# Lab 3

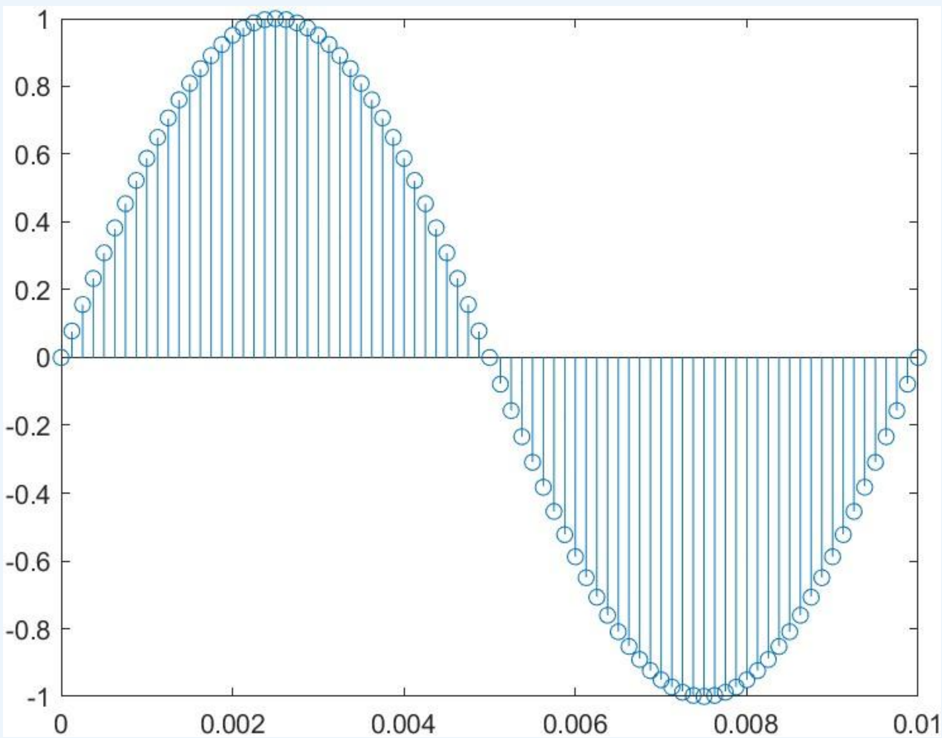
## ELECENG 3TP3

Gurleen Dhillon 400301955

2022-11-21

## Part 1.1

```
clear
f = 100 %Do a plot of a sampled sinusoid with frequency f = 100 Hz
fs = 8000; % Sampling frequency
Ts = 1/fs; %interval
tfinalplot = 10e-3; % Set time duration of plot, i.e., 10 msec.
nplot=0:Ts:tfinalplot; % Make the time vector for the plot
xnT = sin(2*pi*f*nplot); % Sample the sinusoid.
stem(nplot, xnT); % Make the plot
exportgraphics(gcf, 'lab3_part1a.jpg');
```

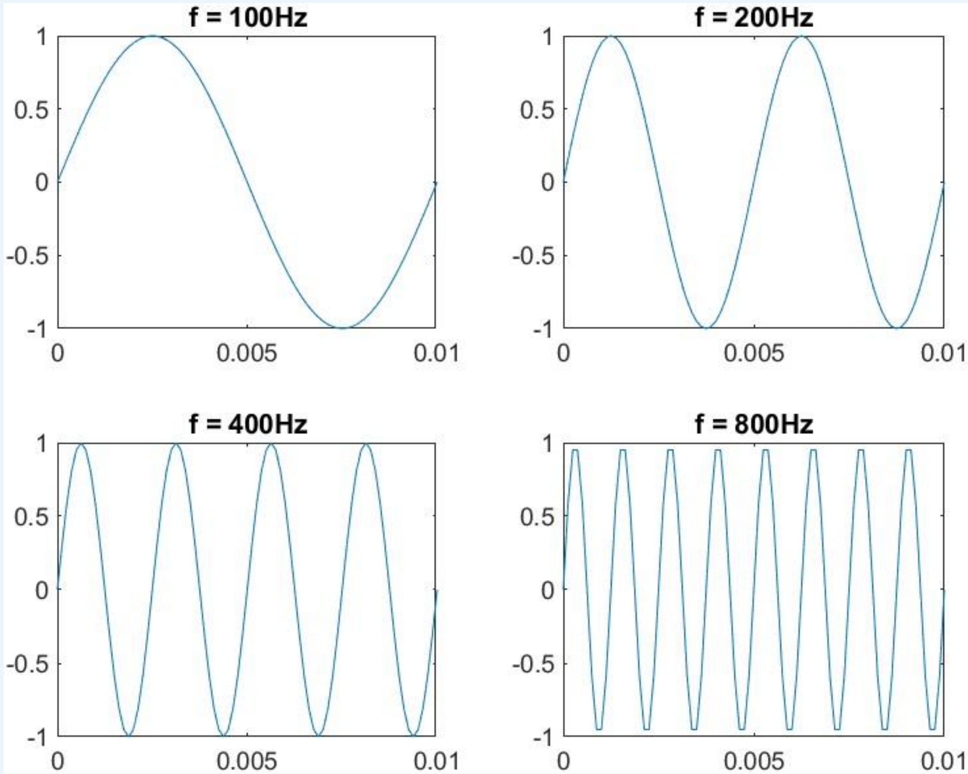



This code allows us to graph the sine function x(nTs) from the lab manual as a stem to individually see the values of the samples at each periodic interval of the sampling period. This graph can be manipulated by simply changing the variable values and equations for f, fs, tfinalplot, and xnT.

## Part 1.2

```
clear
%sinusoid frequencies in Hz
f1 = 100; f2 = 200; f3 = 400; f4 = 800;
fs = 8000; % Sampling frequency in Hz
Ts = 1/fs; %interval
```

```
tfinalplot = 10e-3; % Set time duration of plot 10 msec.
nplot=0:Ts:tfinalplot; % Make the time vector for the plot
tfinal = 2; % Make the time vector for replayed sound spurt
nsound=0:Ts:tfinal; % Play the spurt for 2 seconds

%subplot for sinusoid frequency 100Hz
subplot(2,2,1)
xnT1 = sin(2*pi*f1*nsound); % Sample the sinusoid
plot(nplot, xnT1(1:length(nplot)));
title('f = 100Hz')

%subplot for sinusoid frequency 200Hz
subplot(2,2,2)
xnT2 = sin(2*pi*f2*nsound); % Sample the sinusoid
plot(nplot, xnT2(1:length(nplot)));
title('f = 200Hz')

%subplot for sinusoid frequency 400Hz
subplot(2,2,3)
xnT3 = sin(2*pi*f3*nsound); % Sample the sinusoid
plot(nplot, xnT3(1:length(nplot)));
title('f = 400Hz')

%subplot for sinusoid frequency 800Hz
subplot(2,2,4)
xnT4 = sin(2*pi*f4*nsound); % Sample the sinusoid
plot(nplot, xnT4(1:length(nplot)));
title('f = 800Hz')

%export data
exportgraphics(gcf, 'lab3_part1b.jpg');
audiowrite('lab3_part1b.wav', cat(2, xnT1, xnT2, xnT3, xnT4), fs);
```

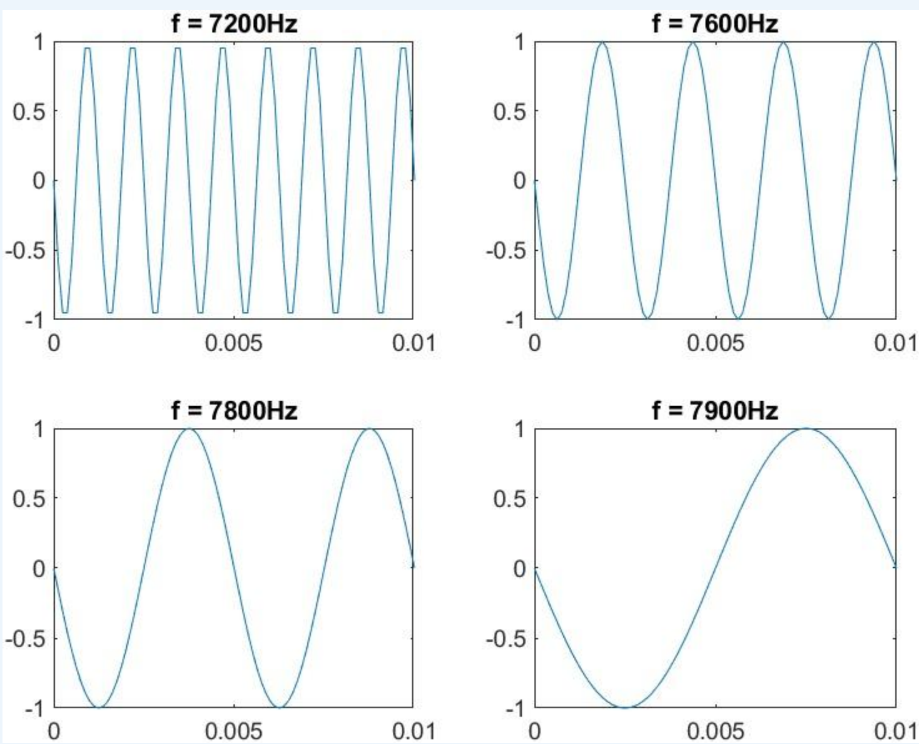



To continue building on the previous code we had, we have now separated the variable f into f1, f2, f3, and f4 in order to graph all 4 frequencies into the same plot using the subplot function. Along with this, we have also added tplot and nsound in order to convert the wave we see into a wave we can hear. As seen from the graphs, as the input frequency increases, the output frequency also increases by a similar rate. By listening to this file, we can understand that as the input frequency is increasing, the pitch also gets higher.

## Part 1.3

```
clear
%sinusoid frequencies in Hz
f1 = 7200; f2 = 7600; f3 = 7800; f4 = 7900;
fs = 8000; % Sampling frequency in Hz
Ts = 1/fs; %interval
tfinalplot = 10e-3; % Set time duration of plot 10 msec.
nplot=0:Ts:tfinalplot; % Make the time vector for the plot
tfinal = 2; % Make the time vector for replayed sound spurt
nsound=0:Ts:tfinal; % Play the spurt for 2 seconds

%subplot for sinusoid frequency 7200Hz
subplot(2,2,1)
```

```
xnT1 = sin(2*pi*f1*nsound); % Sample the sinusoid
plot(nplot, xnT1(1:length(nplot)));
title('f = 7200Hz')

%subplot for sinusoid frequency 7600Hz
subplot(2,2,2)
xnT2 = sin(2*pi*f2*nsound); % Sample the sinusoid
plot(nplot, xnT2(1:length(nplot)));
title('f = 7600Hz')

%subplot for sinusoid frequency 7800Hz
subplot(2,2,3)
xnT3 = sin(2*pi*f3*nsound); % Sample the sinusoid
plot(nplot, xnT3(1:length(nplot)));
title('f = 7800Hz')

%subplot for sinusoid frequency 7900Hz
subplot(2,2,4)
xnT4 = sin(2*pi*f4*nsound); % Sample the sinusoid
plot(nplot, xnT4(1:length(nplot)));
title('f = 7900Hz')

%export data
exportgraphics(gcf, 'lab3_part1c.jpg');
audiowrite('lab3_part1c.wav', cat(2, xnT1, xnT2, xnT3, xnT4), fs);
```



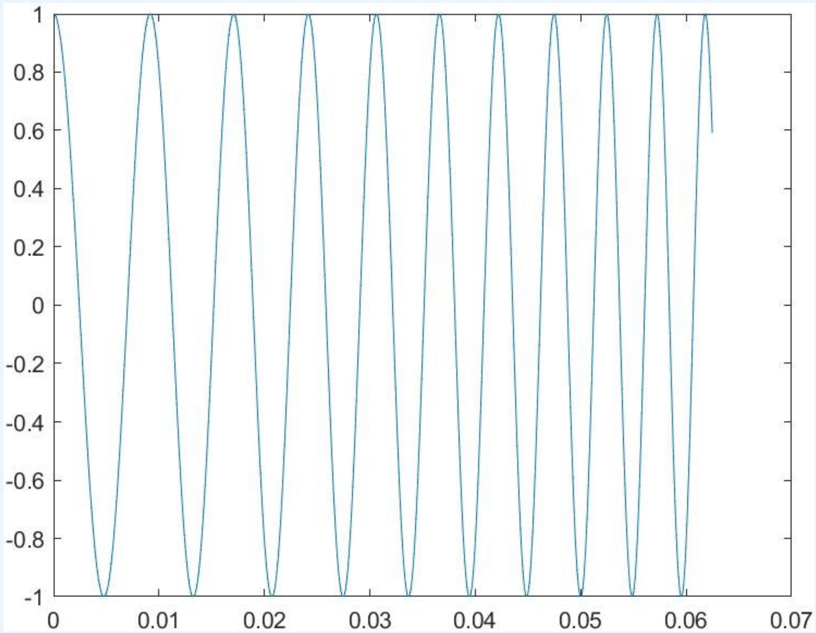

The code for this section is very similar to the last section, but this time we are replacing the values for f1, f2, f3, and f4 with values that are in the thousands compared to the hundreds. As seen from the graphs, as the input frequency is increasing, the output frequency is decreasing, which is the opposite of what was happening in the previous section. With the lower frequencies, the pitch was increasing, but with these higher range frequencies, the pitch is decreasing. This happens because of aliasing and because of the fact that the minimum sampling rate to avoid aliasing falls between the highest frequency from the first section and the lowest frequency from this section. By listening to this file, we can understand that as the input frequency is increasing, the pitch decreases.

## Part 1.4

In this section, we only used anti-aliasing which prevented many distortions in the audio from being transmitted, and if filtering was also added, only frequencies below the sampling frequency would have been used. Anti-aliasing pre-filtering helps remove the portions of the signals which would cause aliasing.

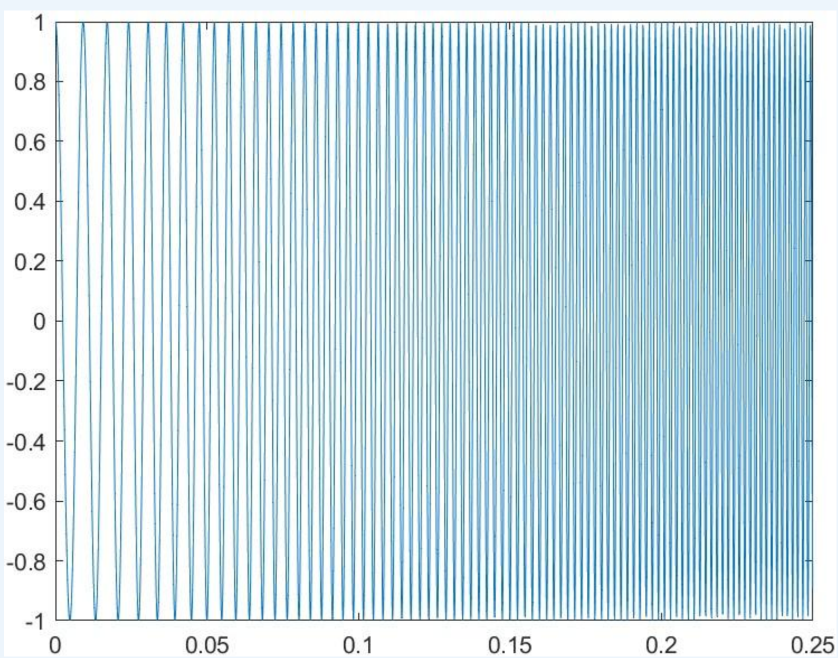
## Part 2.1

```
clear
f = 100; %sinusoid frequency in Hz
u = 2000;
fs = 32000; % Sampling frequency in Hz
ts = 1/fs; %interval
nSamples = 2000;
t=0:ts:8; % Make and play the time vector for replayed sound spurt

%plot
cnT = cos(pi*u*t.^2 + 2*pi*f*t); % Sample the sinusoid
plot(t(1:nSamples), cnT(1:nSamples));

%export audio
audiowrite('lab3_part2a.wav', cnT', fs);
```




As seen from the graphs, as the sampling frequency is decreasing, the output frequency increases. From the audio clips we can conclude that as the sampling frequency decreases, the maximum output pitch decreases.

## Part 2.2



Throughout all 3 parts, as the sampling frequency decreased, the maximum pitch each wave produced was much lower each time. Along with this, the duration for how long the maximum pitch remains on also decreased as the sampling frequency decreased. Due to this, I was only able to hear the pitch hit its maximum value only once in the first section, but in the third section, I was able to hear the pitch twice.

Over a telephone that uses anti-aliasing filtering, aliasing would not occur. This is because telephones operate using the Nyquist sampling rate which is more than double the human voice energy frequency which ensures that the sampling rate is never met.

By experimenting with fs values, it was discovered that as the fs value increases, the higher the maximum pitch gets and the duration that pitch is heard is much longer. By experimenting with $\mu$ values, it was discovered that as the $\mu$ value increases, the wave is played more often and it is played much faster, and due to the increase in its speed, it sounds as if it also has a slightly higher pitch. By experimenting with f1 values, it was discovered that as the f1 value increases, the earlier the output frequency has started to play and causes a positive phase shift.