MIDDLE EAST TECHNICAL UNIVERSITY

ELECTRICAL-ELECTRONICS ENGINEERING DEPARTMENT

EE-301 Term Project

Analyzing Spectral Content of a Tone

Mustafa Barış Emektar
2304533

The sampling theorem implies that a continuous-time signal can be fully expressed at points evenly spaced in time by its sample values under some conditions. How many samples a digital audio device uses for the recording of the audio signal per second is determined by the sample rate. In other words the higher the sampling rate, the higher frequencies a device can record.

PART 1

1. In the first part, a tone is recorded produced by the piano with different sampling rates such as F_{S1}=44100 Hz, F_{S2}=11025 Hz, F_{S3}=4900 Hz, F_{S4}=2756 Hz. Then, the frequency spectrum differences between these recordings observed.

```
x1 = audiorecorder(44100,16,1);
pause(2);
disp('Play!')
recordblocking(x1, 3);
disp('Stop!')
x1.play;
%%
x2 = audiorecorder(11025,16,1);
pause(2);
disp('Play!')
recordblocking(x2, 3);
disp('Stop!')
x2.play;
%%
x3 = audiorecorder(4900,16,1);
pause(2);
disp('Play!')
recordblocking(x3, 3);
disp('Stop!')
x3.play;
%%
x4 = audiorecorder(2756,16,1);
pause(2);
disp('Play!')
recordblocking(x4, 3);
disp('Stop!')
x4.play;
%%
x1.play; pause(3);
x2.play; pause(3);
x3.play; pause(3);
x4.play; pause(3);
```

Comment: It is quite obvious that, the audio whose sampling rate is higher (44000 Hz), sounds better. So it can be said that, as the sampling rate increases the higher frequencies can be recorded and vice versa.

2.

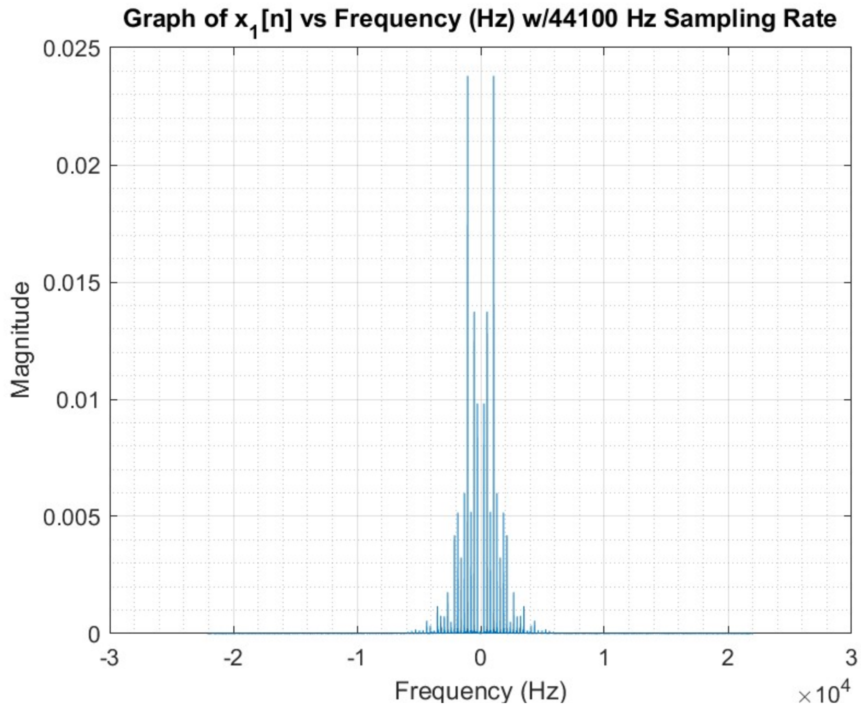

Figure 1: Graph of x_1[n] vs Frequency (Hz) w/44100 Hz Sampling Rate

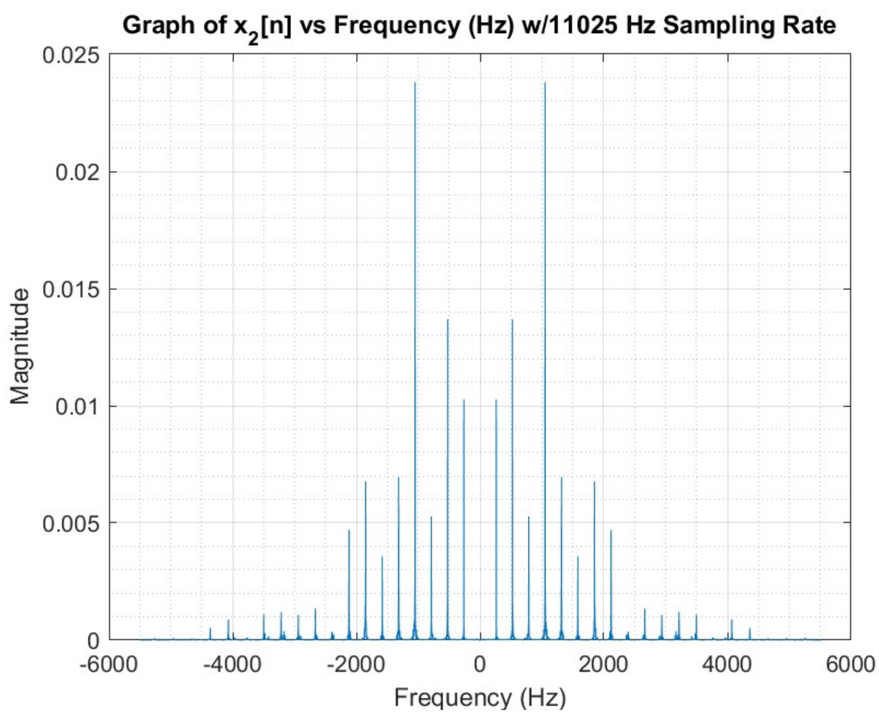

Figure 2: Graph of x_2[n] vs Frequency (Hz) w/11025 Hz Sampling Rate

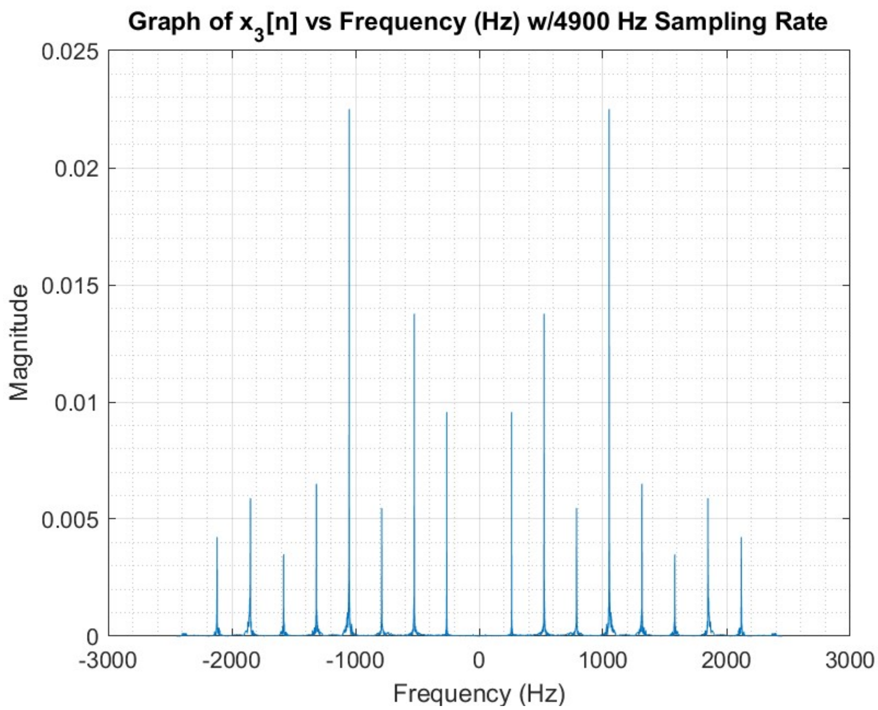

Figure 3: Graph of $x_3[n]$ vs Frequency (Hz) w/4900 Hz Sampling Rate

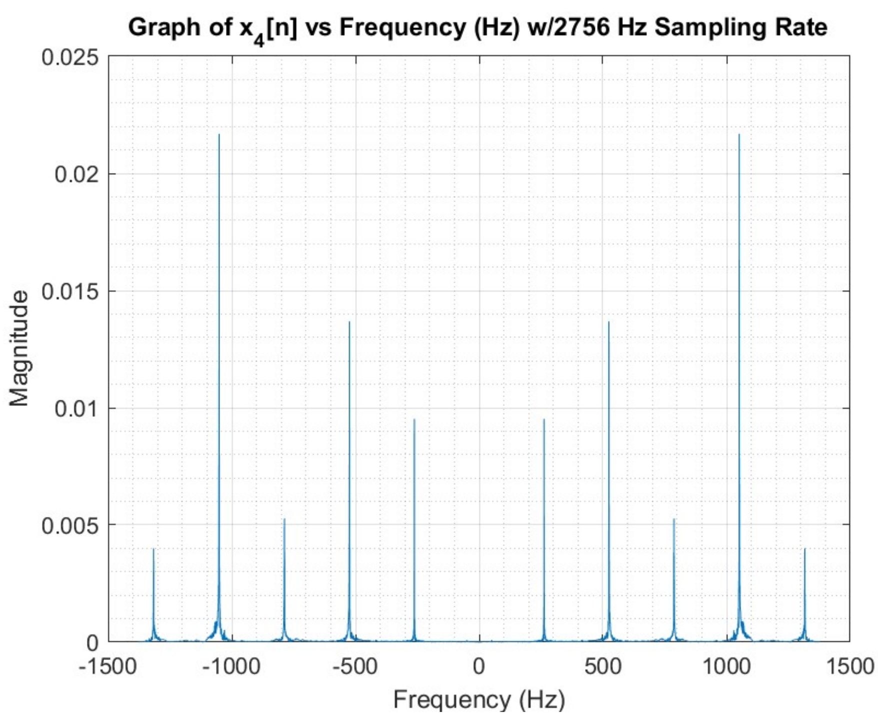

Figure 4: Graph of $x_4[n]$ vs Frequency (Hz) w/2756 Hz Sampling Rate

```
x1_data = getaudiodata(x1);
x2_data = getaudiodata(x2);
x3_data = getaudiodata(x3);
x4_data = getaudiodata(x4);

x1_dft = fftshift(fft(x1_data));
x2_dft = fftshift(fft(x2_data));
x3_dft = fftshift(fft(x3_data));
x4_dft = fftshift(fft(x4_data));

x1_mag = abs(x1_dft)/length(abs(x1_dft));  %normalized
x2_mag = abs(x2_dft)/length(abs(x2_dft));  %normalized
x3_mag = abs(x3_dft)/length(abs(x3_dft));  %normalized
x4_mag = abs(x4_dft)/length(abs(x4_dft));  %normalized

f1 = linspace(-44100/2,44100/2, length(x1_data));
f2 = linspace(-11025/2,11025/2, length(x2_data));
f3 = linspace(-4900/2,4900/2, length(x3_data));
f4 = linspace(-2756/2,2756/2, length(x4_data));


plot(f1',x1_mag); grid minor; grid on;
xlabel('Frequency (Hz)'); ylabel('Magnitude'); title('Graph of x_1[n]...
vs Frequency (Hz) w/44100 Hz Sampling Rate')
saveas(gcf,'x1.png')
figure; plot(f2',x2_mag); grid minor; grid on;
xlabel('Frequency (Hz)'); ylabel('Magnitude'); title('Graph of x_2[n]...
vs Frequency (Hz) w/11025 Hz Sampling Rate')
saveas(gcf,'x2.png')
figure; plot(f3',x3_mag); grid minor; grid on;
xlabel('Frequency (Hz)'); ylabel('Magnitude'); title('Graph of x_3[n]...
vs Frequency (Hz) w/4900 Hz Sampling Rate')
saveas(gcf,'x3.png')
figure; plot(f4',x4_mag); grid minor; grid on;
xlabel('Frequency (Hz)'); ylabel('Magnitude'); title('Graph of x_4[n]...
vs Frequency (Hz) w/2756 Hz Sampling Rate')
saveas(gcf,'x4.png')

sound(x1_data,44100)
pause(3);
sound(x2_data,11025)
pause(3);
sound(x3_data,4900)
pause(3);
sound(x4_data,2756)
```

PART 2

In the second part of the project, Discrete Time Fourier Transform (DTFT) is applied to the previous signals. It is observed that, as the sampling rate of the recording increases, DTFT of the signal's magnitude for bigger frequency values increases and vice versa. Also notice that it is much more appropriate to use 44100Hz sampling rate because the maximum frequency of the sound is 20kHz. So there will be no aliasing if the tone is is recorded with 44100Hz sampling rate. After observing the graphs, $x_1[n]$ is downsampled by 16. So $x_5[n] = x_1[16n]$

3.

Comment: Designing the anti-aliasing filter with the following values are more appropriate values for us to filter aliasing. The values determined by the Nyquist Theorem(which states that a periodic signal must be sampled at more than twice the highest frequency component of the signal.) and the filter is exported as h[n]. Finally when $x_1[n]$ is convolved with h[n] and obtained $x_6[n]$, it sounds much clearer than $x_5[n]$ and is much more similar to $x_4[n]$.

$$w_{pass} = 0.064 \ \& \ w_{stop} = 0.08$$

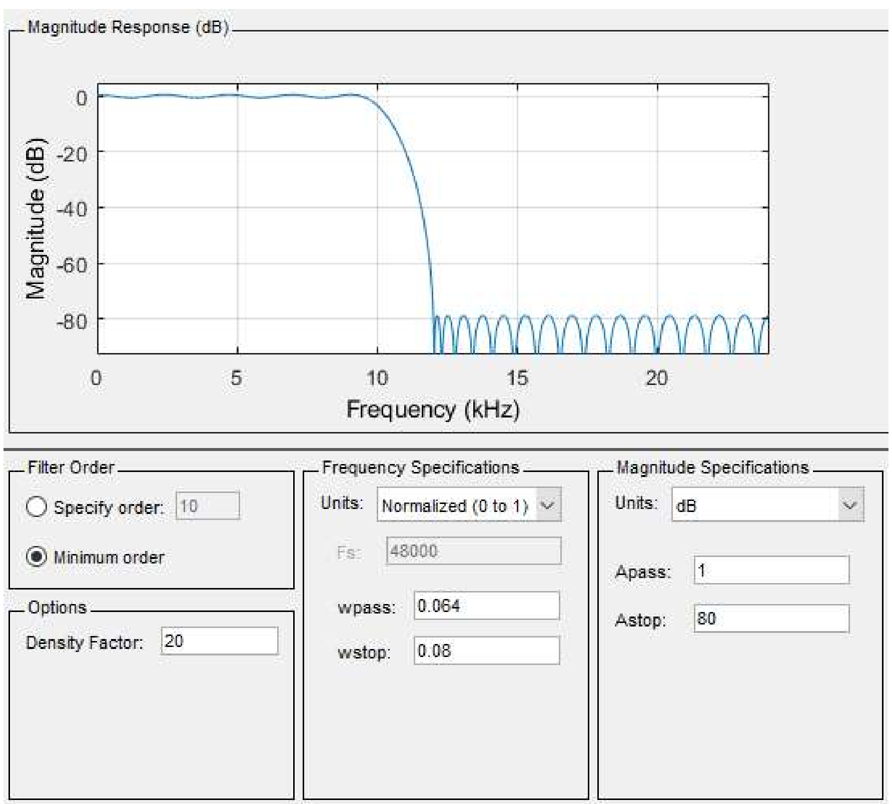

Figure 5: Frequency specifications and magnitude response of low pass filter

4.

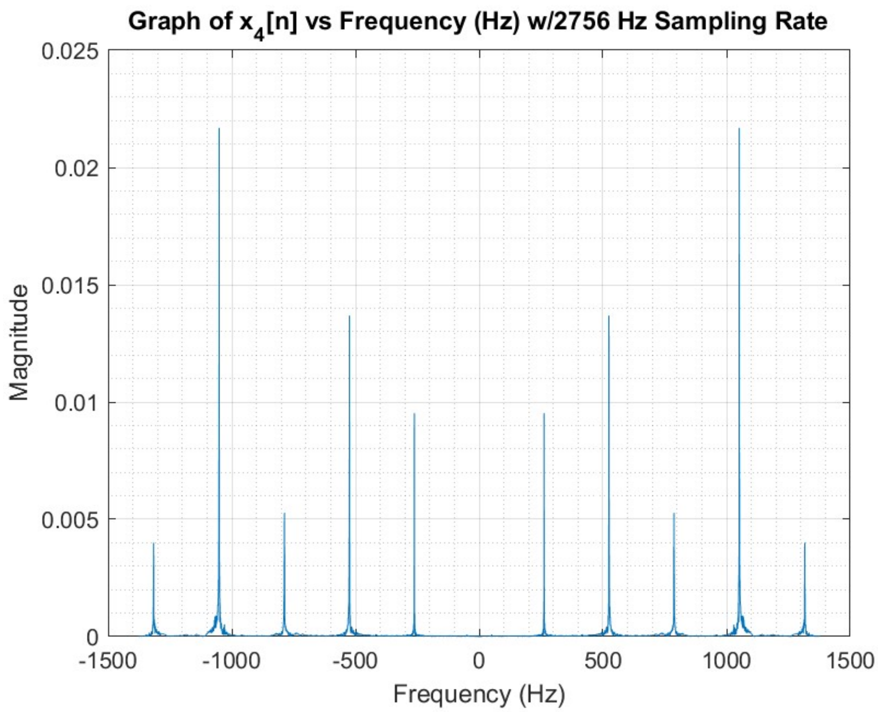

Figure 6: Graph of $x_4[n]$ vs Frequency (Hz)

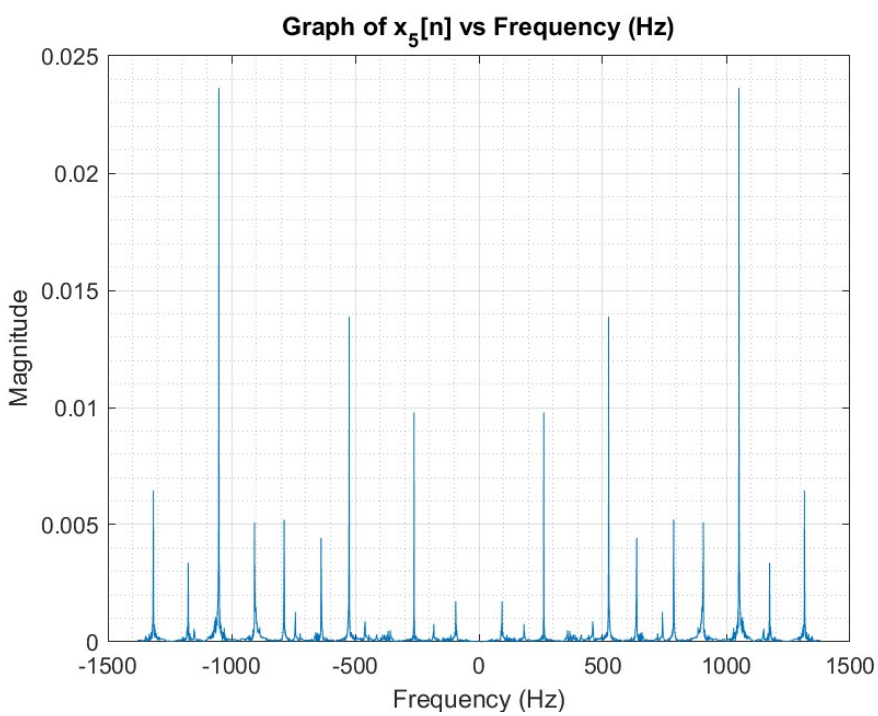

Figure 7: Graph of $x_5[n]$ vs Frequency (Hz)

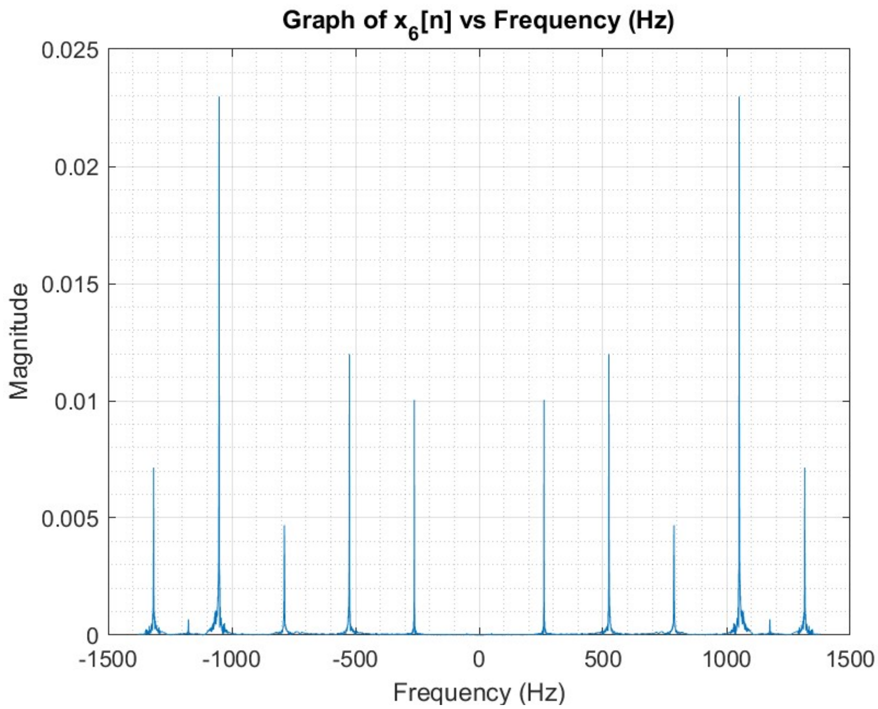

Figure 8: Graph of $x_6[n]$ vs Frequency (Hz)

In the last part, the $x_4[n]$, $x_5[n]$ and $x_6[n]$ are obtained and plotted. It is obvious that $x_6[n]$ is much more similar to $x_4[n]$ than $x_5[n]$ both from graphs and as sound. However $x_5[n]$ has less quality than others because as expected, downsampling causes the aliasing. To prevent aliasing, a lowpass filter need to be used.

```
x1_data = getaudiodata(x1);
x4_data = getaudiodata(x4);

x5_data = downsample(x1_data,16);

%w_pass = 0.064 ; w_stop = 0.08
x6_data = downsample(conv(x1_data,h),16);

x1_dft = fftshift(fft(x1_data));
x4_dft = fftshift(fft(x4_data));
x5_dft = fftshift(fft(x5_data));
x6_dft = fftshift(fft(x6_data));

x1_mag = abs(x1_dft)/length(abs(x1_dft));   %normalized
x4_mag = abs(x4_dft)/length(abs(x4_dft));   %normalized
x5_mag = abs(x5_dft)/length(abs(x5_dft));   %normalized
x6_mag = abs(x6_dft)/length(abs(x6_dft));   %normalized

f4 = linspace(-2756/2,2756/2, length(x4_mag));
f5 = linspace(-2756/2,2756/2, length(x5_mag));
f6 = linspace(-2756/2,2756/2, length(x6_mag));


plot(f4',x4_mag); grid minor; grid on;
xlabel('Frequency (Hz)'); ylabel('Magnitude'); title('Graph of x_4[n]... vs Frequency (Hz)')
saveas(gcf,'3-x4.png')
figure; plot(f5',x5_mag); grid minor; grid on;
xlabel('Frequency (Hz)'); ylabel('Magnitude'); title('Graph of x_5[n] ... vs Frequency (Hz)')
saveas(gcf,'3-x5.png')
figure; plot(f6',x6_mag); grid minor; grid on;
xlabel('Frequency (Hz)'); ylabel('Magnitude'); title('Graph of x_6[n] ... vs Frequency (Hz)')
saveas(gcf,'3-x6.png')

sound(x4_data,2756);
pause(3);
sound(x5_data,2756);
pause(3);
```