Full Name: Orkun İbrahim Kök

Department: EEE

Course Code: EEE321

Section: 02

Experiment Number: 01

Date: 13.02.2024

Lab 1 Report

Introduction:

The purpose of this experiment is to generate various sinusoidal signals in MATLAB and observe how different frequency values change the sound of the sinusoids. In further parts of the experiment sum of two sinusoidal signals and sinusoids with varying amplitudes, frequencies, and phases are going to be observed.

musical note	frequency (Hz)
A	440
B flat (B♭)	466
B	494
C	523
C sharp (C#)	554
D	587
D sharp (D#)	622
E	659
F	698
F sharp (F#)	740
G	784
A flat (A♭)	831
A	880

Figure 1: Musical notes and the corresponding frequencies over one octave

Analysis:

Part 1: Fundamental Frequency and Harmonics

In this part, the sinusoidal signal form given below (Eqn. 1) wanted to be tested with $f_0 = 440Hz$, sampling interval $T_s = 0.0001s$ over 3.0 seconds.

$$x_1(t) = \sin(2\pi f_0 t) \qquad \text{(Eqn. 1)}$$

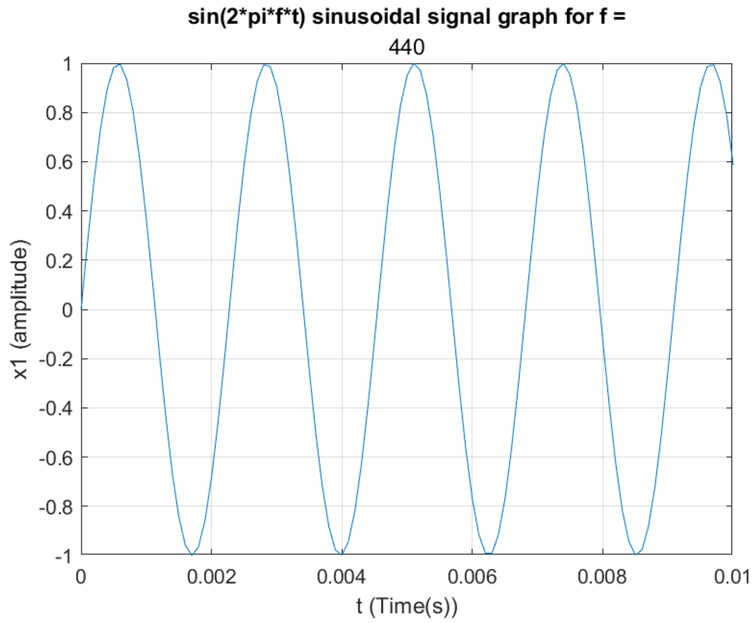

Figure 2: x1(t) graph with $f_0 = 440Hz$

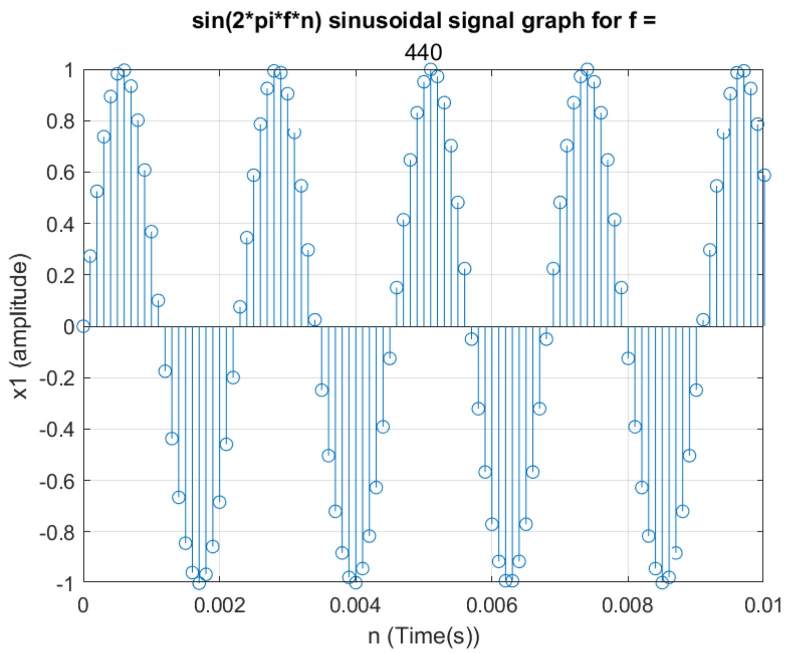

Figure 3: x1[n] graph with $f_0 = 440Hz$

Both of the graphs above (Figure 2 and Figure 3) shows $x_1(t) = \sin(2\pi f_0 t)$ signal form with $f_0 = 440Hz$. Becuase the instruction says discretize your signal, I used stem function (Figure 3). All of the code can be seen in Appendix.

Both continuous and discrete time graphs of $x_1(t) = \sin(2\pi f_0 t)$ sinuoid form with frequency values $f_0 = 880Hz\ and\ 1760Hz$ respectively are given in below figures.

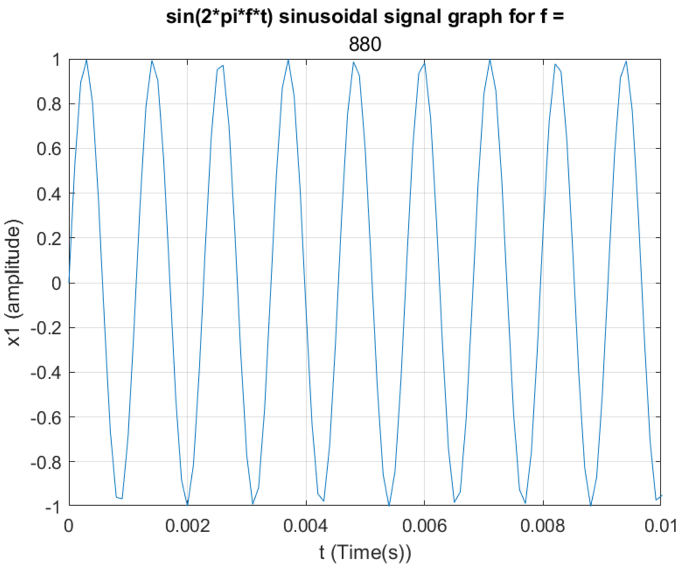

Figure 4: x1(t) graph with $f_0 = 880Hz$

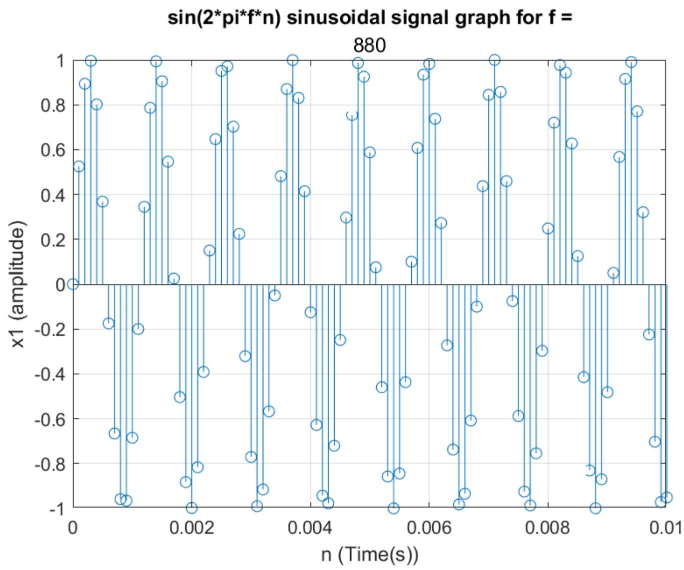

Figure 5: x1[n] graph with $f_0 = 880Hz$

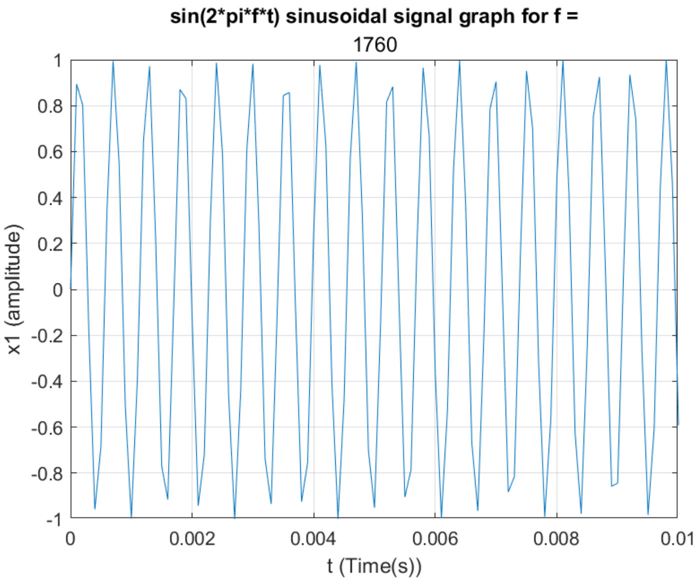

Figure 6: x1(t) graph with $f_0 = 1760Hz$

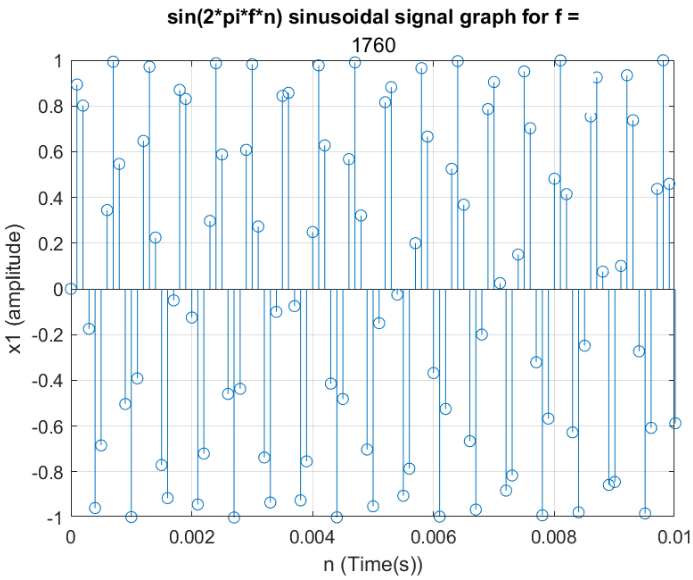

Figure 7: x1[n] graph with $f_0 = 880Hz$

As observed in MATLAB, when the frequency values increase, the pitch of the sound also increases. That is to say, the sound becomes more sharper compared to lower frequency values.

Part 1.2: Major Triad

The goal of this part is combining A (440Hz), C# (554Hz), and E (659Hz) additively in order to create a major triad. The equation form of this combination is given below (Eqn. 2). Again, the sampling interval is chosen as $T_s = 0.0001s$.

$$s(t) = \sin(2\pi 440t) + (2\pi 554t) + (2\pi 659t) \qquad \text{(Eqn. 2)}$$

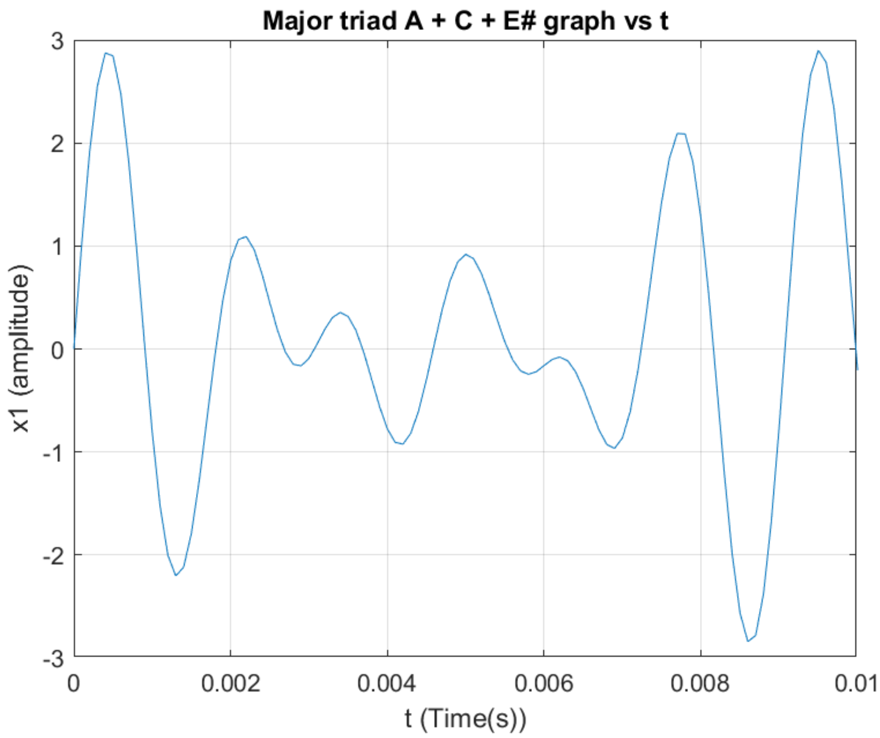

Figure 8: A Major triad graph

Part 2: The Effect of Phase

This part want us to inspect what will be change when the phase angle is changed. For this part, frequency $f_0 = 587Hz$ chosen, which corresponds to note D. The equation of the sinusiod is given in below equation (Eqn. 3).

$$x_2(t) = \cos(2\pi f_0 t + \emptyset) \qquad \text{(Eqn. 3)}$$

At first, $\emptyset = 0$ value chosen. The graph below (Figure 9) shows the corresponding waveform of $x_2(t) = \cos(2\pi f_0 t)$.

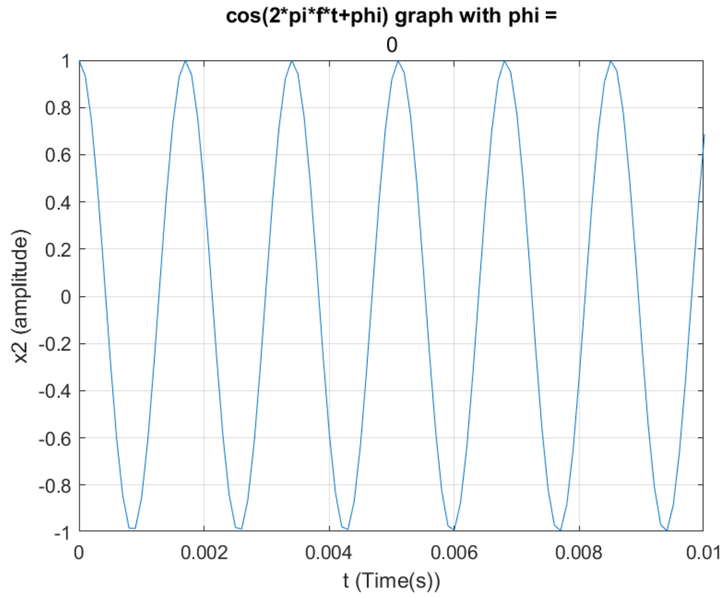

Figure 9: $x_2(t)$ graph with $\emptyset = 0$

Now, the graphs below are same sinuoids with different pahse angles. Starting from Figure 10, phase angles are $\emptyset = \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, and\ \pi$ respectively.

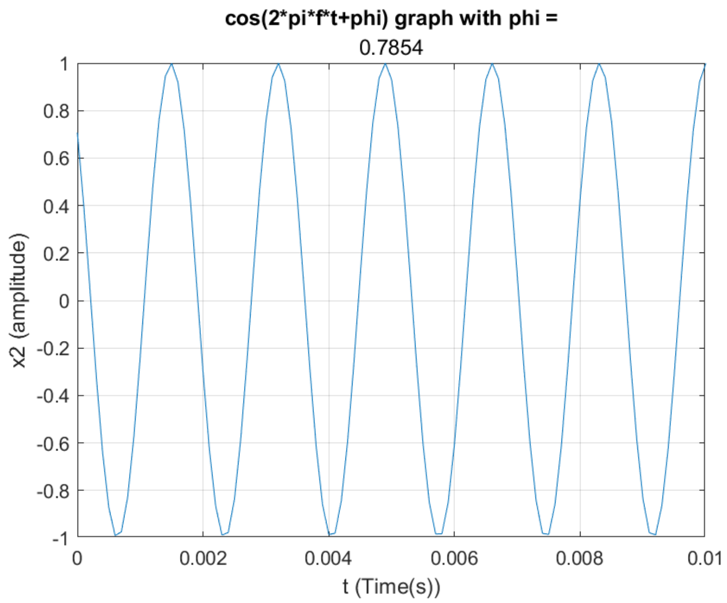

Figure 10: $x_2(t)$ graph with $\emptyset = \frac{\pi}{4}$

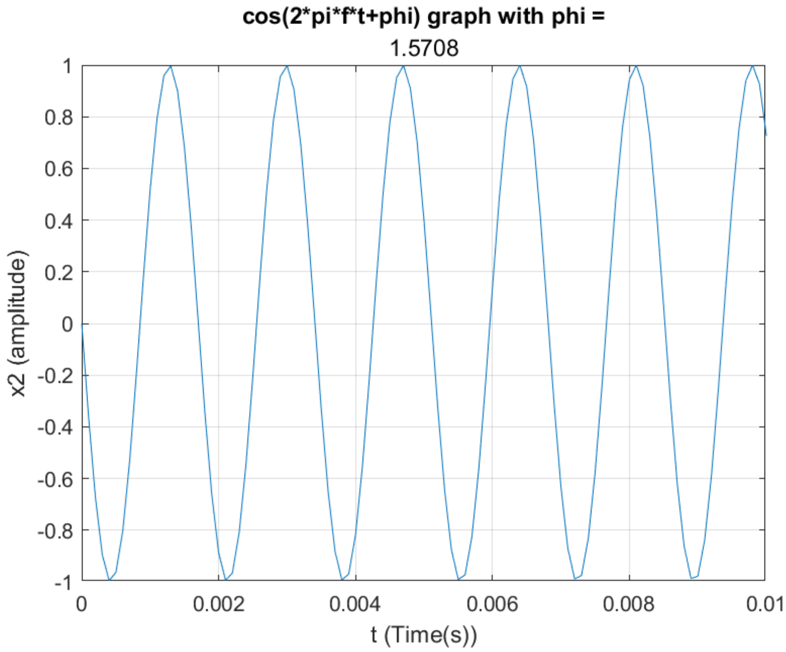

Figure 11: $x_2(t)$ *graph with* $\emptyset = \frac{\pi}{2}$

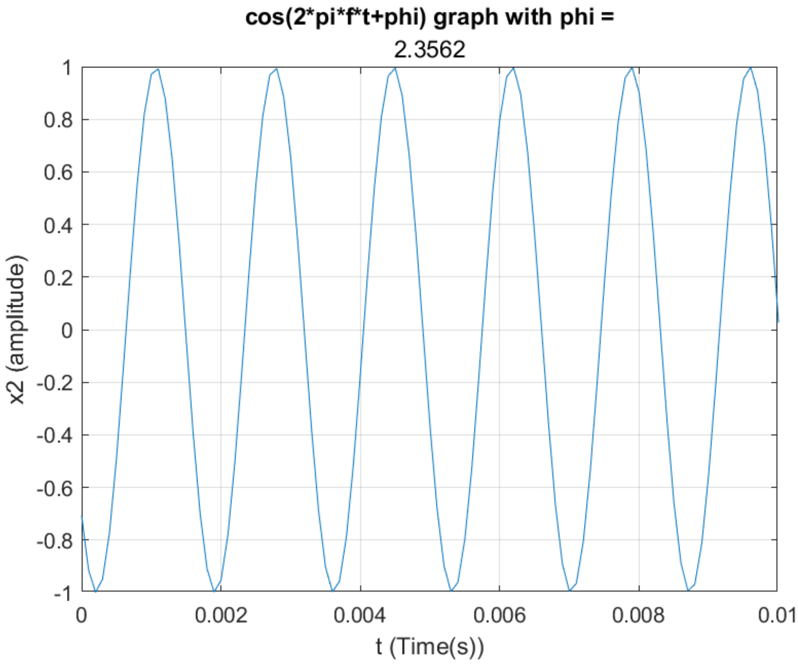

Figure 12: $x_2(t)$ *graph with* $\emptyset = \frac{3\pi}{4}$

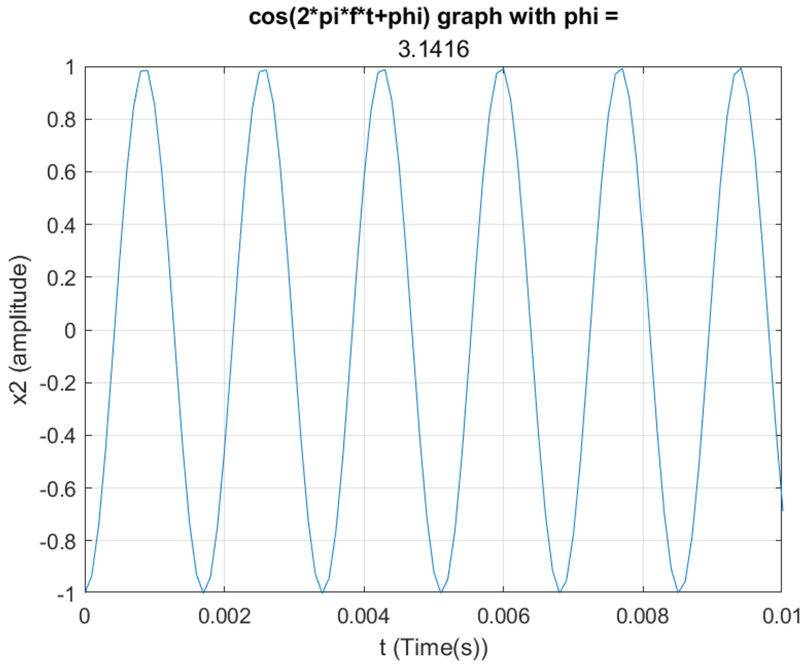

Figure 13: $x_2(t)$ graph with $\emptyset = \pi$

As seen from the graphs above, phase angle Ø makes a difference only at shifting the graph. Also, while listening the sound sinusoid form generates, neither pitch of the sound nor amplitude changed. Therefore, it can be said that phase angle Ø only shifts the graph of the sinusoid signal whether right or left.

Part 3: Sinusoid with Exponentially Decaying Envelope

In this part of the experiment, a combination of sinusoid with an exponential is given and wanted to be observed. The equation of the sinusoid form is given below (Eqn. 4).

$$x_3(t) = e^{-(a^2+2)t}\cos(2\pi f_0 t) \qquad \text{(Eqn. 4)}$$

In MATLAB, to be able to multiply an exponential with a cosine function, I used element-wise multiplication feature.

Frequency value is chosen as $f_0 = 440Hz$ and $a = 2$ initially. The corresponding graph is given below (Figure 14).

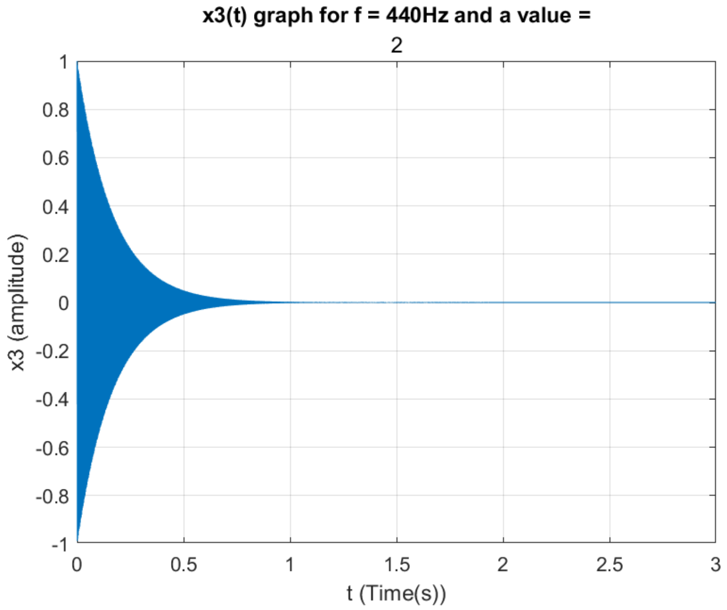

Figure 14: x3(t) graph with a = 2

Below, $x_3(t)$ graphs can be seen with $a = 1 \; and \; a = 3$ values respectively.

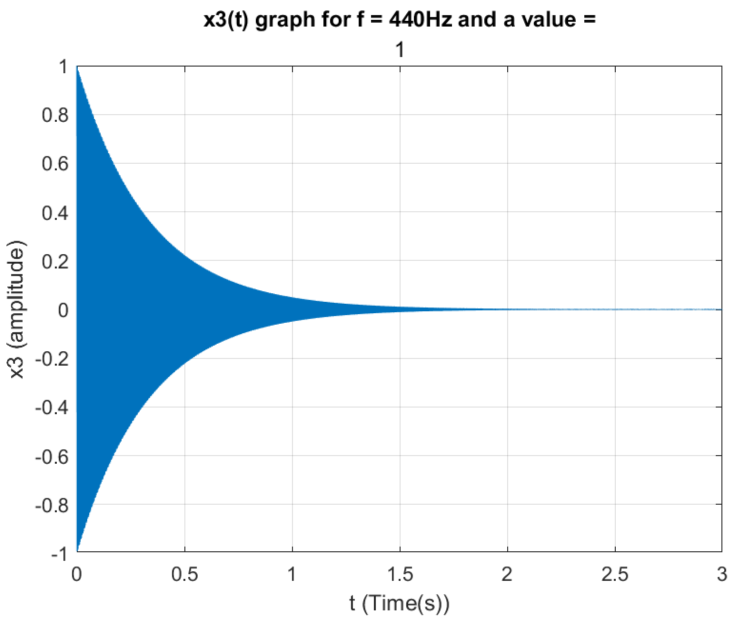

Figure 15: x3(t) graph with a = 1

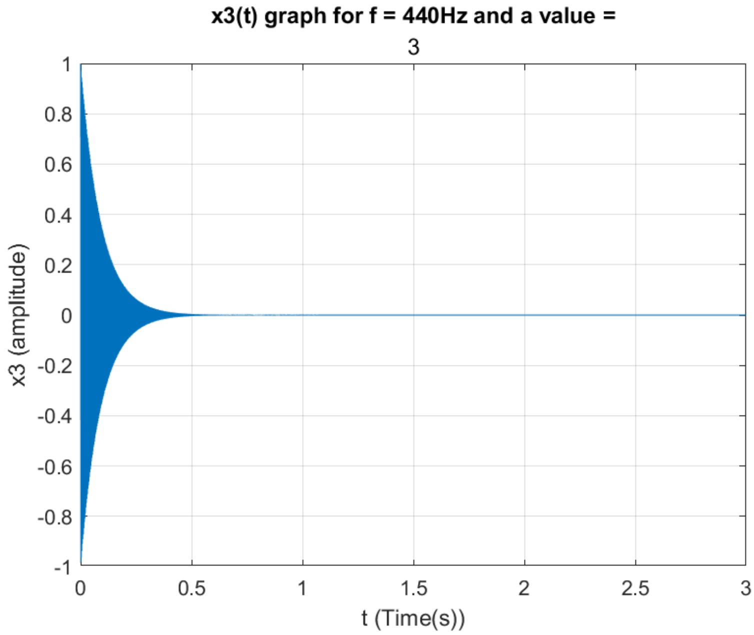

Figure 16: x3(t) graph with a = 3

First of all, because exponential has a negative (-) signed power, it is a decaying function. Futhermore, because we are multiplying it with a cosine function, the cosine function will decay eventually. As a value increases, the speed of decay increases, which gives us a graph like Figure 16 (a = 3) compared to Figure 14 (a = 2). With the same logic, when a value decreases, the speed of decay also slows down. For instance, when we compare Figure 15 (a = 1) with Figure 14 (a = 2), it is seen that higher the a value, tighter the graph gets. In a nutshell, higher the a gets, duration of the sound decreases.

The heard sound resembles to a resonance sound. Comparing Part 3 to Part 1 of the experiment, because both sinusoids have frequency value $f_0 = 440Hz$, the initial sounds resemble each other. However, in Part 1, the sound continues without decay, in Part 3, sound decays eventually.

I think $x_1(t)$ signal resembles to piano sound and $x_2(t)$ signal resembles more to flute sound.

Part 4: Beat Notes and Amplitude Modulation

In this part, purpose is observing two multiplied cosine function signal. If we choose the frequencies far apart from each other, we can get a beat note. The only conditions is to choose one frequency value very small and the other around 1kHz. In this part, the chosen frequency values are $f_1 = 10Hz \text{ and } f_2 = 1000Hz$. The equation of the considered signal form is given below (Eqn. 5).

$$x_4(t) = \cos(2\pi f_1 t)\cos(2\pi f_2 t) \qquad \text{(Eqn. 5)}$$

Graph below (Figure 17) shows how a beat node graph looks like.

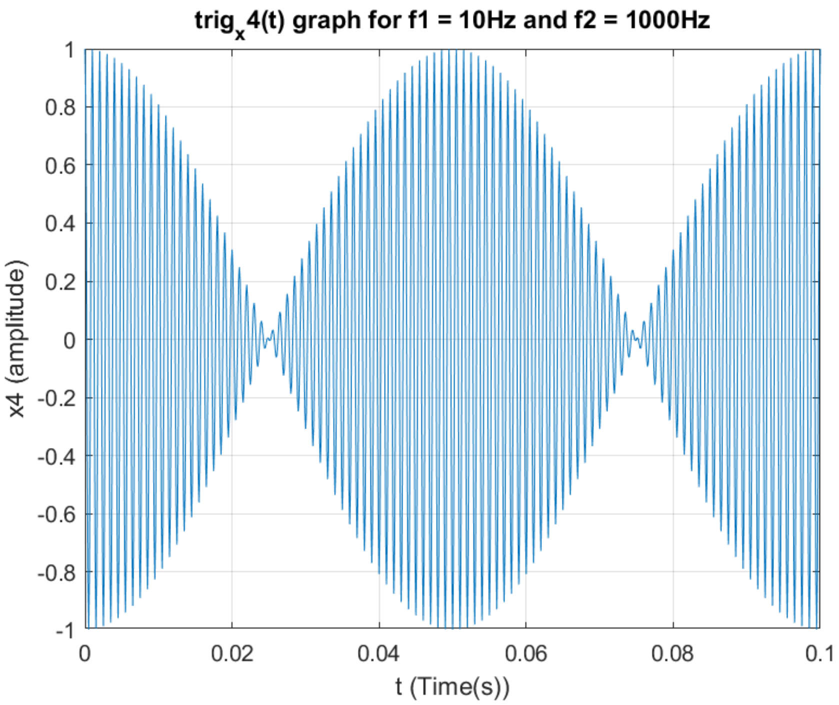

Figure 17: x4(t) graph with $f_1 = 10Hz\ and\ f_2 = 1000Hz$

To make a comparison about how low-frequency cosine term changes the sound, $x_4(t)$ graph with $f_1 = 5Hz\ and\ 15Hz$ respectively given below.

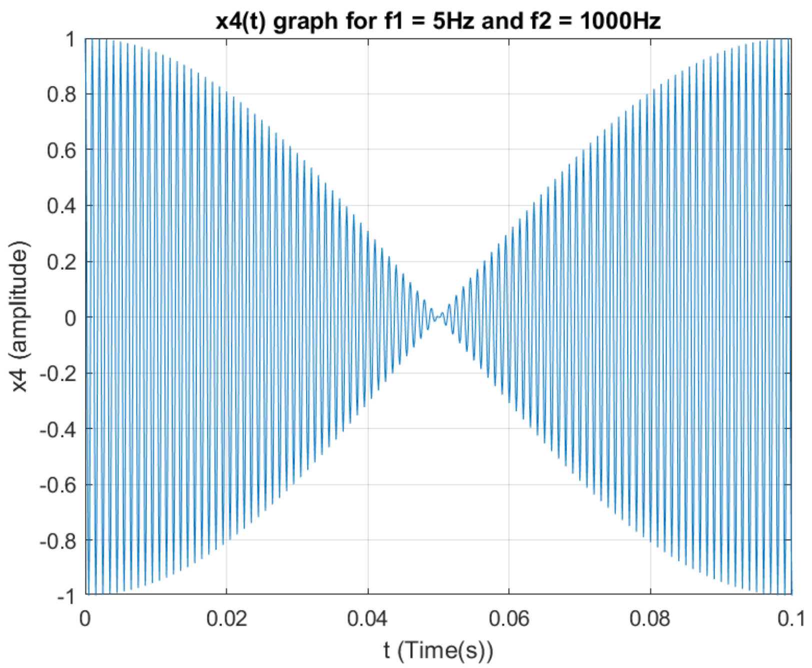

Figure 18: x4(t) graph with $f_1 = 5Hz\ and\ f_2 = 1000Hz$

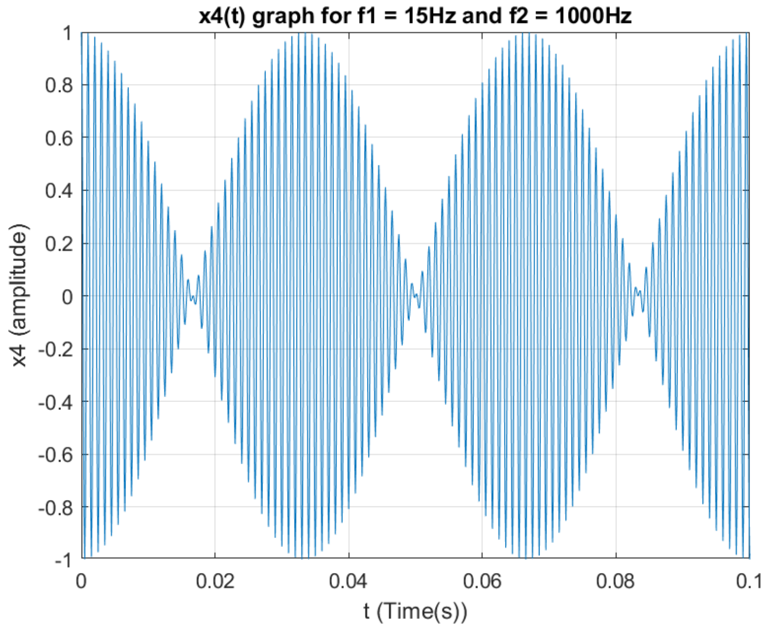

Figure 19: x4(t) graph with $f_1 = 15Hz\ and\ f_2 = 1000Hz$

As the f_1 value increases, duration between the beats become narrower and vice versa when f_1 decreases. Therefore, the low-frequency cosine term determines the rate of the beat and the duration between each beat. Such a process which multiplying a low frequency signal with a high frequency signal is called "*amplitude modulation (AM)*".

Using the well-known trigonometric identity (Eqn. 6), I computed the same signal form. Because they are the same signal expressed in different ways, their signal form graphs are same. Below given graphs for (Eqn. 6) for $f_2 = 1000Hz$ and $f_1 = 5Hz, 10Hz, and\ 15Hz$ respectively.

$$\cos(\theta_1)\cos(\theta_2) = \frac{1}{2}[\cos(\theta_1 + \theta_2) + \cos(\theta_1 - \theta_2)] \qquad \text{(Eqn. 6)}$$

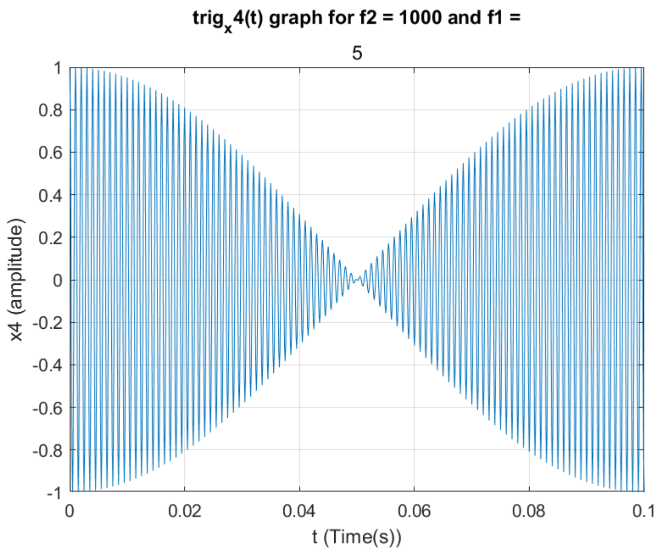

Figure 20: x4(t) graph with $f_1 = 5Hz\ and\ f_2 = 1000Hz$ with Eqn.6

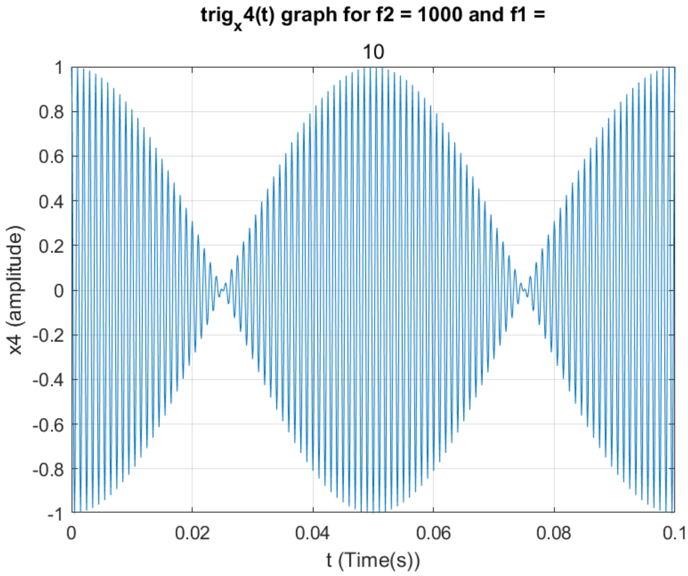

Figure 21: x4(t) graph with $f_1 = 10Hz\ and\ f_2 = 1000Hz$ with Eqn.6

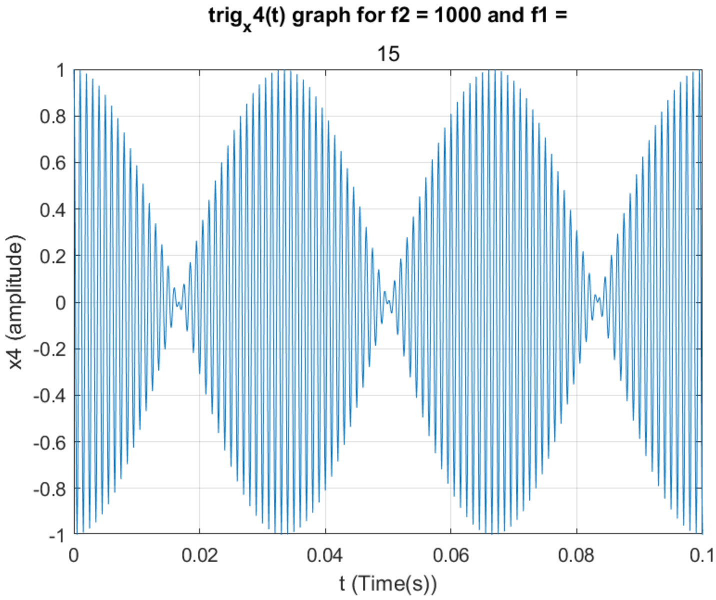

Figure 22: x4(t) graph with $f_1 = 15Hz\ and\ f_2 = 1000Hz$ with Eqn.6

Part 5: Chirp Signals and Frequency Modulation

In this part, time varying frequencies are going to be considered and observed. (Eqn. 7) shows the general form of a cosine function with varying angle.

$$x(t) = Acos[\psi(t)] \qquad \text{(Eqn. 7)}$$

The derivative with respect to time (t) of angle $\psi(t)$ in the equation above gives us a radian frequency changing in time (Eqn. 8).

$$\omega(t) = \frac{d}{dx}\psi(t) \qquad \text{(Eqn. 8)}$$

A chirp signal is a sinusoid whose frequency is linearly swept so that it changes linearly from a starting value to an ending one. The general formula for such a signal can be seen below (Eqn. 9).

$$\psi(t) = 2\pi\mu t^2 + 2\pi f_0 t + \phi \qquad \text{(Eqn. 8)}$$

Taking the derivative with respect to time of (Eqn. 8) will give us the equation below (Eqn. 9).

$$f_i(t) = 2\mu t + f_0 \qquad \text{(Eqn. 9)}$$

In the equation above, f_i represents the ending frequency value and f_0 represents the starting frequency value. This type of frequency variation porduced by a time varying angle is called “frequency modulation (FM)”. These type of signals create a sound similar to a siren or chirp, that is why they are called “chirp signals”.

The equation below shows the signal form used in this part of the experiment (Eqn. 10).

$$x_5(t) = \cos(2\pi\mu t^2 + 2\pi f_0 t + \phi) \qquad \text{(Eqn. 10)}$$

In this case, $f_i = 500Hz$ and $f_0 = 2500Hz$ over 2 seconds. The sampling interval is $T_s = 0.0001s$. Because the frequency is varying, the graph of this signal does not comprise of repeating curves. Figure 23 explains the occassion better what is time-varying frequency in a graph.

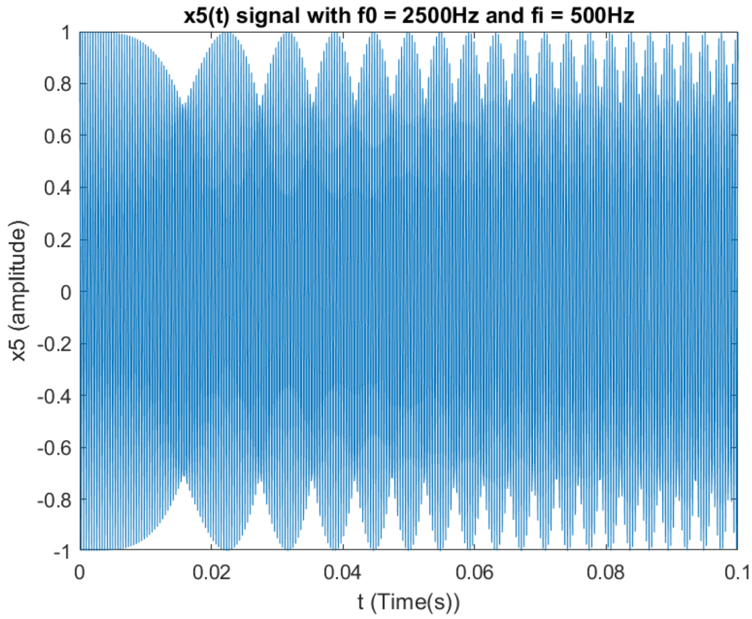

Figure 23: x5(t) signal waveform with , $f_i = 500Hz$ and , $f_0 = 2500Hz$

As seen from the graph above, the length between the peak points of the sinusoid are varying, or can be said decreasing. The logic of this is because of the starting and end frequency values. We started with 2500Hz frequency and ended at 500Hz frequency. The sound of the signal is something like a siren. The sound starts with a bass-sound and finishes with a high-pitched sound. Using (Eqn.9), μ value is found as -500.

When $f_i = 500Hz$ and $f_0 = 2500Hz$ values are changed between each other, namely $=$ $2500Hz$ and $f_0 = 500Hz$ are the new values, we get the reversed version of a sound of the signal form observed above. In this case, sound starts with a high-pitch nad finishes with a bass sound compared to the starting sound. Again using (Eqn.9), μ value is found as 500.

When μ value is doubled, the sound of the signal becomes more rapid, and as expected, when we halve μ value, sound became slower over the time. That is to say, from starting frequency to end frequency, throughout this duration the increasing or decreasing time of the signal is related to the μ value. If μ value is big, time interval from starting frequency to end frequency is shorter, and if μ value is smaller, time interval from starting frequency to end frequency is longer.

Part 5.2: Chirp Puzzle

For this part, frequency starts from positive value and ends at negative frequency value. Eqn. 10 is again used in this part. Start and end frequency values are chosen as $f_i = -2000Hz$ and $f_0 = 3000Hz$. However, because cosine is an even function, negative frequency value does not change anything and signal will act like as positive frequency value.

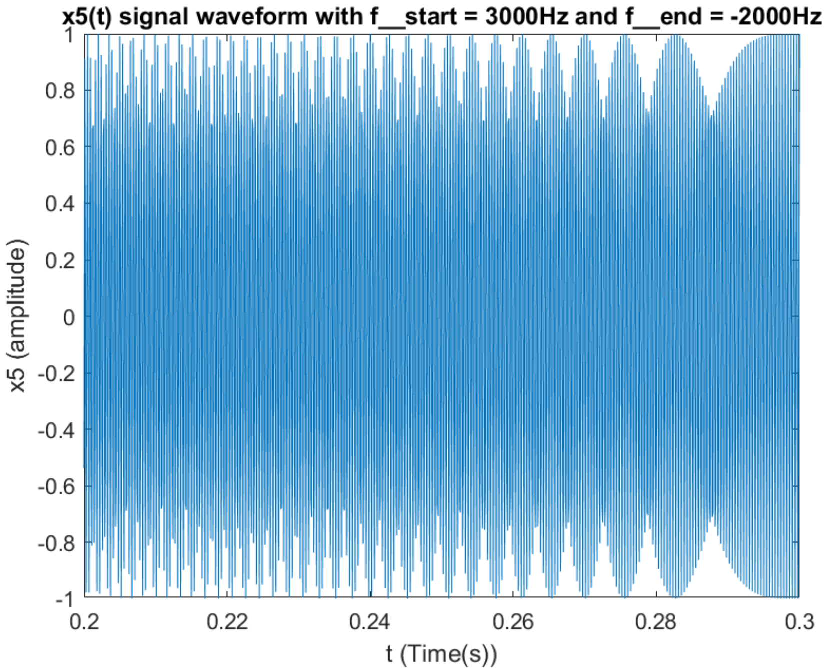

Figure 24: x5(t) signal waveform with , $f_i = -2000Hz$ *and ,* $f_0 = 3000Hz$

Starting from 3000Hz frequency, the sound of the signal first chirp down to 0Hz, and then it chirps up to -2000Hz. However, as I mentioned above, cosine function will take -2000 value as 2000 value.

Part 6

The code for this part can also be found in Appendix. The audio file of the composed music file is uploaded to Moodle.

Appendix

PART 1

```
t = [0:0.0001:3]
f = [440, 880 1760]
for freq = f
    x1 = sin(2*pi*freq*t)
    figure;
    stem(t,x1)
    xlabel('n (Time(s))')
    ylabel('x1 (amplitude)')
    xlim([0 0.01])
    grid
    title('sin(2*pi*f*n) sinusoidal signal graph for f =',num2str(freq))
    soundsc(x1, 10000)
    pause(3.0)
end

%more sharp sound is heard when the frequency is increased
```

PART 1.2

```
s = sin(2*pi*440*t) + sin(2*pi*554*t) + sin(2*pi*659*t)
t = [0:0.0001:3]
plot(t,s)
xlabel('t (Time(s))')
ylabel('x1 (amplitude)')
xlim([0 0.01])
grid
title('Major triad A + C + E# graph vs t')
soundsc(s, 10000)
```

PART 2

```
f = 587
t = [0:0.0001:3]
phi_values = [0, pi/4, pi/2, 3*pi/4, pi];
for phi = phi_values
    x2 = cos(2*pi*f*t+phi);
    figure;
    plot(t,x2);
    xlabel('t (Time(s))');
    ylabel('x2 (amplitude)');
    grid on;
    title('cos(2*pi*f*t+phi) graph with phi =', num2str(phi) )
    xlim([0 0.01]);
    soundsc(x2, 10000);
    pause(3);
end

%nothing has been changed when the phi values are changed. Phi values only
%make a difference in the plot of the signal to shift. The pitch and volume
%of the sound stays the same.
```

PART 3

```
a_val = [1, 2, 3];
f = 440;
t = [0:0.0001:3];
for a = a_val
    x3 = exp(-((a^2)+2)*t).*cos(2*pi*f*t);
    figure;
    plot(t,x3);
    xlabel('t (Time(s))');
    ylabel('x3 (amplitude)');
    title('x3(t) graph for f = 440Hz and a value =', num2str(a));
    grid on;

    soundsc(x3,10000);
    pause(3);
end

%the duration of the sound decreases when the a value is increasing. A
%resonance is heard from the x3(t) signal. In part 1, the amplitude of the
%signal does not change, however in part 3, the amplitude is gradually
%decreases and becomes zero eventually. The exponential term maybe the
%reason behind it. I think x1 resembles to piano and x2 resembles to flute
%sound.
%As a increases decay speed also increases therefore we hear shorter sound
```

PART 4

```
f1 = [5, 10, 15];
f2 = 1000;
t = [0:0.0001:3]
for freq = f1
      x4 = cos(2*pi*freq*t).*cos(2*pi*f2*t)
      trig_x4 = 0.5*(cos(2*pi *(freq+f2)*t)+cos(2*pi*(f2-freq)*t));
      figure;
      plot(t, x4)
      xlabel('t (Time(s))')
      ylabel('x4 (amplitude)')
      title('x4(t) graph for f2 = 1000 and f1 =', num2str(freq))
      grid on;
      xlim([0 0.1])
      soundsc(x4,10000)
      pause(3.0);

      figure;
      plot(t, trig_x4)
      xlabel('t (Time(s))')
      ylabel('x4 (amplitude)')
      title('trig_x4(t) graph for f2 = 1000 and f1 =', num2str(freq))
      grid on;
      xlim([0 0.1])
      soundsc(x4,10000)
      pause(3.0)
end
%the low frequency term determines the rate of the beats. For example when
%the low freq is 5, the time interval between the beats are more visible
%than it is when the frequency value is 10.
%Also nothing is changed when I wrote the eqaution as sum of two cosine
```

```
%functions.
%As frequency difference increases sound becomes more hearable.
```

PART 5

```
%2500 start, 500 end
T = 2
t = [0:0.0001:2]
f_s1 = 2500
f_end1 = 500
u1 = (f_end1-f_s1)/(2*T)
phi = 0
x5_1 = cos(2*pi*u1*t.^2 + 2*pi*f_s1*t + phi)
figure;
plot(t,x5_1)
xlabel('t (Time(s))')
ylabel('x5 (amplitude)')
title('x5(t) signal with f0 = 2500Hz and fi = 500Hz')
xlim([0 0.1])
soundsc(x5_1, 10000)
pause(3)

%500 start, 2500 end
f_s2 = 500
f_end2 = 2500
u2 = (f_end2-f_s2)/(2*T)
x5_2 = cos(2*pi*u2*t.^2 + 2*pi*f_s2*t + phi)
soundsc(x5_2, 10000)
pause(3)

%double u
double_u = (u2)*2
x5_3 = cos(2*pi*double_u*t.^2 + 2*pi*f_s2*t + phi)
soundsc(x5_3, 10000)
pause(3)

%half u
u2 = (f_end2-f_s2)/(2*T)
half_u = (u2)/2
x5_4 = cos(2*pi*half_u*t.^2 + 2*pi*f_s2*t + phi)
soundsc(x5_4, 10000)

%From 2500Hz to 500Hz or 500Hz to 2500Hz, a whistle type of sound is
%observed. As the frequency gets higher, sharper the sound gets and vice
%versa.
%As the u value increases, the sound heard is more rapid, like the
% sharpness of the increase in the sound is quicker compared to a low u value.
```

PART 5.2

```
f_s = 3000
f_end = -2000
T = 3
t = [0:0.0001:3]
phi = 0
u = (f_end-f_s)/(2*T)
```

```
x5 = cos(2*pi*u*t.^2 + 2*pi*f_s*t + phi)
plot(t,x5)
xlim([0.2 0.3])
title('x5(t) signal waveform with f__start = 3000Hz and f__end = -2000Hz')
xlabel('t (Time(s))')
ylabel('x5 (amplitude)')
soundsc(x5, 10000)

%first the sharpness (freq) of the sound decreases and becomes a deep
%voice, then the sharpness increases again but not to the starting value.
%So, first the sound chirps down, then it chirps up.
```

PART 6

```
function [note] = notecreate(frq_no, dur)
note = sin(2*pi*[1:dur]/8192*(440*2.^((frq_no-1)/12)));
end

notename = {'A', 'A#', 'B', 'C', 'C#', 'D', 'D#', 'E', 'F', 'F#', 'G', 'G#'};
song = {'E', 'E', 'E', 'E', 'E', 'E', 'E', 'G', 'C', 'D', 'E', 'F', 'F', 'F', 'E',
'E', 'E', 'D', 'E', 'D', 'G', 'E', 'E', 'E', 'E', 'E', 'E', 'E', 'G', 'C', 'D',
'E','F', 'F', 'E', 'E', 'E', 'G', 'G', 'F', 'E', 'D',};
songidx = zeros(1, length(song));
for k1 = 1:length(song)
    idx = strcmp(song{k1}, notename);
    songidx(k1) = find(idx);
end
dur = 0.3 * 8192;
songnote = [];
for k1 = 1:length(songidx)
    songnote = [songnote; [notecreate(songidx(k1), dur) zeros(1, 75)]'];
end
soundsc(songnote,8192);
audiowrite('myaudio_jinglebells.wav', songnote, 8192)
```