M. Enes İnanç

22001799

09.02.2024

Part 1:

As frequency goes up the pitch also goes up. High frequency signals are related to the high pitched sound. Our ears detects the high frequency signal as a shrill voice. As can be seen in the figures 1,2,3; as frequency goes up, the waves close to each other.

These figures were plotted for x1(t) = sin (2πfot).

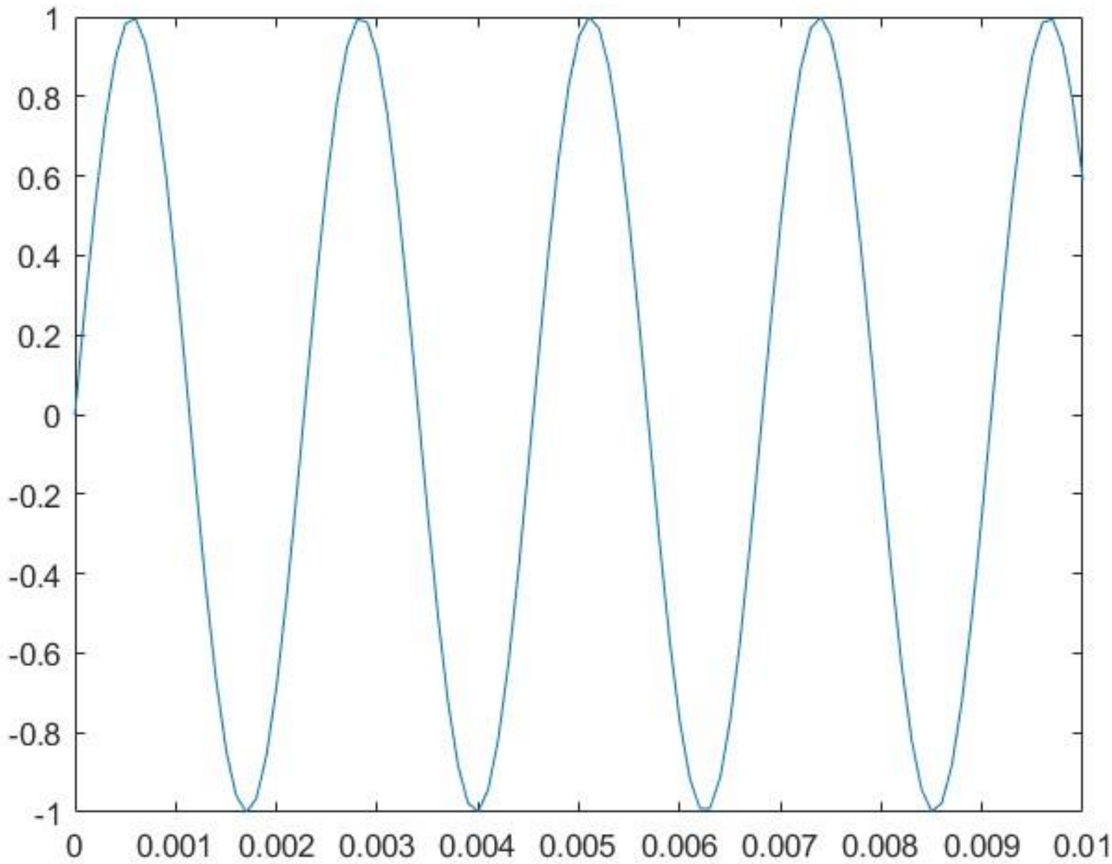

Figure 1 When f0 = 440 Hz

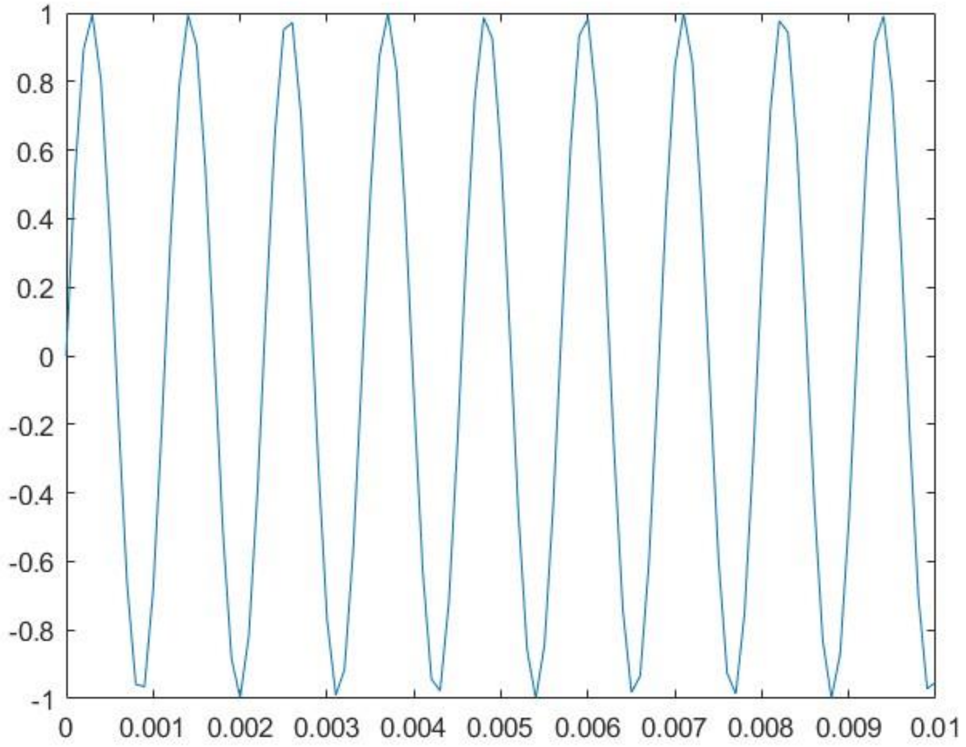

Figure 2 When f0 = 880 Hz

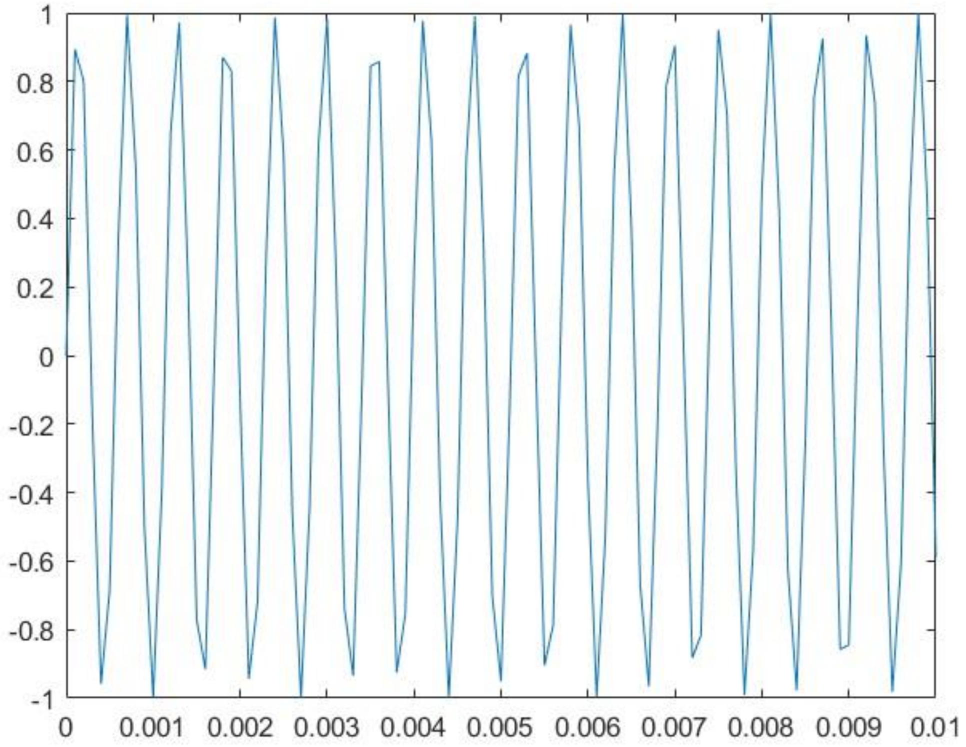

Figure 3 When f0 = 1760 Hz

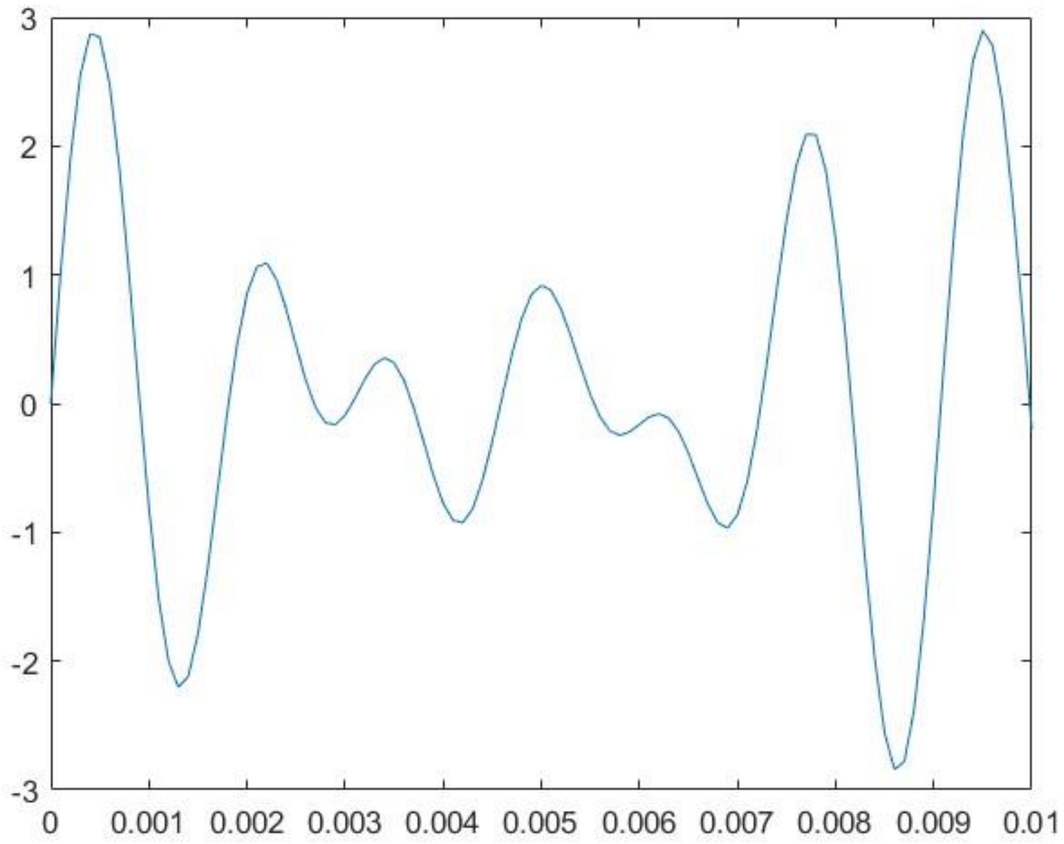

Figure 4 $s(t) = \sin(2\pi 440\ t) + \sin(2\pi 554\ t) + \sin(2\pi 659\ t)$

The major triad is a basic building block in Western music harmony and is commonly associated with a bright, consonant, and pleasing sound. This combination of frequencies is a key element in creating a major chord, contributing to the harmonic richness and musicality of the signal.

Part2:

a) In the second part, we investigate the impact of phasing on a cosine signal with a frequency of 587 Hz. Initially, the phase angle (φ) is set to 0.

$$x_2(t) = \cos(2\pi f_0 t + \varphi)$$

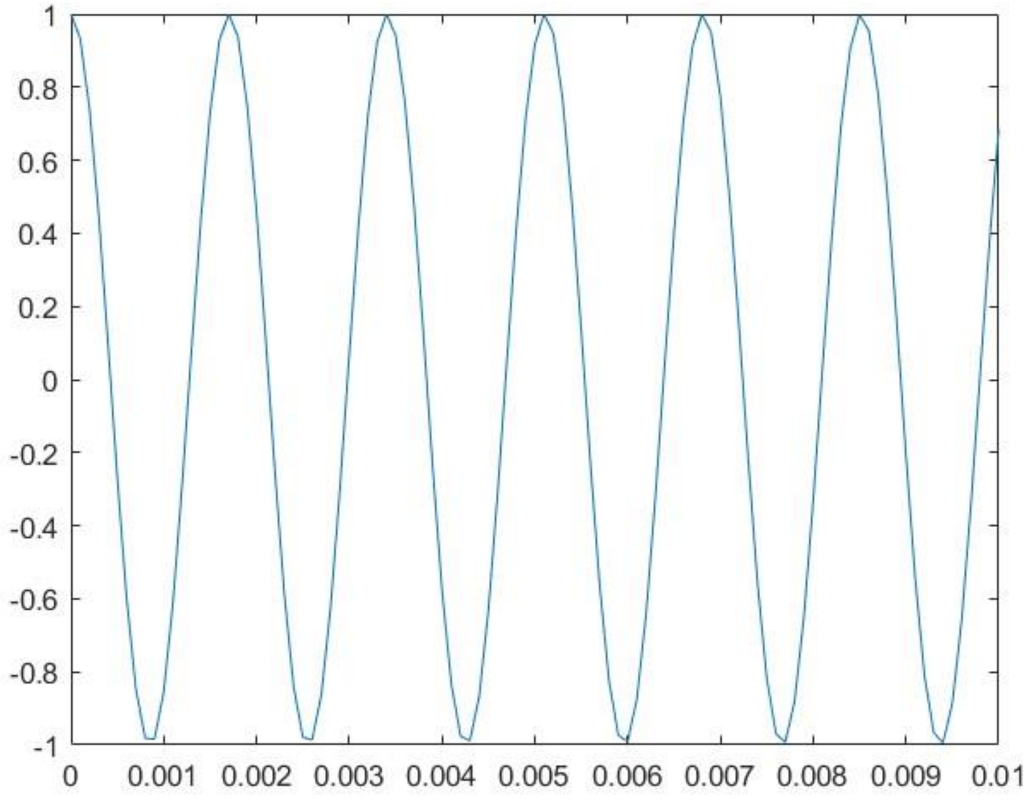

Figure 5 when φ = 0

b) Afterward, φ has been set successively to π/4, π/2, 3π/4, and π.

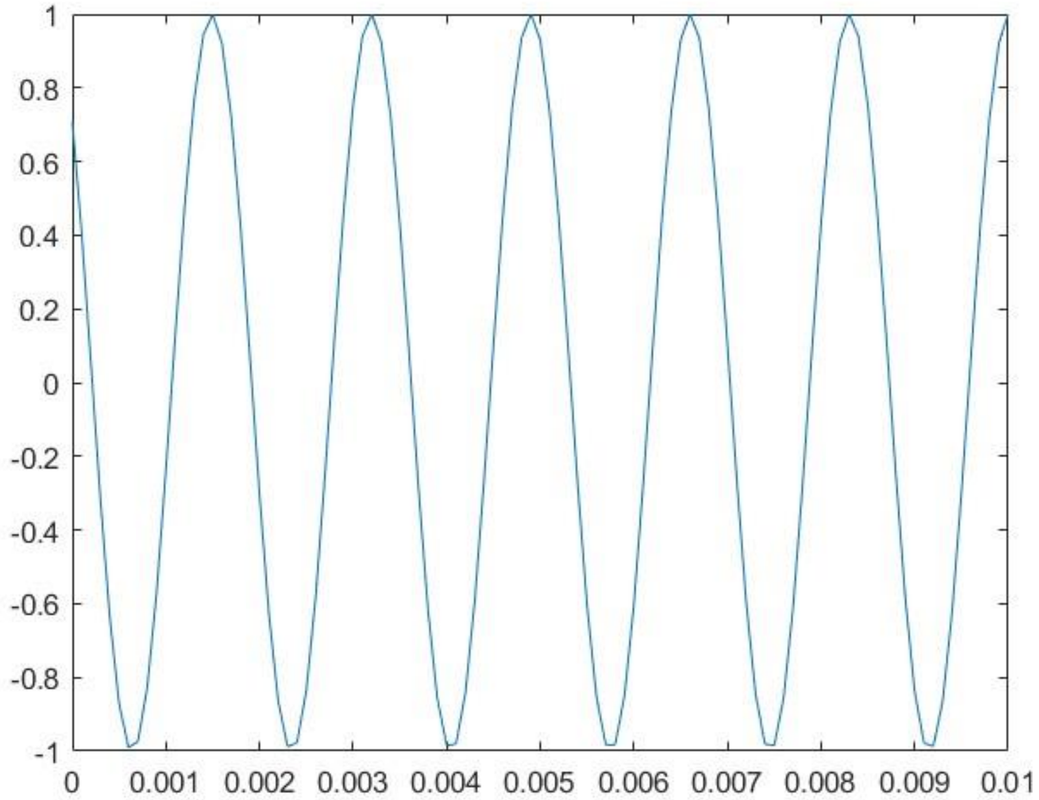

Figure 6 when φ = π/4

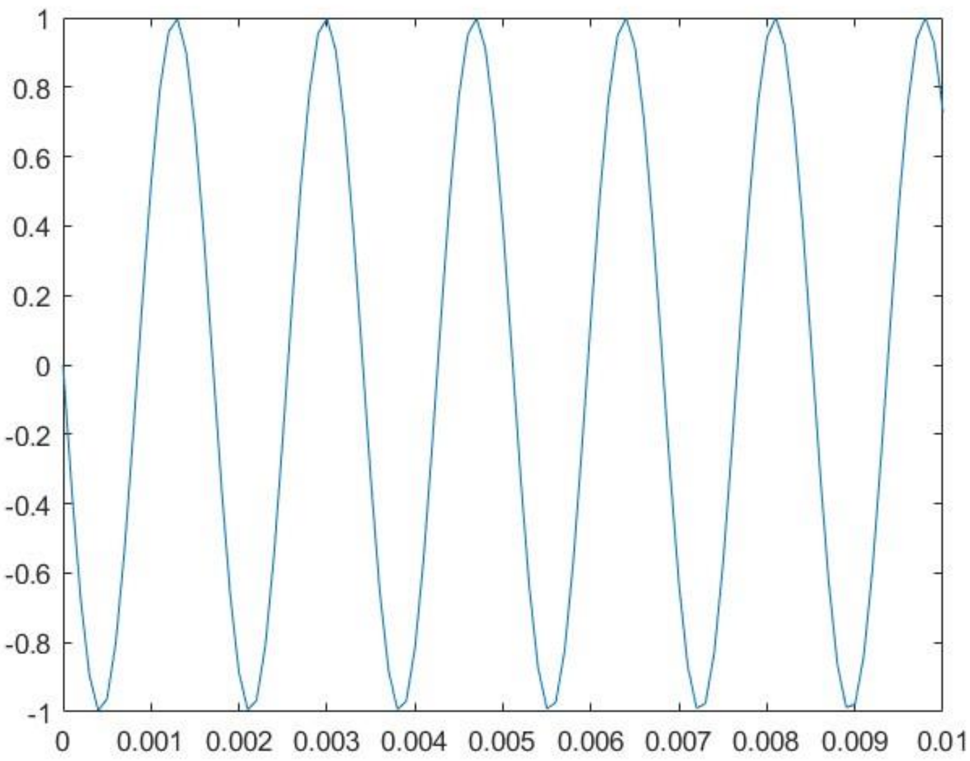

Figure 7 when φ = π/2

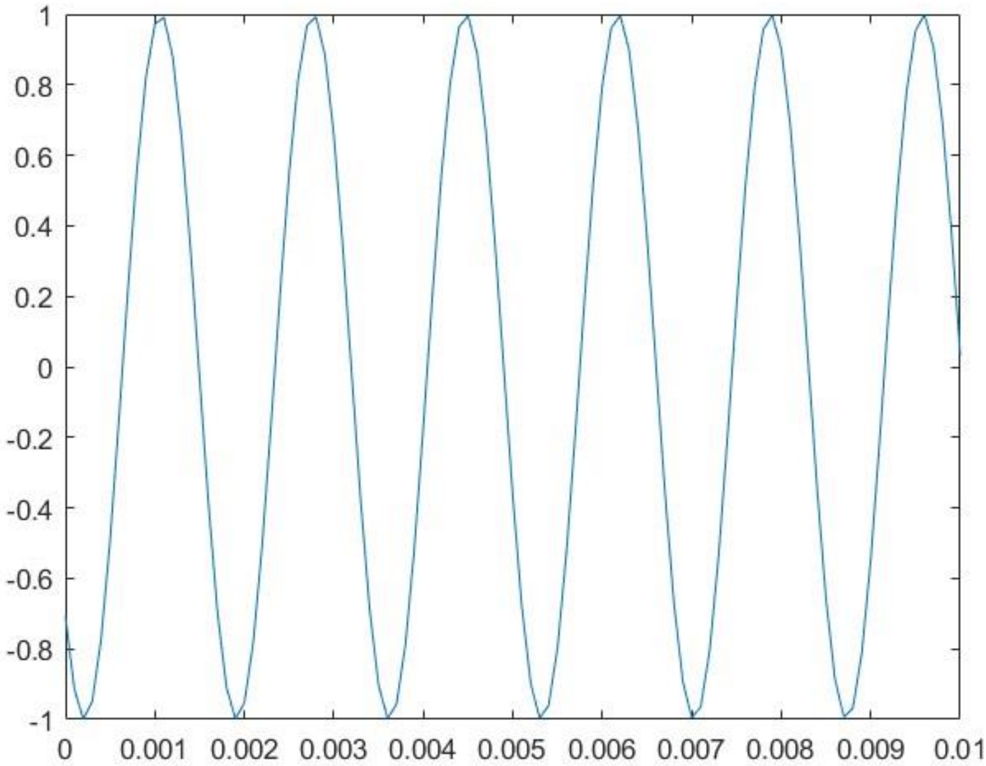

Figure 8 when φ = 3π/4

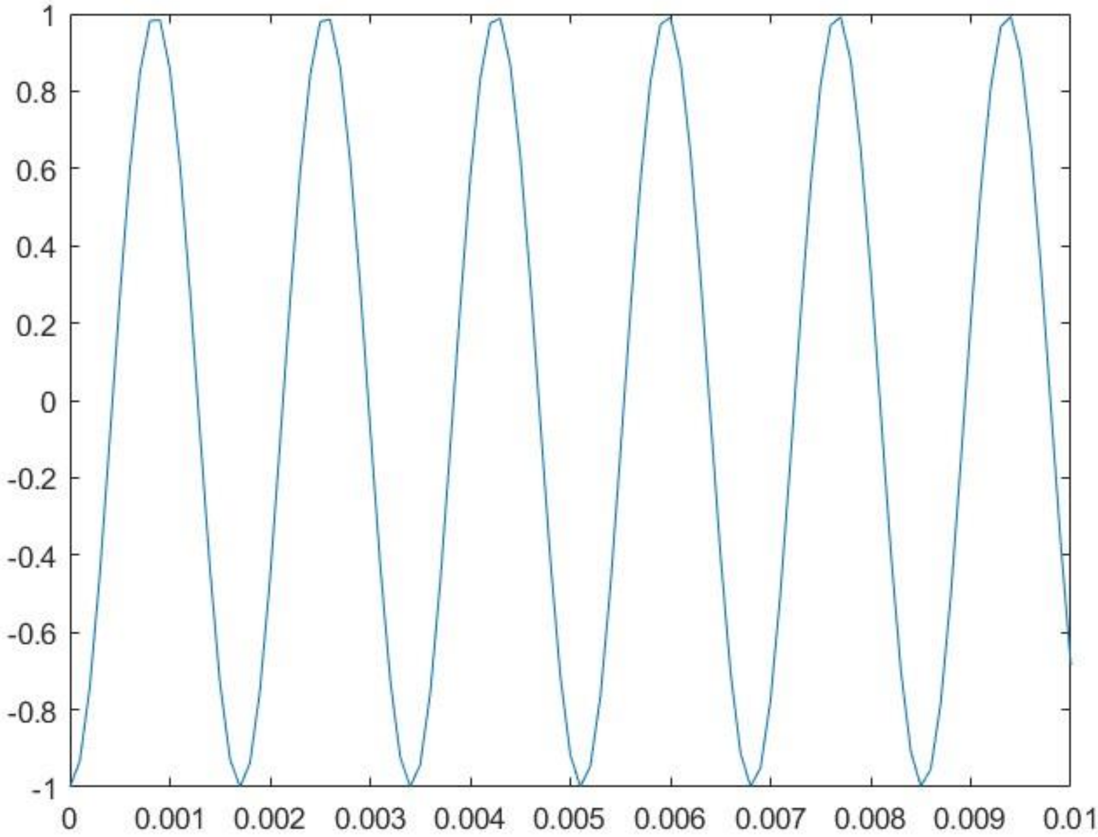

Figure 9 when $\varphi = \pi$

The human ear perceives no discernible alteration. The rapidity of the cosine signal's phase is such that it reaches our ears as a consistent sound. The pitch of the sound I hear remains constant, unaffected by the phase.

Part 3:

The x3(t) signal, unlike the x1(t) signal, is a decaying signal over time. While the x1(t) signal sustains indefinitely, the x3(t) signal is composed of a cosine signal multiplied element-wise by the term e^−(a^2+2)t. Since the term e^−(a^2+2)t represents a decaying signal, the element-wise multiplication results in x3(t) also being a decaying signal over time.

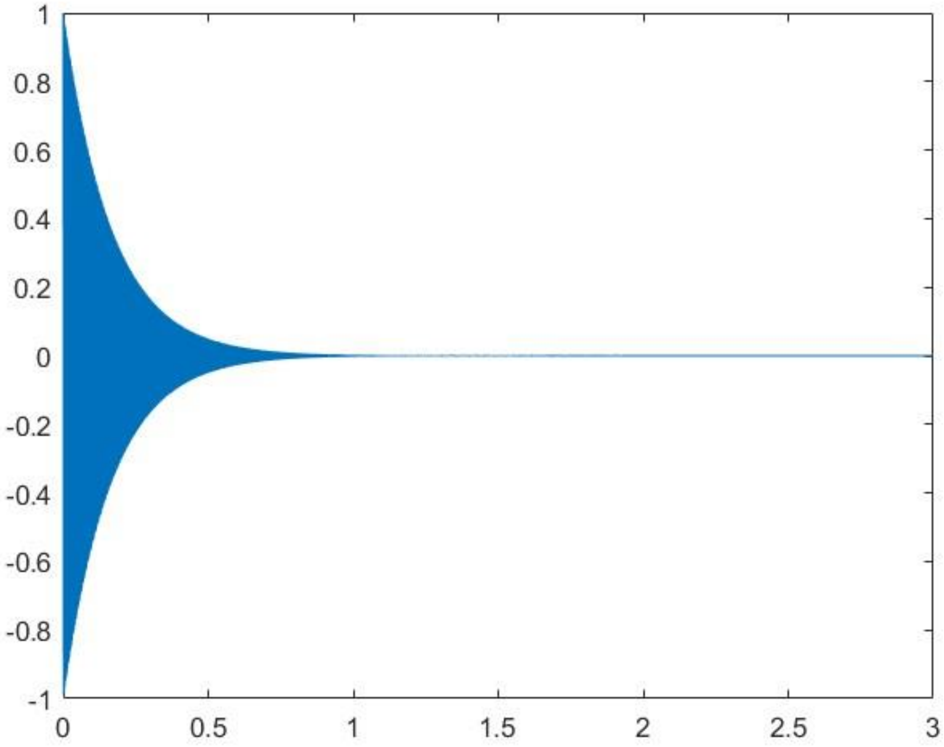

Figure 10 a = 2

As 'a' increases, the sound duration of x3(t) decreases due to a faster decay. Higher 'a' values result in more rapid damping and shorter audible signals. However, the timbre of the sound remains unchanged. The variation in 'a' only affects the decay duration of the signal.

The mass-spring system with damping is a good example of the real-world application of these systems. In this system, the motion is damped due to environmental effects. Additionally, since x1 has a fuller and damped sound, it resembles the sound of a piano more.

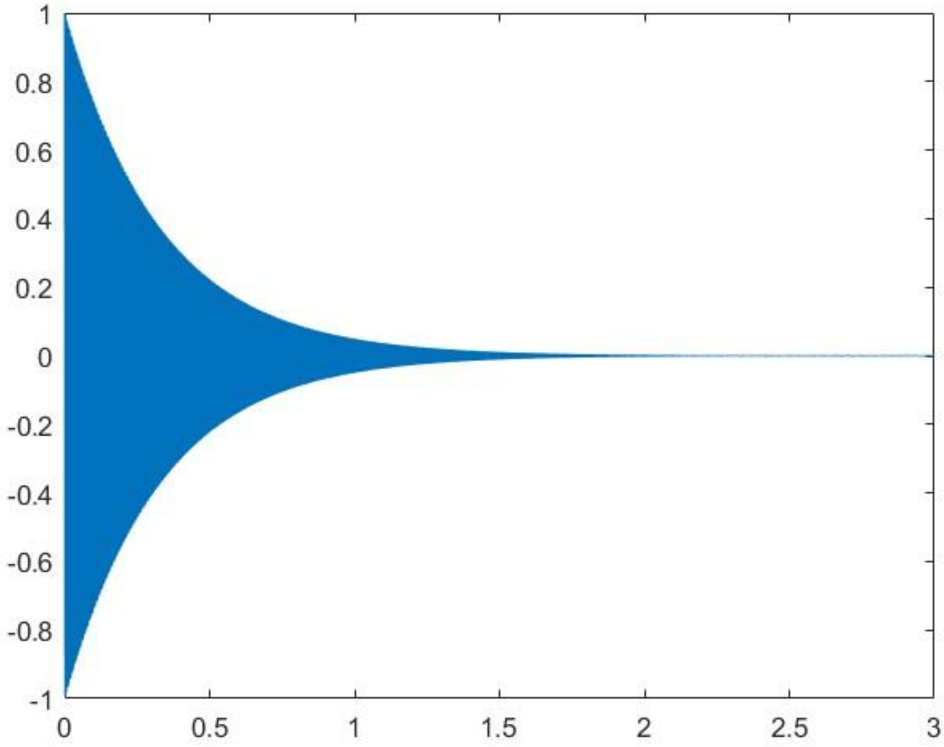

Figure 11 a =1

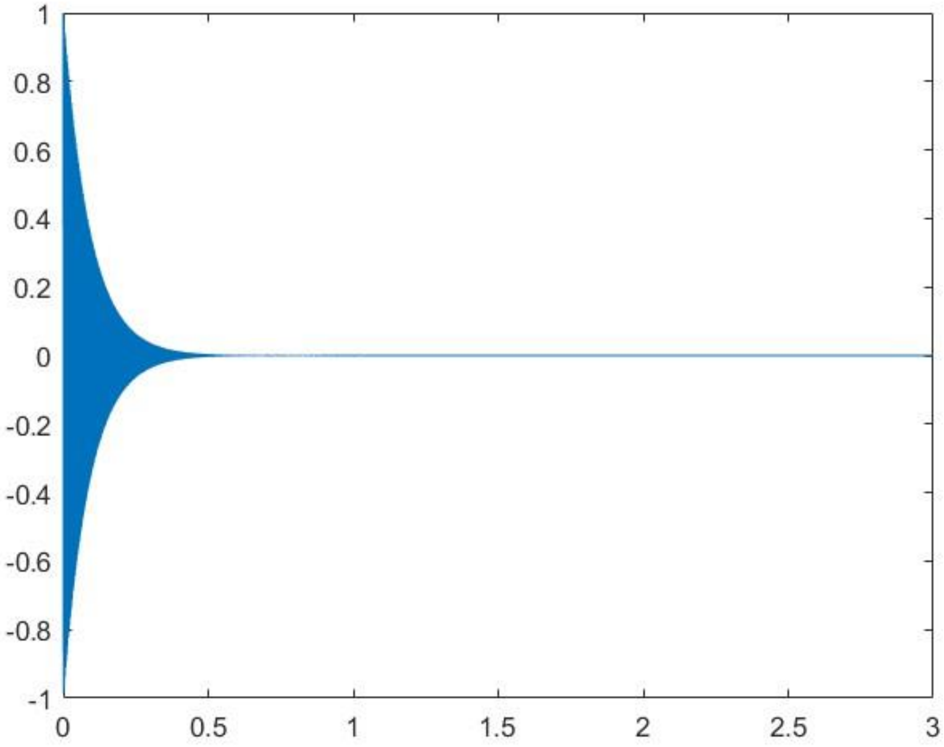

Figure 12 a =3

Part 4:

In the fourth section, our aim was to generate a phenomenon known as a beat note. This effect is most pronounced when one frequency is set significantly low (10 Hz) and the other is around 1 kHz, achieved through the multiplication of two sinusoidal waves.

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

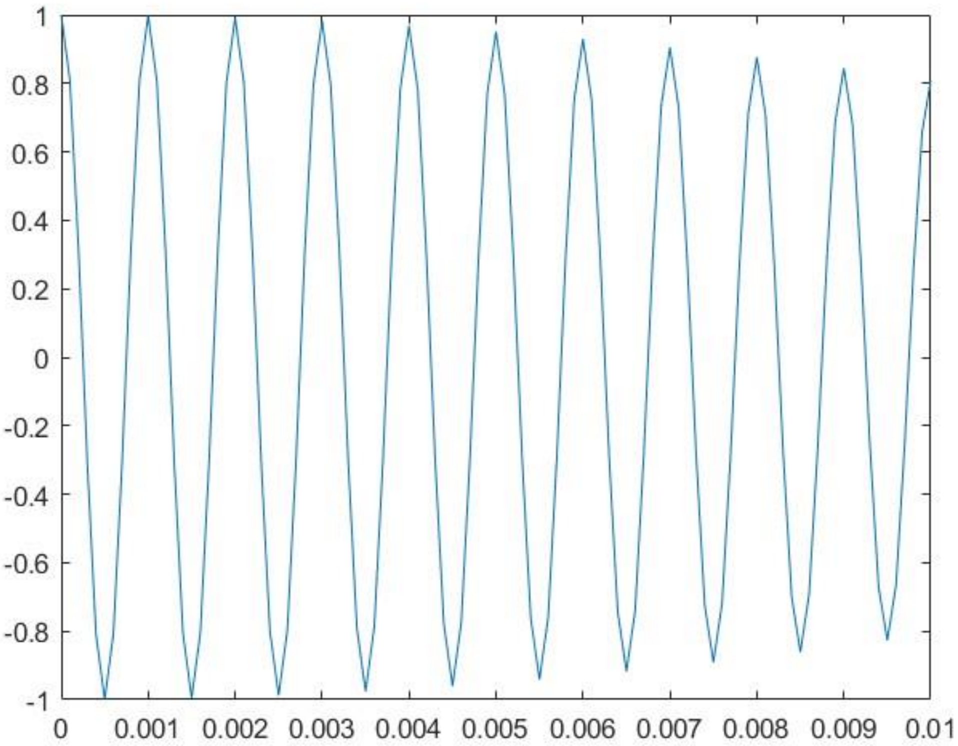

Figure 13 when f1=10Hz f2=1000Hz

Then we generate this signal by changing f1 to 5Hz and 15Hz.

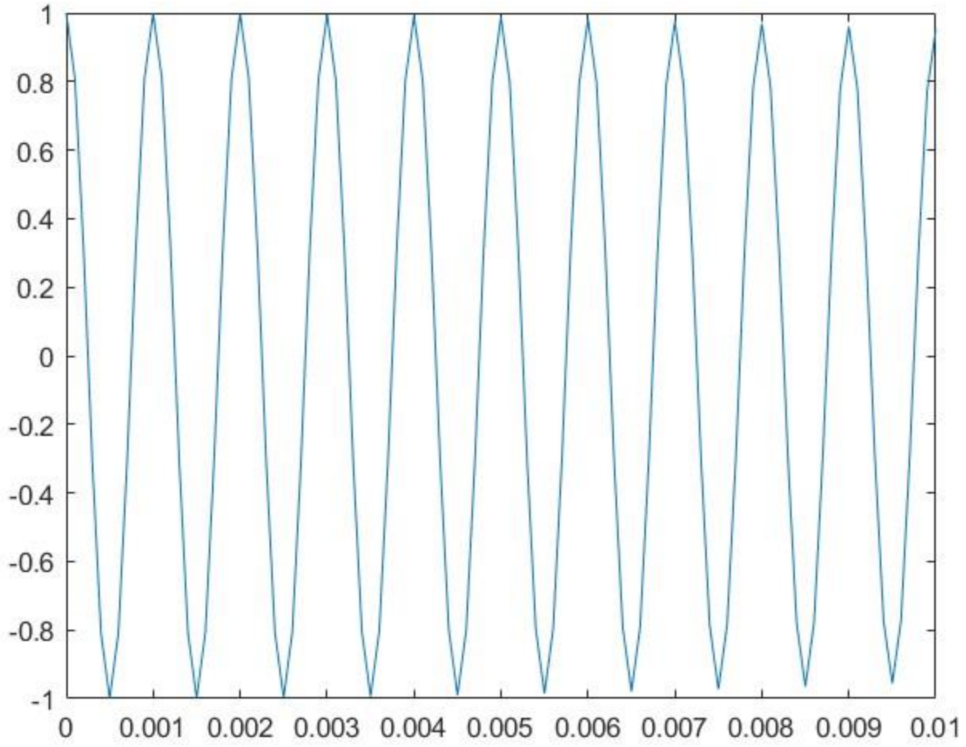

Figure 14 when f1=5Hz f2=1000Hz

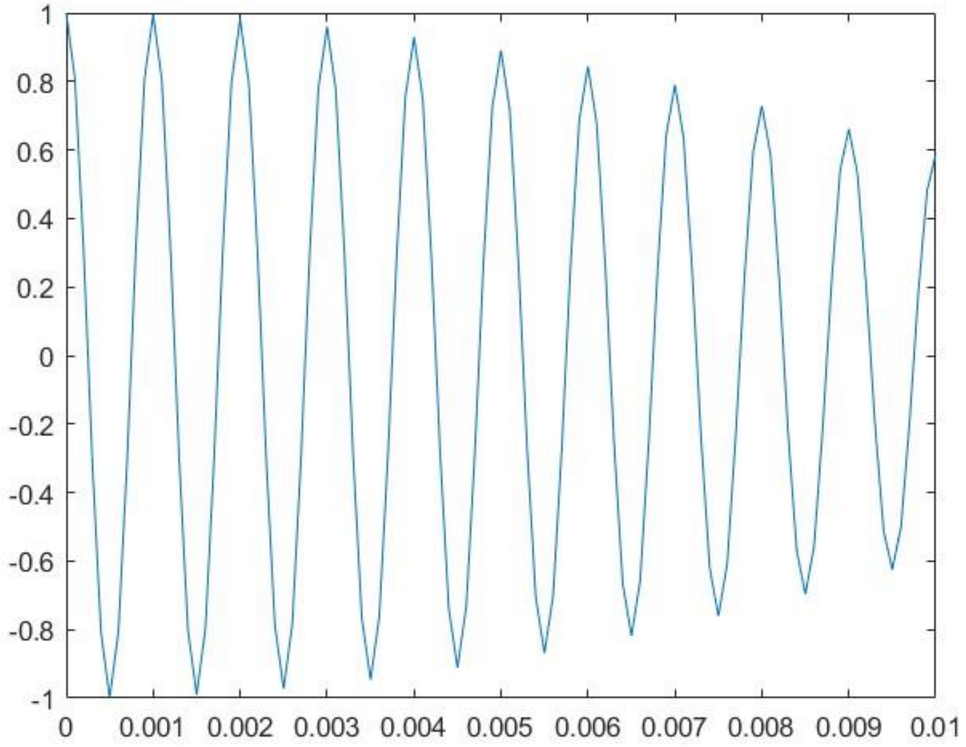

Figure 15 when f1=15Hz f2=1000Hz

As f1 decreases, as can be understood from the given formula, the frequency of the resulting new signal also decreases. This results in a more stable sound. As f1 decreases, a signal that is more easily perceivable by humans is formed. However, when compared to the x1 signal, the frequency is still high, and the x4 signal resembles the sound of a flute.

$$\cos(\theta_1)\cos(\theta_2) = \frac{1}{2}\left[\cos(\theta_1 + \theta_2) + \cos(\theta_1 - \theta_2)\right].$$

Part 5:

Chirp signals are signals in which the frequency changes linearly over time. The μ coefficient represents the rate of change of frequency in a chirp signal. The frequency of the first chirp signal decreases linearly, while the frequency of the second chirp signal increases linearly. As a result, when listening to the second signal, we notice that the sound becomes progressively higher in pitch, whereas when listening to the first signal, the sound becomes progressively lower in pitch. When the μ coefficient is doubled or halved, the rate of change of the signal's frequency is affected. The human ear perceives a faster thinning or thickening of the sound when the μ coefficient is doubled and a slower change when it is halved.

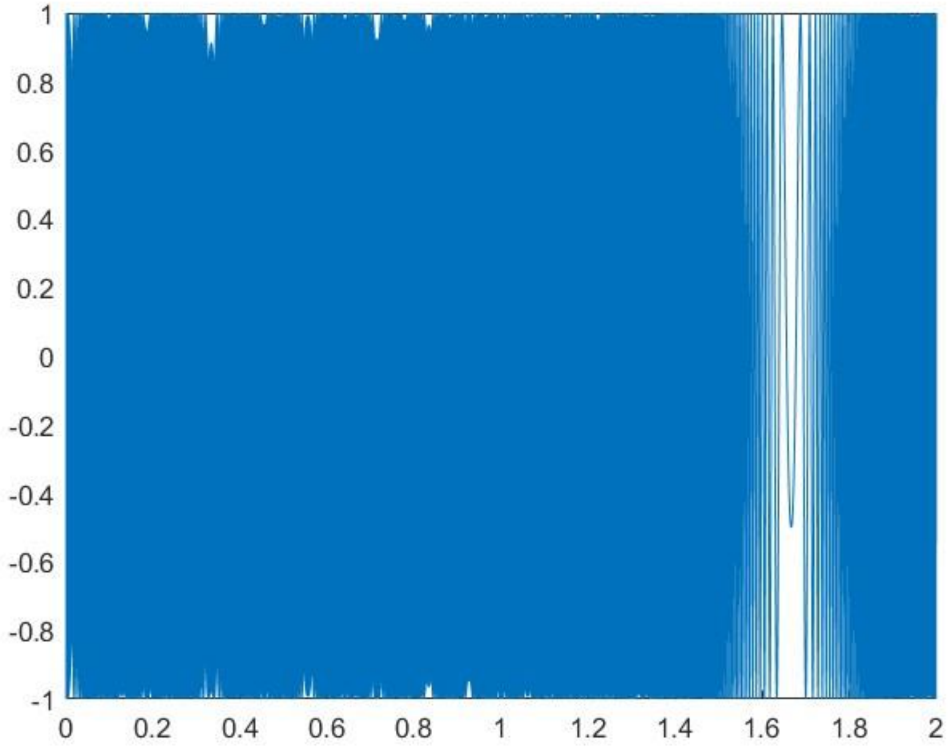

Figure 16 Goes Lower Pitch

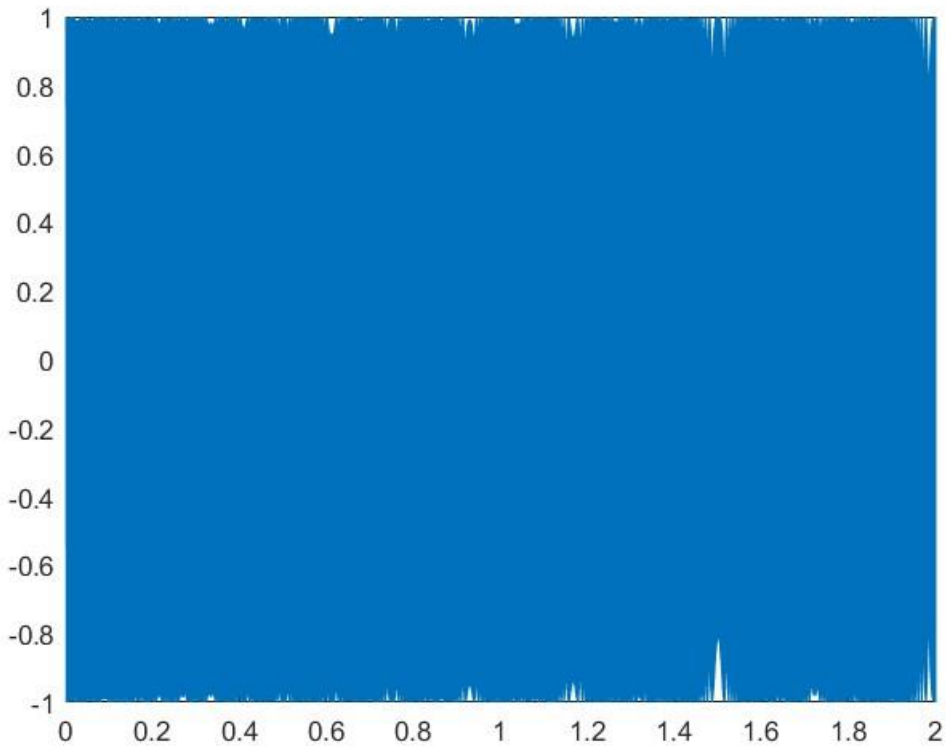

Figure 17 Goes Higher Pitch

Part 6:

In the 6th part, the code was already provided to us. Research was conducted on how the provided code and notes were generated, and the logic was thoroughly understood. In this part, I also created my own song, and a very enjoyable song emerged.

Appendices:

Codes:

Part 1:

```
 t = [0:0.0001:3]
f0 = 440
x1n = sin(2*pi*f0*t)
t1 = [0:0.0001:0.01]
x1 = sin(2*pi*f0*t1)
plot(t1,x1)
soundsc(x1n)
```

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

sn = sin(2*pi*440*t1) + sin(2*pi*554*t1) + sin(2*pi*659*t1)

plot(t1,sn)

soundsc(s)
```

Part 2:

```
t = [0:0.0001:3]
f0 = 587
fi = 0
x1n = cos(2*pi*f0*t +fi)
t1 = [0:0.0001:0.01]
x1 = cos(2*pi*f0*t1 +fi)
figure;
plot(t1,x1)
soundsc(x1n)
```

Part 3:

```
t = [0:0.0001:3]
f0 = 440
a = 2
x3n = (exp(-(a*a+2)*t)) .* cos(2*pi*f0*t)
t1 = [0:0.0001:0.01]
x3 = (exp(-(a*a+2)*t1)) .* cos(2*pi*f0*t1)
plot(t,x3n)
soundsc(x3n)
```

Part 4:

```
t = [0:0.0001:3]
f1 =5
f2 =15
x4 = cos(2*pi*f1*t) .* cos(2*pi*f2*t)
t1 = [0:0.0001:0.01]
x4n = cos(2*pi*f1*t1) .* cos(2*pi*f2*t1)
plot(t1,x4n)
soundsc(x4)
```

Part 5:

```
t = [0:0.0001:2]
f0 = 2500
u = -750
x5 = cos(2*pi*u.*t.*t + 2*pi*f0.*t )
soundsc(x5)
figure;
plot(t,x5)



f1 = 500
u1 = 500
x6 = cos(2*pi*u1.*t.*t + 2*pi*f1.*t )
soundsc(x6)
figure;
plot(t,x6)
```

Part 6:

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