3TP3 LAB THREE

McMaster University

2021.11.16

Cheng Fei_400228518

Yichen Lu_400247938

Q1.

Given the figure 1, a discrete time signal, which is also a sine wave discrete signal, with a period of 0.01s can be seen when the signal is plotted using the stem command.

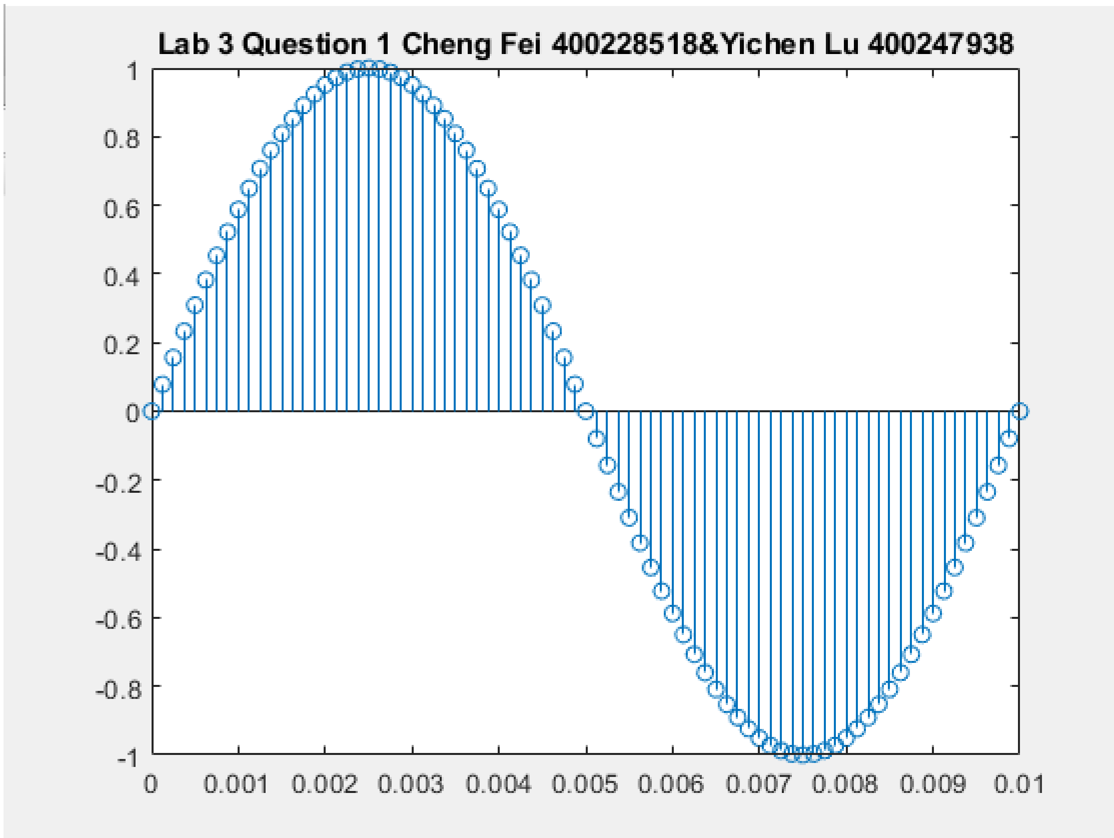


Figure 1

```
%Cheng Fei 400228518&Yichen Lu 400247938
% Q1
% Do a plot of a sampled sinusoid with frequency f = 100 Hz
clc;clear;
f = 100;
% Sampling frequency and interval
fs = 8000;
Ts = 1/fs;
%
% Set time duration of the plot, i.e., 10ms.
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 Question 1 Cheng Fei 400228518&Yichen Lu 400247938');
%
% Uncomment/edit this next line to save the graph.
saveas(gcf, 'Q1_stem_plot.jpg'); %

% Answer to the question:
% We can see that the data is discrete in the graph, but it can be
% understood by human eyes as a sinosoidal wave because the sampling
% frequency is much higher than the wave frequency.
```

The code for Q1

Q2.

Given the figure 2, the sound file produced four sounds, which the first of which was low in frequency and increased in frequency every two seconds. At first, the sound loudness is low and deep, but as the frequency rises, it becomes louder.

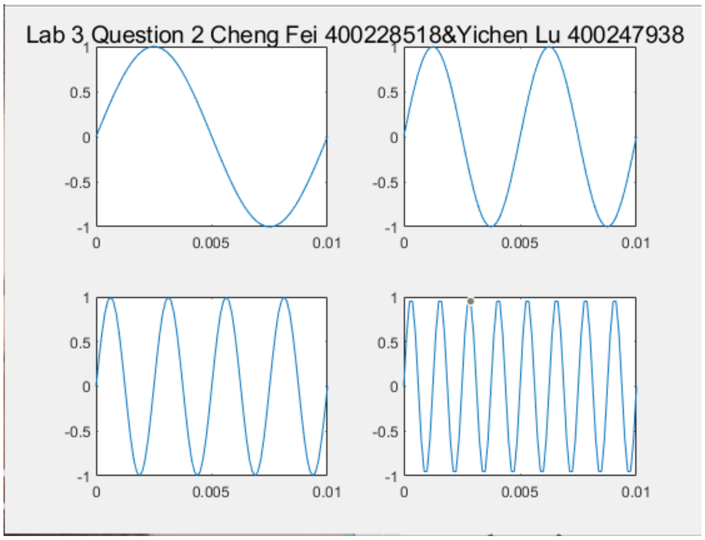


Figure 2

```
%Cheng Fei 400228518&Yichen Lu 400247938
clc;clear;
% 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;
%
% Sample the sinusoid.
xnT1 = sin(2*pi*f1*nsound);
xnT2 = sin(2*pi*f2*nsound);
xnT3 = sin(2*pi*f3*nsound);
xnT4 = sin(2*pi*f4*nsound);
%
% Make the plot
hold on
%suptitle('Lab 3 Question 2 Cheng Fei 400228518&Yichen Lu 400247938')
subplot(2,2,1)
plot(nplot, xnT1(1:length(nplot)));
subplot(2,2,2)
plot(nplot, xnT2(1:length(nplot)));
subplot(2,2,3)
plot(nplot, xnT3(1:length(nplot)));
subplot(2,2,4)
plot(nplot, xnT4(1:length(nplot)));
hold off
%
% Save xnT as a wav sound file, soundfile.wav.
audiowrite('Lab3_Q2_1.wav', xnT1, fs);
audiowrite('Lab3_Q2_2.wav', xnT2, fs);
audiowrite('Lab3_Q2_3.wav', xnT3, fs);
audiowrite('Lab3_Q2_4.wav', xnT4, fs);
%
% Uncomment/edit this next line to save the graph.
saveas(gcf, 'Lab3_Q2.jpg');
%
```

The code for Q2

Q3.

We modify the input frequency to [7200 7600 7800 7900] in this inquiry. However, the output sound wave file sounds louder and has a higher frequency at first, then steadily decreases in volume and frequency every 2 seconds. When contrasted to the pattern observed at lower frequency input, the output frequency is discordant with the input frequency. This is because the sample rate is insufficient due to the input frequency being too high. As mentioned in the handbook, the sampling rate is usually twice that of the input signal. Aliasing may occur if this is not done.

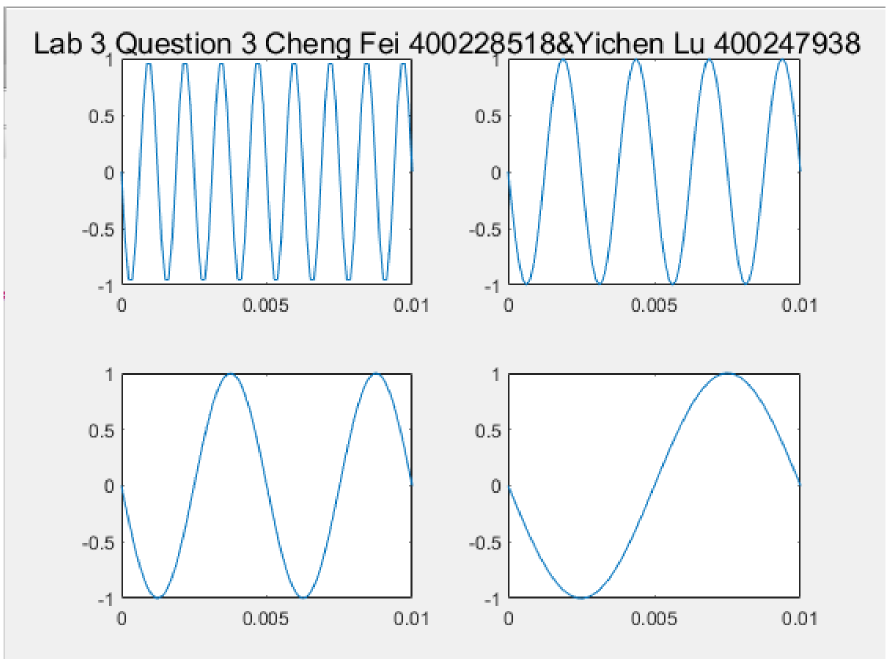


Figure 3

```
%Cheng Fei 400228518&Yichen Lu 400247938
clc;clear;
% 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;
%
% Sample the sinusoid.
xnT1 = sin(2*pi*f1*nsound);
xnT2 = sin(2*pi*f2*nsound);
xnT3 = sin(2*pi*f3*nsound);
xnT4 = sin(2*pi*f4*nsound);
%
% Make the plot
hold on
suptitle('Lab 3 Question 3 Cheng Fei 400228518&Yichen Lu 400247938');
subplot(2,2,1)
plot(nplot, xnT1(1:length(nplot)));
subplot(2,2,2)
plot(nplot, xnT2(1:length(nplot)));
subplot(2,2,3)
plot(nplot, xnT3(1:length(nplot)));
subplot(2,2,4)
plot(nplot, xnT4(1:length(nplot)));
hold off
%
% Save xnT as a wav sound file, soundfile.wav.
audiowrite('Lab3_Q3_1.wav', xnT1, fs);
audiowrite('Lab3_Q3_2.wav', xnT2, fs);
audiowrite('Lab3_Q3_3.wav', xnT3, fs);
audiowrite('Lab3_Q3_4.wav', xnT4, fs);
%
% Uncomment/edit this next line to save the graph.
saveas(gcf, 'Lab3_Q3.jpg');
%
```

The code for Q3

Q4.

If anti-aliasing pre-filtering is not employed, the telephone system's performance will suffer dramatically because sound signals above 4KHz will not be properly sent to the handset on the other end of the line. The behavior of the output signal will differ from that of the original input signal (ex: frequency, volume, etc.). Anti-aliasing pre-filtering, on the other hand, will filter out those signals by lowering their frequency, allowing them to be precisely copied and sent. The behavior of the input signal will be accurately transferred to the output signal in this situation。

If filtering is applied to the above experiments, the output frequency is going to increase if the input frequency is increasing. However, since the signal is first low pass filtered to reduce the frequency components above 3.5KHz, the difference between different output frequencies will be smaller and smaller at high input frequencies.

Q5.

1. The sound file begins with a low frequency and progressively grows in frequency over time (continuously signal). The graph that is being created is shown given figure 5.1.

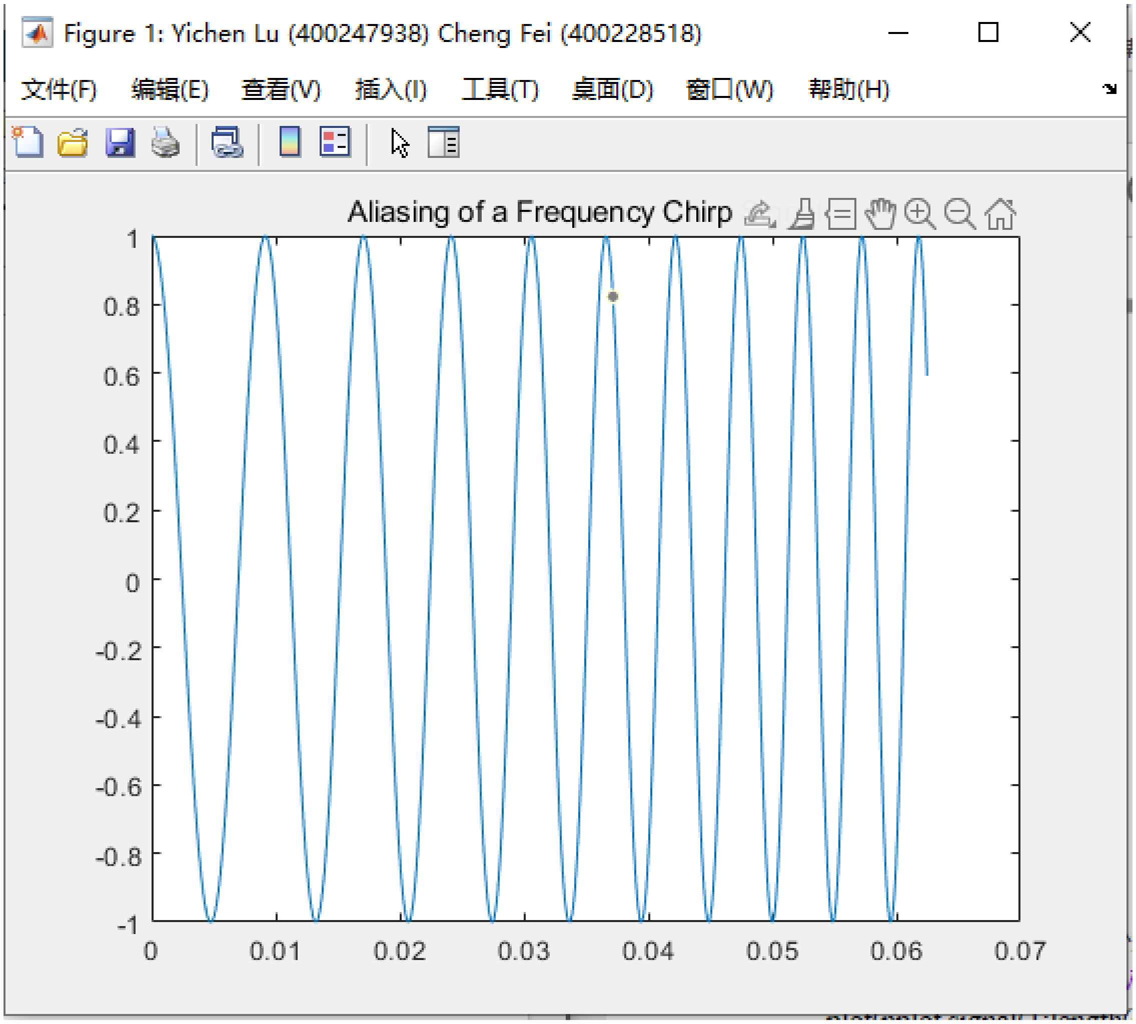


Figure 5.1

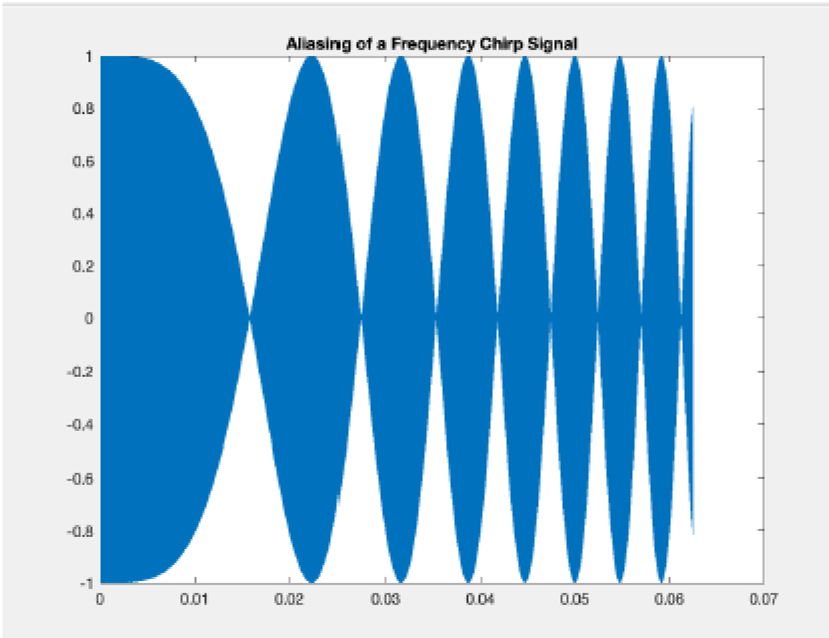
```
%Cheng Fei 400228518&Yichen Lu 400247938
f=100;
u=2000;
fs=32000;
Ts=1/fs;
nplot=0:Ts:Ts*(u-1);
tfinal=8;
nsound=0:Ts:tfinal;

signal=cos(pi*u*nsound.^2+2*pi*f*nsound);
fig=figure('Name', 'Yichen Lu (400247938) Cheng Fei (400228518)');
plot(nplot,signal(1:length(nplot)));
title('Aliasing of a Frequency Chirp Signal');
% Save xnT as a wav sound file, soundfile.wav.
audiowrite('soundfileq5.wav', signal, fs);
%
% Uncomment/edit this next line to save the graph.
exportgraphics(gcf, 'Q5.jpg');
```

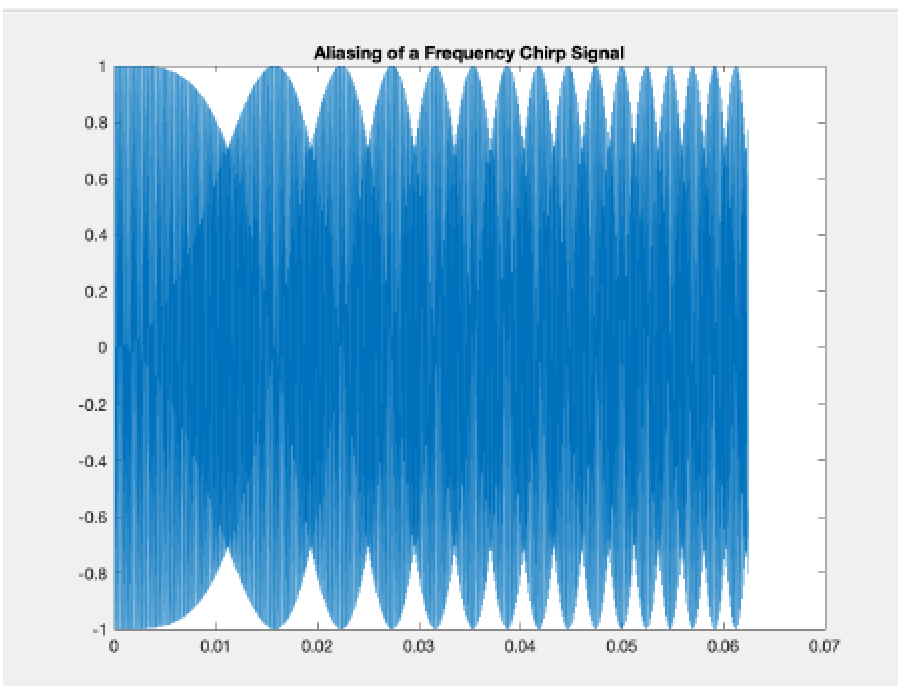
The code for Q5

2. When sampling at 16KHz, the sound grows at first, then progressively drops in frequency until it is exactly the same as it was at the start. When compared to a sound file with a frequency of 16KHz, the sound file with an 8KHz frequency plays twice as fast. Due to the insufficient value of the sampling frequency, the first one generated with 16KHz produced an inconsistent output signal. If no anti-aliasing pre-filtering is used, the output sound generated with 8KHz will be identical to the input sound. If the signal is passed through anti-aliasing filtering, the frequency of the output will grow at a slower rate over time. When the value of f1 is merely increased, the sound will increase in frequency faster and subsequently fall until the sampling frequency is no longer sufficient for sampling. The sound will not change as the fs value is increased since the input frequency will remain within the sampling frequency's range. However, if we gradually lower the fs value, the sound may get distorted since the sampling frequency is lower than the signal's NYquist sampling rate. The frequency of the output signal will increase faster as the value of u is increased until the signal repeats itself. When the value of u is reduced, the frequency of the output signal increases more slowly.

16KHZ


Aliasing of a Frequency Chirp Signal
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07

8KHZ


Aliasing of a Frequency Chirp Signal
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07

.