3TP3: Signals & Systems
Lab #3

Instructor: Dr. Kirubarajan

Tamer Rafidi – L02 – rafidit- 400333527
Samer Rafidi – L02 – rafidis- 400333524

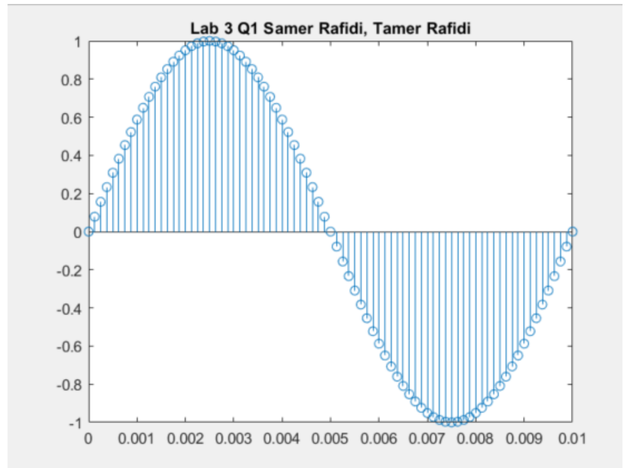
As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by **[Tamer Rafidi, rafidit, 400333527]**

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by **[Samer Rafidi, rafidis, 400333524]**

Aliasing in the Telephone System

Question 1

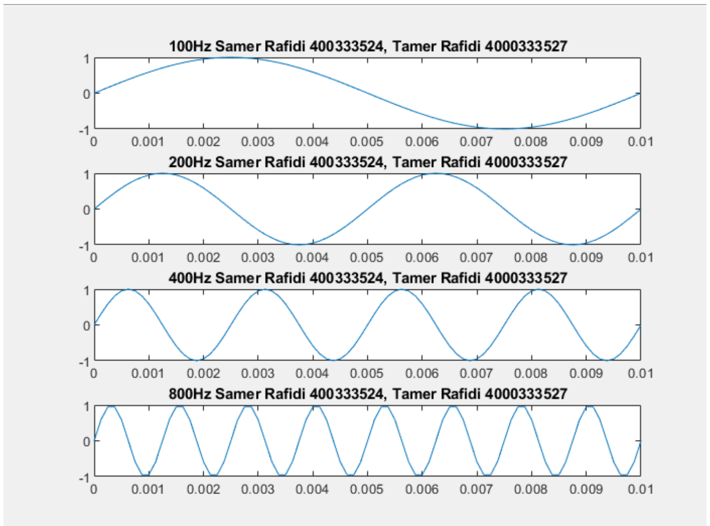
```
% Samer Rafidi 400333524, Tamer Rafidi 400333527
% Do a plot of a sampled sinusoid with frequency f = 100 Hz
%
f = 100
%
% Sampling frequency and interval
fs = 8000;
Ts = 1/fs;
%
% Set time duration of plot, i.e., 10 msec.
tfinalplot = 10e-3;
%
% Make the time vector for the plot
nplot=0:Ts:tfinalplot;
%
% Sample the sinusoid.
xnT = sin(2*pi*f*nplot);
%
% Make the plot
stem(nplot, xnT);
title('Lab 3 Q1 Samer Rafidi, Tamer Rafidi');
%
% Uncomment/edit this next line to save the graph.
exportgraphics(gcf, 'Q1_Plot.jpg');
%

```



Question 2

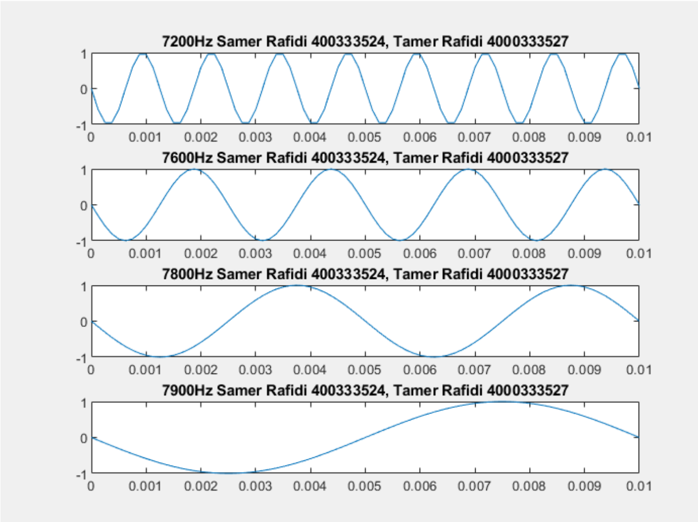
```
% Samer Rafidi 400333524, Tamer Rafidi 400333527
% Use sinusoid frequency f = 300 Hz
f1 = 100
f2 = 200
f3 = 400
f4 = 800
%
% Sampling frequency and interval
fs = 8000;
Ts = 1/fs;
%
% Set time duration of plot, i.e., 10 msec.
tfinalplot = 10e-3;
%
% Make the time vector for the plot
nplot=0:Ts:tfinalplot;
%
% Make the time vector for replayed sound spurt
% Play the spurt for 2 seconds
tfinal = 2;
nsound=0:Ts:tfinal;
%

%creating a subplot for the first graph
subplot(4,1,1)
% Sample the sinusoid.
xnT100 = sin(2*pi*f1*nsound);
plot(nplot, xnT100(1:length(nplot)));
title('100Hz Samer Rafidi 400333524, Tamer Rafidi 4000333527');

%creating a subplot for the second graph
subplot(4,1,2)
% Sample the sinusoid.
xnT200 = sin(2*pi*f2*nsound);
plot(nplot, xnT200(1:length(nplot)));
title('200Hz Samer Rafidi 400333524, Tamer Rafidi 4000333527');
```

```
%creating a subplot for the third graph
subplot(4,1,3)
% Sample the sinusoid.
xnT400 = sin(2*pi*f3*nsound);
plot(nplot, xnT400(1:length(nplot)));
title('400Hz Samer Rafidi 400333524, Tamer Rafidi 4000333527');

%creating a subplot for the fourth graph
subplot(4,1,4)
% Sample the sinusoid.
xnT800 = sin(2*pi*f4*nsound);
plot(nplot, xnT800(1:length(nplot)));
title('800Hz Samer Rafidi 400333524, Tamer Rafidi 4000333527');

xnTotal = [xnT100, xnT200, xnT400, xnT800];

% Save xnT as a wav sound file, soundfile.wav.
audiowrite(%soundfile.wavo, xnTotal, fs);
%
% Uncomment/edit this next line to save the graph.
exportgraphics(gcf, 'Q2_Graph');
%
```



Question 3

```
% Samer Rafidi 400333524, Tamer Rafidi 400333527
% Use sinusoid frequency f = 300 Hz
f1 = 7200
f2 = 7600
f3 = 7800
f4 = 7900
%
% Sampling frequency and interval
fs = 8000;
Ts = 1/fs;
%
% Set time duration of plot, i.e., 10 msec.
tfinalplot = 10e-3;
%
% Make the time vector for the plot
nplot=0:Ts:tfinalplot;
%
% Make the time vector for replayed sound spurt
% Play the spurt for 2 seconds
tfinal = 2;
nsound=0:Ts:tfinal;

%creating a subplot for the first graph
subplot(4,1,1)
% Sample the sinusoid.
xnT7200 = sin(2*pi*f1*nsound);
plot(nplot, xnT7200(1:length(nplot)));
title('7200Hz Samer Rafidi 400333524, Tamer Rafidi 4000333527');

%creating a subplot for the second graph
subplot(4,1,2)
% Sample the sinusoid.
xnT7600 = sin(2*pi*f2*nsound);
plot(nplot, xnT7600(1:length(nplot)));
title('7600Hz Samer Rafidi 400333524, Tamer Rafidi 4000333527');

%creating a subplot for the third graph
subplot(4,1,3)
% Sample the sinusoid.
xnT7800 = sin(2*pi*f3*nsound);
plot(nplot, xnT7800(1:length(nplot)));
title('7800Hz Samer Rafidi 400333524, Tamer Rafidi 4000333527');

%creating a subplot for the fourth graph
subplot(4,1,4)
% Sample the sinusoid.
xnT7900 = sin(2*pi*f4*nsound);
plot(nplot, xnT7900(1:length(nplot)));
title('7900Hz Samer Rafidi 400333524, Tamer Rafidi 4000333527');

xnTotal = [xnT7200, xnT7600, xnT7800, xnT7900];

% Save xnT as a wav sound file, soundfile.wav.
audiowrite('Q2soundfile.wav', xnTotal, fs);
%
% Uncomment/edit this next line to save the graph.
exportgraphics(gcf, 'Q2_Graph');
%
```



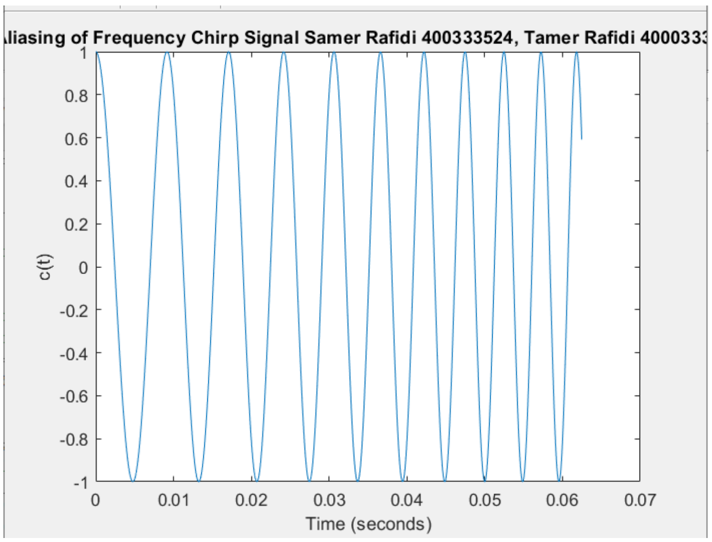
Question 4

Comparing the two different questions, we saw that 100Hz sounds the same as the 7900Hz. This is because the anti-aliasing was not being used for the 7900Hz sound as it only works for sounds below 4000Hz. Anti-aliasing pre-filtering filters out those signals which lowers the frequency. This will allow the sounds to be heard properly. If we would have applied the filtering to the experiments above, the input frequency would increase which would increase the output frequency too. However, since the signal is low pass filtered to lower the frequencies which go past 3500Hz, the output frequencies would not differ as the input frequency increases

Aliasing of Frequency Chirp Signal

Question 1

```
% Samer Rafidi 400333524, Tamer Rafidi 400333527
% Initializing variables
f1 = 100;
u = 2000;
fs = 32000;
Ts = 1/fs;

tfinalplot = (2000-1)*Ts;

% Make the time vector for the plot
nplot=0:Ts:tfinalplot;

% Make the time vector for replayed sound spurt
% Play the spurt for 8 seconds
tfinal = 8;
nsound=0:Ts:tfinal;

%Sample sinusiod
cT = cos(pi*u*nsound.^2+2*pi*f1*nsound);
% Plotting
plot(nplot, cT(1:length(nplot)));
xlabel('Time (seconds)');
ylabel('c(t)');
title('Aliasing of Frequency Chirp Signal Samer Rafidi 400333524, Tamer Rafidi 4000333527');

%Save audio file
audiowrite('Q4soundfile.wav', cT, fs);
```



This sounded similar to what I would hear in an alien movie. Its similar to the start of a police siren and ends with a very high-pitched sound.

When looking at the graph, I see a cosine pattern that keeps going until I have 2000 samples. This happens at just above 60ms.

Question 2

When the sampling frequency is at 16000Hz, I hear what I heard in the 32Khz sound for the first 4 seconds instead of the 8 seconds. For the last 4 seconds, it sounded like the opposite sound was happening to return it back to normal. To describe this in letters, For the 32KHz sound, I heard A B C D in 8 seconds. For 16KHz, I heard A B C D in the first 4 seconds then D C B A in the last 4 seconds.

When the sampling frequency is at 8000Hz, I heard a similar pattern to what I saw in the 32KHz vs 16Khz. However, this pattern is showing for 8KHz vs 16Khz. I am going to explain it similarly to how I explained it above with the letters. In the 16KHz, I heard the A B C D in 4 seconds. In the 8Khz, I heard A B C D in 2 seconds. Comparing 8KHz to 32KHz, I heard the sound 4 times. The pattern I noticed with the fs values was when I decreased the value, the signal happened over a shorter period which results it into repeating itself. If the signal goes through the anti-aliasing filter, we would hear the sound increase in frequency. By the end of the signal, we would not be able to hear what the sound is. If the sound doesn't go through the anti-aliasing filter, we would hear a very similar sound

When I increased the value of f1 from 100Hz to 1000Hz, I found that the pattern was still similar however the sound starts at a much higher pitch and stays becoming higher as the time passed

When I increased the value of u from 2000 to 5000, I saw that it sped up the sound. Knowing this I can assume that lowering the value will decrease how fast it goes through the sound and I will not hear the full sound in the 8 seconds.

Report

For Q1, we can see that the data is discrete by looking at the graph. The shape of the data is like what we see in a sinusoidal graph.

For Q2, we exported the audio file of the 4 tones with a 2 second sample of each. As the tone changed, I started to hear a higher pitch. The first pitch was very low, and the last pitch was very high

For Q3, we exported the audio file of the 4 tones with a 2 second sample of each. As the tone changed, I started to hear a lower pitch. The first pitch is very high (similar to what I heard at 800Hz), and the last pitch was very low (similar to what I heard at 100 Hz). This makes sense when comparing the 2 subplots from Q2 and Q3, we can see that the outputs were complete opposite which was exactly what was heard in the sound files. We believe this is happening because the frequency is too high, and the sampling rate has to be 2x of the input frequency. The significant aliasing is not occurring because of this.