# EEE391

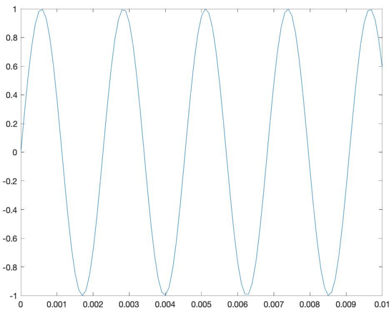
## Basics of Signals and Systems
## MATLAB Assignment 1

**Muhammad Arham Khan**
**21701848**
**3rd March, 2020**

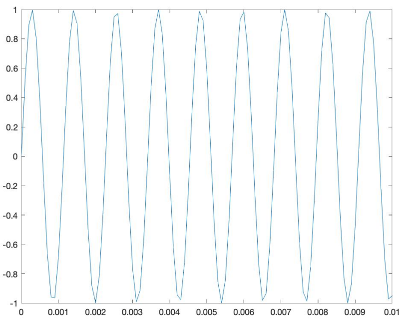
# Part 1:

$$x_1(t) = \sin(2\pi f_o t).$$

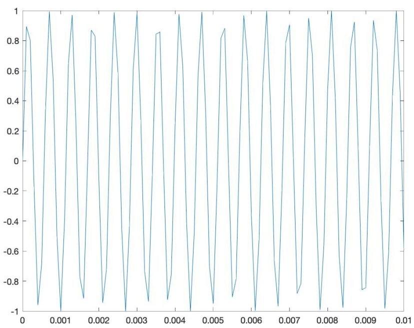
Frequency 440Hz:




Frequency 880Hz:

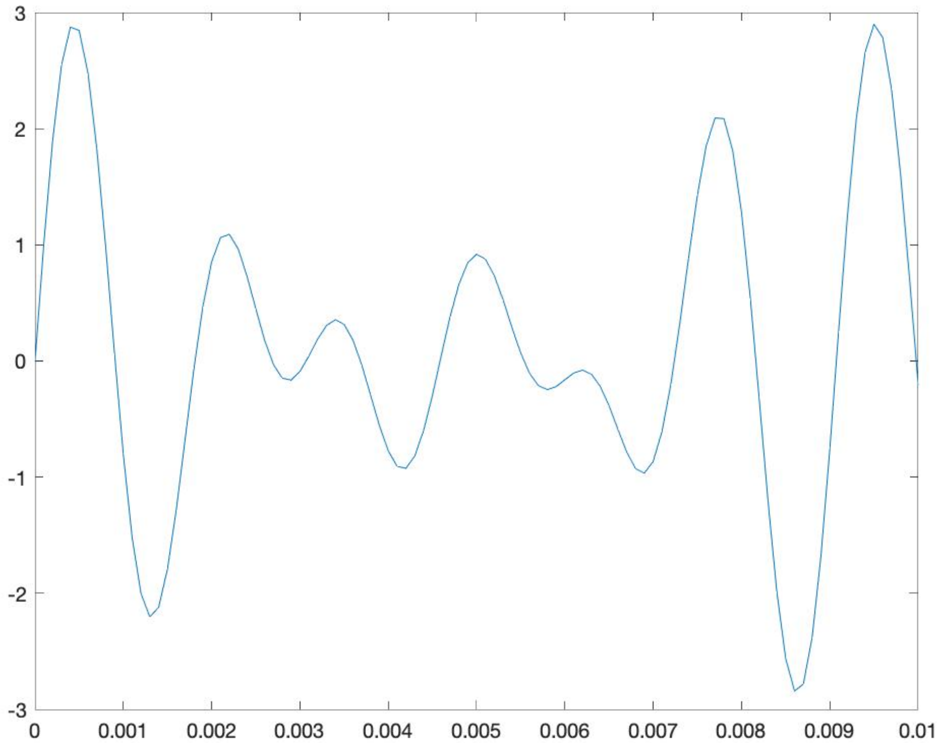



Frequency 1760Hz:



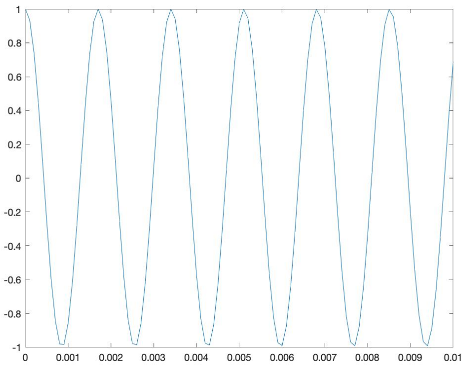

**Analysis:** As we increase the frequency of the signal (from 440Hz to 1760Hz), the pitch of the sound produced increases as well and the sound sounds sharper

**$s(t) = \sin(2\pi 440\ t) + \sin(2\pi 554\ t) + \sin(2\pi 659\ t)$**

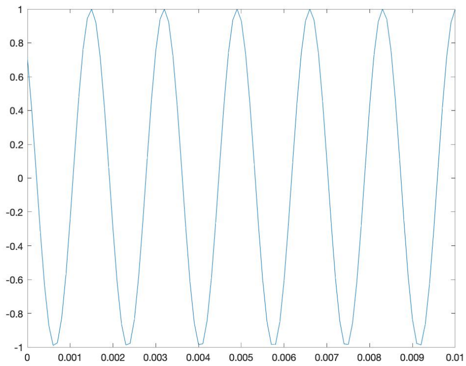


## Part 2:

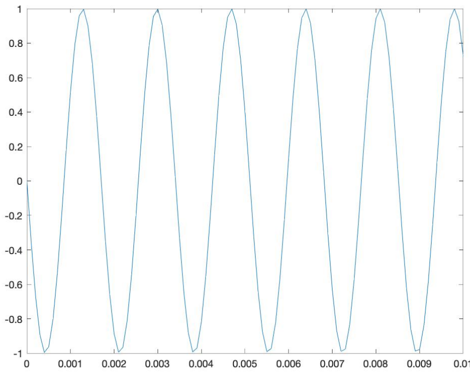
$\Phi = 0$



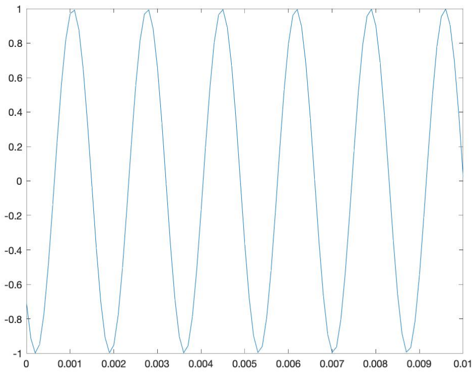

$\Phi = \pi / 4$



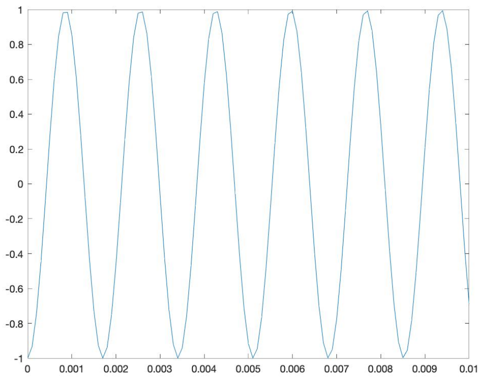

$\Phi = \pi / 2$




$\Phi = 3\pi / 4$




$\Phi = \pi$

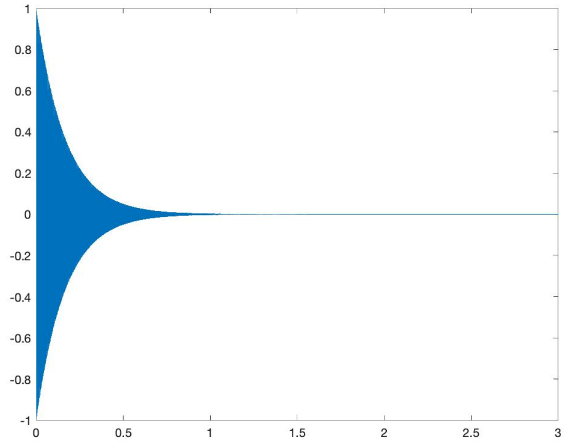



**Analysis:** There is no observed change in the pitch of the sound when the phase changes. But there is a change in the overall sound heard due to the phase difference. The sound oscillations vary per time and hence, the sound although has the same volume and pitch, it is out-of-phase (oscillations do not align).

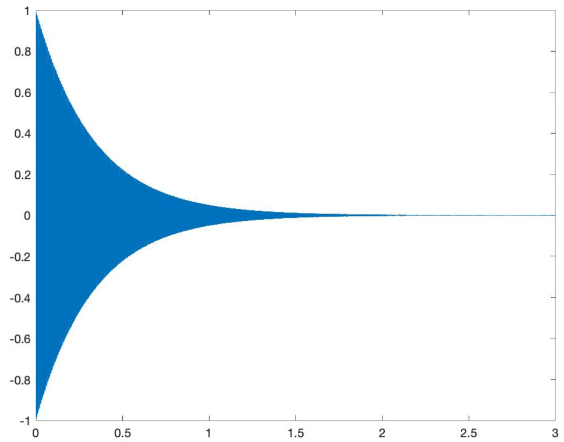
## Part 3:

$$x_3(t) = e^{-(a \wedge 2+2)t} \cos(2\pi f_o t)$$

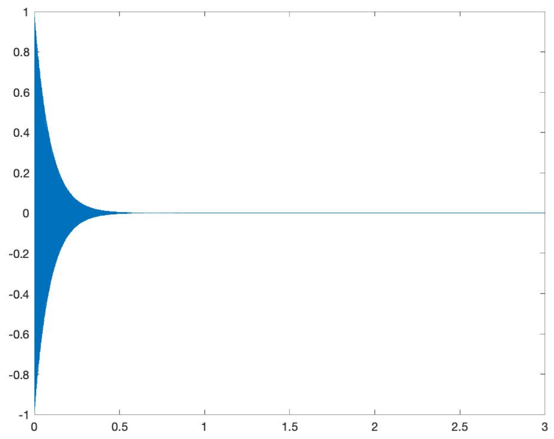
$f_0$ = 440Hz, a = 2:




$f_0$ = 440Hz, a = 1:




$f_0$ = 440Hz, a = 3:




Single line code:

**$X_3$ = cos( 2 * pi * f1 * t) .* cos( 2 * pi * f2 * t);**

**Analysis:**
When we compare x1(t) with this result, it is evident that the initial sound is the same but since x3(t) has an exponentially decreasing gradient, it's amplitude dampens over time while the x1(t) has the same amplitude and a constant voice.

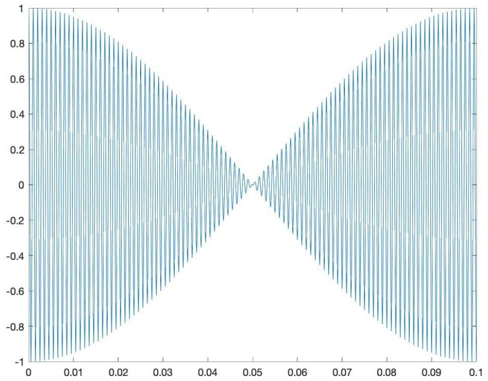
X2(t) produces a sound similar to that of a piano while x1(t) produces one that is like a flutes sound.

As the value of a increases, the duration that the sound is audible for decreases. This is due to the exponential decrease on the output values which present a dampening effect on the amplitude of the signal.
I can relate this situation with the dampening effect a door stopping spring has on the door. It doesn't stop the door right away but decreases it's amplitude

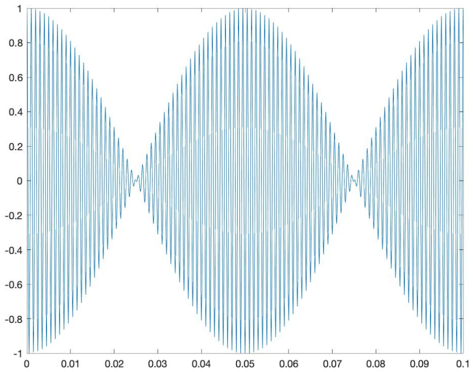
## Part 4:

$$x_4(t) = \cos(2\pi f_1 t)\cos(2\pi f_2 t)$$

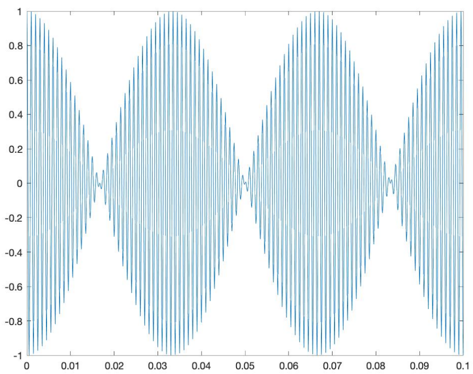
**$f_1$ = 10Hz:**




**$f_1$ = 5Hz:**



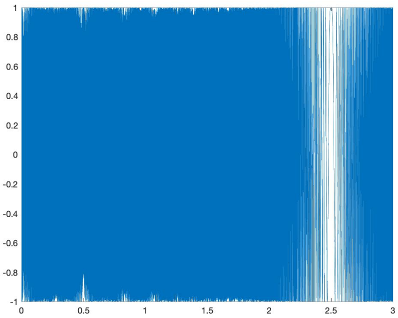

**$f_1$ = 15Hz:**



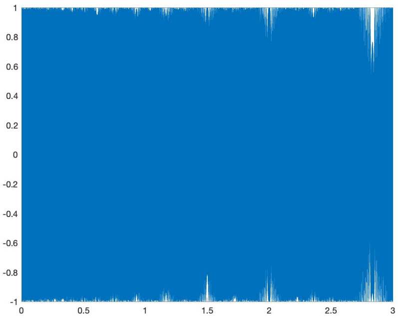
## Part 5:

$$x_5(t) = \cos(2\pi\mu t^2 + 2\pi f_o t + \varphi)$$

2500Hz to 500Hz:



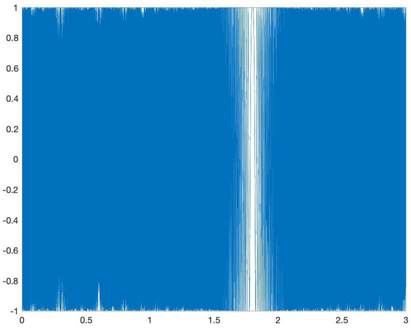
500Hz to 2500Hz:



The second chirp is different from the first chirp because instead of the spaced-out sound being played at the end (the drop in sound), the drop in waveform is relatively shorter and hence the sound observed appears to be more uniform as compared to the first waveform.

When the value of μ is halved, the drop in sound waves goes later in the waveform. When the value of μ is doubled, the drop in sound waves goes earlier and is more spaced out.

Chirp Puzzle:



The chirp waveform (amplitude) of the chirp goes down and then goes up as time progresses. According to the theory of frequency spectrum, the frequency decreases in the first half of the chirp and increases in the second half. Hence, the frequency spectrum varies.

## Part 6:

**Song notes: song = ["A", "B", "D#", "E", "C", "F#", "E", "E", "D", "G", "C#", "C#", "A", "B", "A", "A"];**

**Code:**

```
notename = ["A", "A#", "B", "C", "C#", "D", "D#", "E", "F", "F#", "G", "G#"];
song = ["A", "B", "D#", "E", "C", "F#", "E", "E", "D", "G", "C#", "C#", "A", "B",
"A", "A"];
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)

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

## Codes:

```
func1();
func2();
func3();
func4();
func5();
func6();
func7();

notename = ["A", "A#", "B", "C", "C#", "D", "D#", "E", "F", "F#", "G",
"G#"];
song = ["A", "B", "D#", "E", "C", "F#", "E", "E", "D", "G", "C#", "C#",
"A", "B", "A", "A"];
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)

% Part 1
function [x1] = func1()
    f = 440;
    t = [0:0.0001:3.0];
    x1 = sin(2 * pi * f * t);
    soundsc(x1);

    plot( t, x1);
    xlim([0 0.01]);
end

function [s] = func2()
    t = [0:0.0001:3.0];
    s = sin(2 * pi * 440 * t) + sin(2 * pi * 554 * t) + sin(2 * pi * 659 *
t);
    soundsc(s);

    plot( t, s);
    xlim([0 0.01]);
end

%Part 2
function [s] = func3()
    phase = pi;
    f = 587;

    t = [0:0.0001:3.0];
    s = cos((2 * pi * f * t) + phase);
    soundsc(s);

    plot( t, s);
    xlim([0 0.01]);
end
```

```matlab
%Part3
function [x3] = func4()
    a = 3;
    f0 = 440;
    t = [0:0.0001:3.0];
    x3 = exp(-t * (a.^2+2)) .* (cos( 2 * pi * f0 * t));
    soundsc(x3);

    plot(t, x3);
end

%Part 4
function [x4] = func5()
    f1 = 5;
    f2 = 1000;
    t = [0:0.0001:3.0];
    x4 = cos( 2 * pi * f1 * t) .* cos( 2 * pi * f2 * t);
    soundsc(x4);

    plot(t,x4);
    xlim([0 0.1]);
end

%Part 5
function [x5] = func6()
    mu = -500;
    phi = 500;
    t = [0:0.0001:3.0];
    f0 = 2500;
    x5 = cos(2 * pi * mu * t .^ 2 + 2 * pi * f0 .* t + phi);

    soundsc(x5);
    plot(t,x5);
%     xlim([0 0.01]);
end

function [x5] = func7()
    mu = 2500/3;
    phi = -2000;
    t = [0:0.0001:3.0];
    f0 = 3000;
    x5 = cos(2 * pi * mu * t .^ 2 + 2 * pi * f0 .* t + phi);
    soundsc(x5);

    plot(t,x5);
end


%Part 6
function [note] = notecreate(frq_no, dur)
    note = sin(2*pi*[1:dur]/8192*(440*2.^((frq_no-1)/12)));
end
```