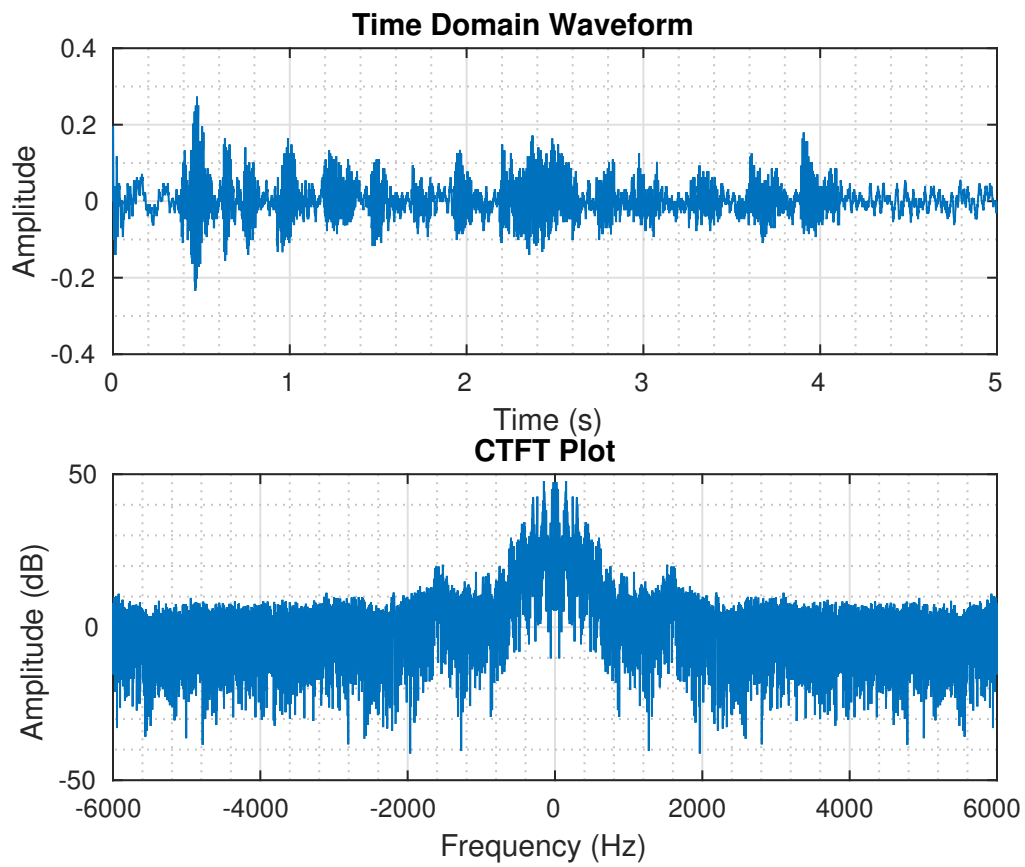


The frequency domain plot shows that for frequencies greater than 2500 Hz have very low energy and most of the information is present within 2500 Hz.

Downsampling the recording by a factor of 2 removes the high frequency components(i.e. $> 6kHz$) and we get a zoomed version of the frequency domain.

The downsampling(by 2) has no significant impact on the voice recording.

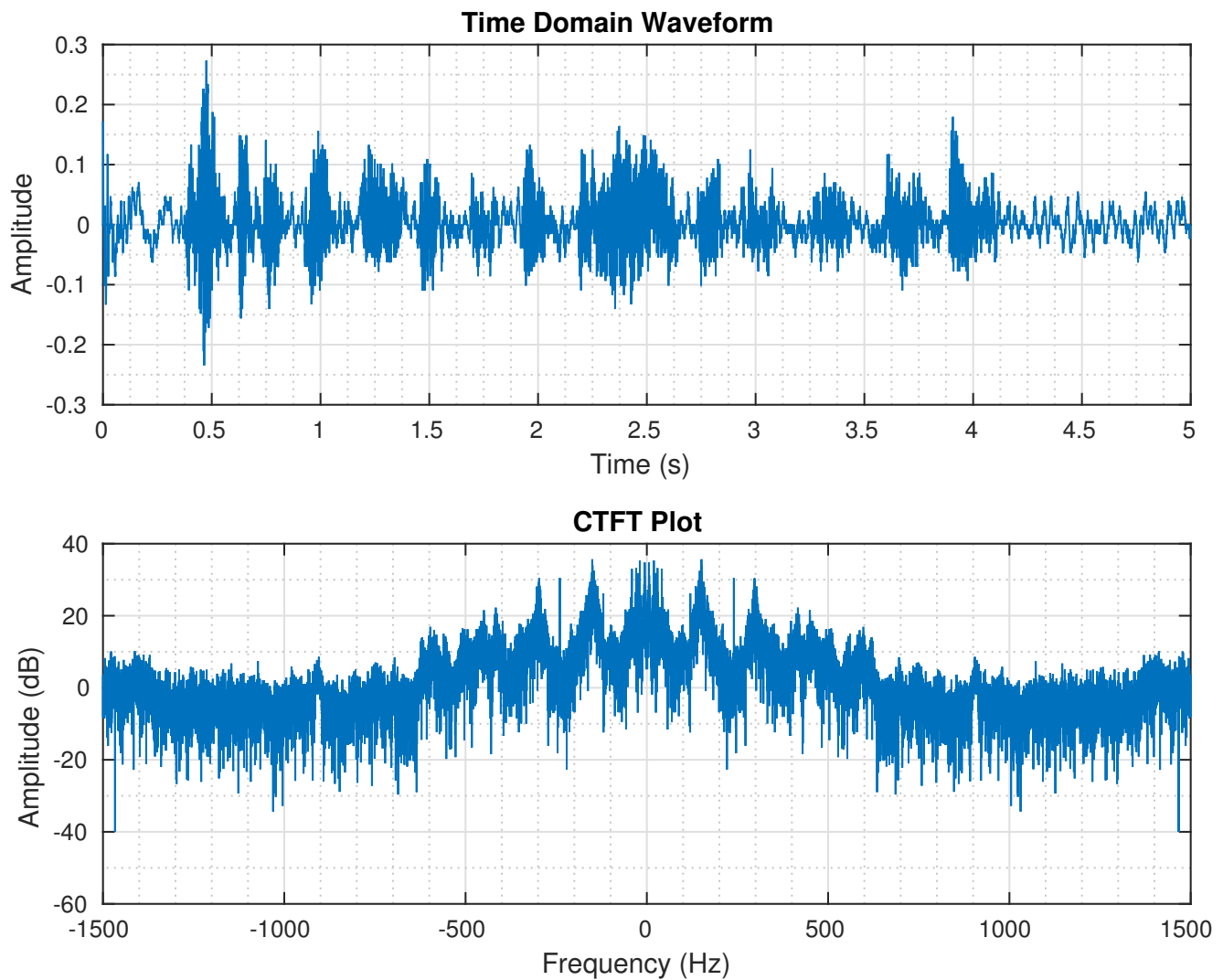
This plot shows the downsampled time and frequency domain of the signal:



While downsampling by 8 we loose information in the frequencies $> 1.5kHz$ and we get a further zoomed version of the frequency domain.

This (downsampling by 8) has noticable impact on the recording however, the recording is still perceivable.

The plot shows the downsampled by 8 time and frequency domain of the signal:



Matlab code:

```
1 clc; clear; close all;
2 load('recording.mat');
3 x = getaudiodata(voice.sample);
4 fs = 24000;
5 Ts = 1/fs;
6 samp_time = 5;
7 [Xf,f] = CTFT(x,fs);
8 figure;
9 subplot(2,1,1)
10 plot(linspace(0,samp_time,samp_time/Ts),x)
11 title('Time Domain Waveform')
12 xlabel('Time (s)')
13 ylabel('Amplitude')
14 grid on
15 grid minor
16 subplot(2,1,2)
17 plot(f,20*log10(abs(Xf)))
18 title('CTFT Plot')
```

```

19 xlabel('Frequency (Hz)')
20 ylabel('Amplitude (dB)')
21 grid on
22 grid minor
23 print('Task2.original','-depsc')
24
25 % Downsample by 2
26 DS_factor = 2;
27 N = length(x);
28 for i = 1:N/DS_factor
29     x_ds(i) = x(DS_factor * i);
30 end
31 [Xf_ds,f_ds] = CTFT(x_ds,fs/DS_factor);
32 figure;
33 subplot(2,1,1)
34 plot(linspace(0,samp_time,samp_time/(DS_factor*Ts)),x_ds)
35 title('Time Domain Waveform')
36 xlabel('Time (s)')
37 ylabel('Amplitude')
38 grid on
39 grid minor
40 subplot(2,1,2)
41 plot(f_ds,20*log10(abs(Xf_ds)))
42 title('CTFT Plot')
43 xlabel('Frequency (Hz)')
44 ylabel('Amplitude (dB)')
45 grid on
46 grid minor
47 print('Task2.DS2','-depsc')
48
49 % Downsample by 8
50 DS_factor = 8;
51 N = length(x);
52 for i = 1:N/DS_factor
53     x_ds8(i) = x(DS_factor * i);
54 end
55 [Xf_ds8,f_ds8] = CTFT(x_ds8,fs/DS_factor);
56 figure;
57 subplot(2,1,1)
58 plot(linspace(0,samp_time,samp_time/(DS_factor*Ts)),x_ds8)
59 title('Time Domain Waveform')
60 xlabel('Time (s)')
61 ylabel('Amplitude')
62 grid on
63 grid minor
64 subplot(2,1,2)
65 plot(f_ds8,20*log10(abs(Xf_ds8)))
66 title('CTFT Plot')
67 xlabel('Frequency (Hz)')
68 ylabel('Amplitude (dB)')
69 grid on
70 grid minor
71 print('Task2.DS8','-depsc')

```