Bilkent University

EE 391

MatLab Assignment 1

03/03/20

Okan Şen

21202377

Section 2

NOTE:

I have written the code in such a way that it executes all parts on after another with “z” seconds of waits between each part, and “b” seconds of waits within the internal sections of parts(such as; changing the value of “a”, “phi”, etc…).

These “z” and “b” values can be changed at the beginning of the code.

The plots are printed on the screen in a maximized window and all the parts plots draw on top of each other so that they can be compared. At the beginning of each new part, the older part’s plots are erased.

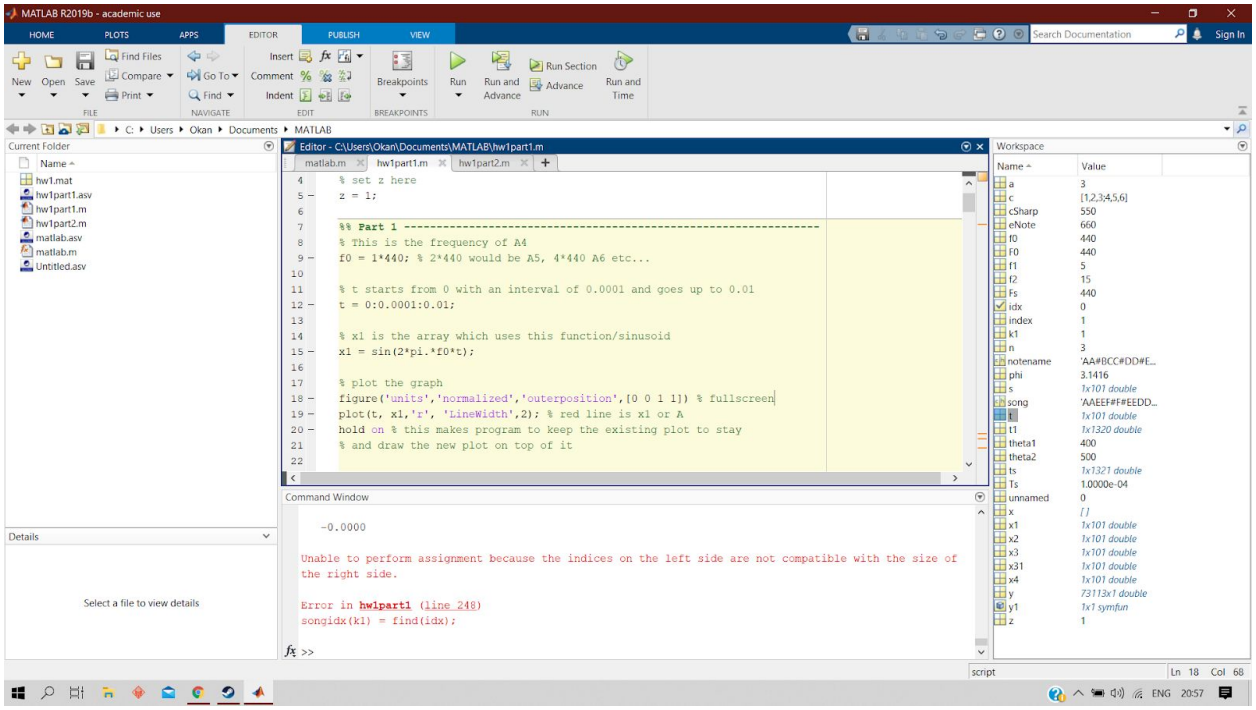
The assignment tells to print plots at a duration of 0.01 seconds, but upon hearing very short sounds I took the liberty of expanding the duration to 3 seconds. I will attach each plot’s detailed version under every plot posted in this report. Above plots will showcase 3 seconds waves, below plots will show 0.01 duration waves.

For testing, one of these methods can be used,

1) Uncomment “t = 0:0.0001:0.01;” and comment “t = linspace(0, 24576, 8192)/Fs;” for 0.01 durations
2) Keep it as it is for 3 second durations which don’t overlap.
3) Uncomment “t = 0:0.0001:0.01;” and and change 0.01 to 3 on the right most place, and comment “t = linspace(0, 24576, 8192)/Fs;” for 3 durations which overlap sounds, but have more timbre variation.

Part 1

Screenshot as asked in assignment(evidence):



At first, I used this code line to traverse through every 0.0001 seconds within 0 to 3 seconds;

t = 0:0.0001:0.01;

But upon getting not very different results with 1-millisecond duration note plays, I changed the line to this;

t = linspace(0, 24576, 8192)/Fs;

This actually made the notes play out longer and, now they had much more personalized sounds. The former code created simple curves in the plots, while the latter created actual soundwaves.

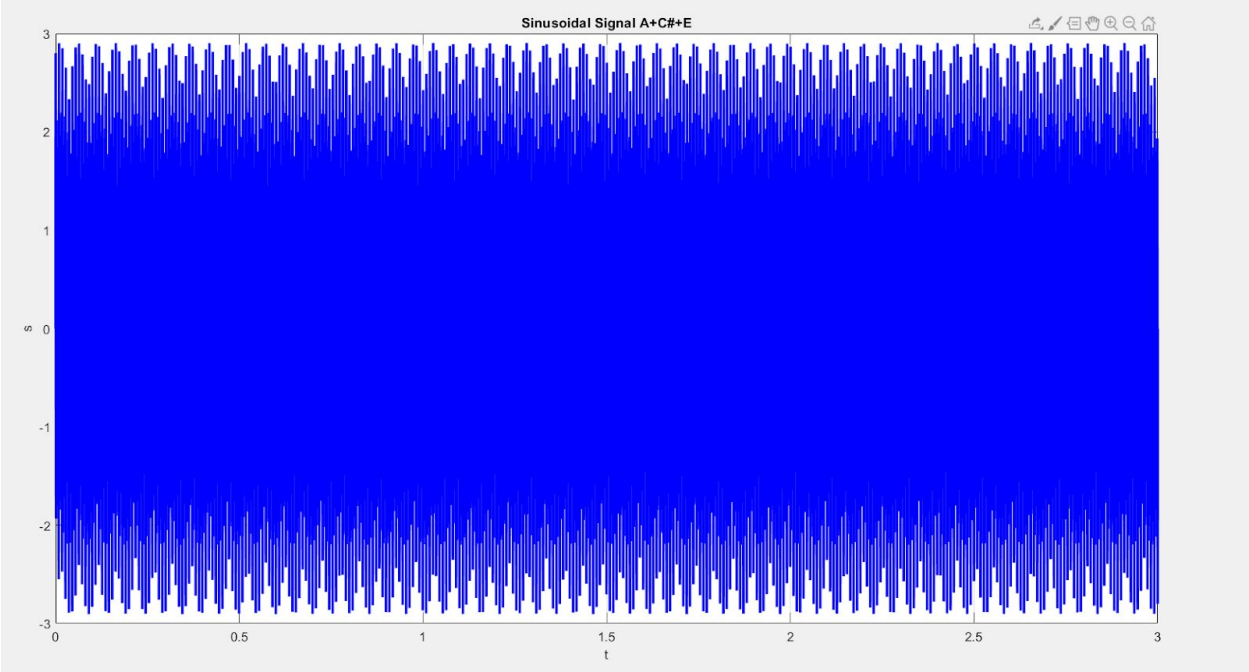
I tried changing 0.01 to 3 seconds in this line too;

t = 0:0.0001:0.01; |-> t = 0:0.0001:3;

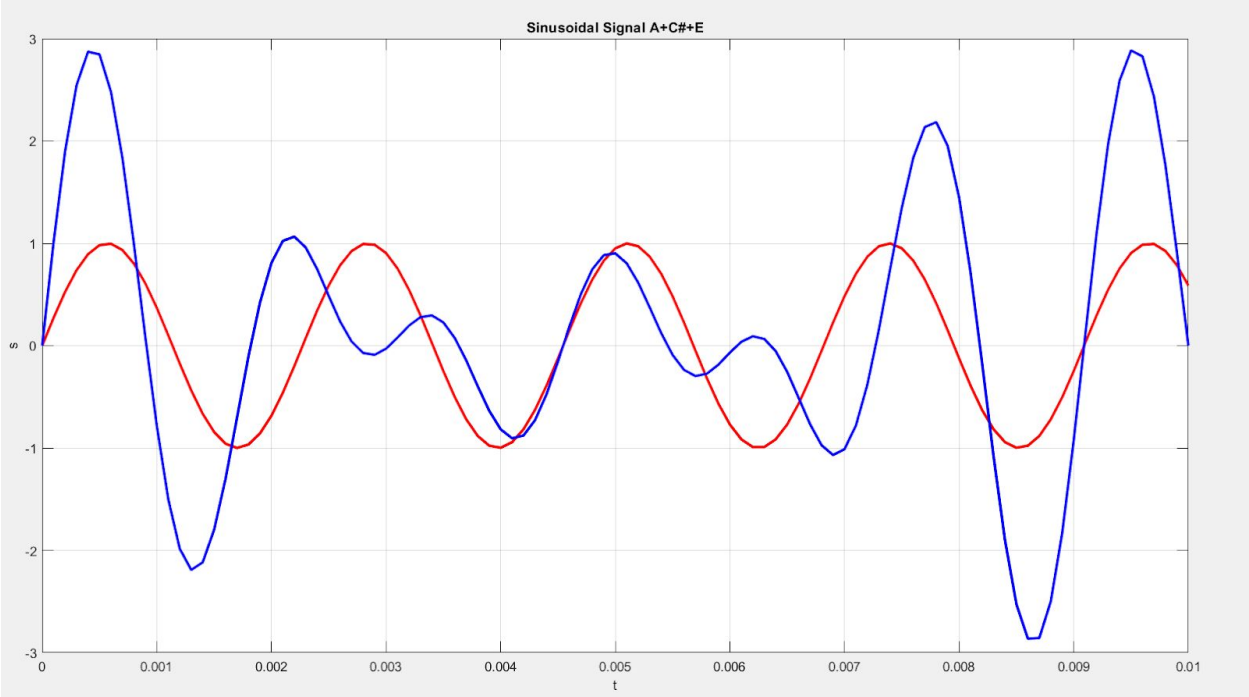
And the results showed more characterised sounds compared to using linspace method. I was unsure which method to use but the sounds overlapped so I went with linspace for a more organized work.

“What happens to the pitch of the sound as
the frequency increases?”

As expected, as f0 grows higher the frequency goes higher which results in the note getting higher in pitch as well. (2*f0, 8*f0)



3 second duration waves part 1



0.01 second duration waves part 1

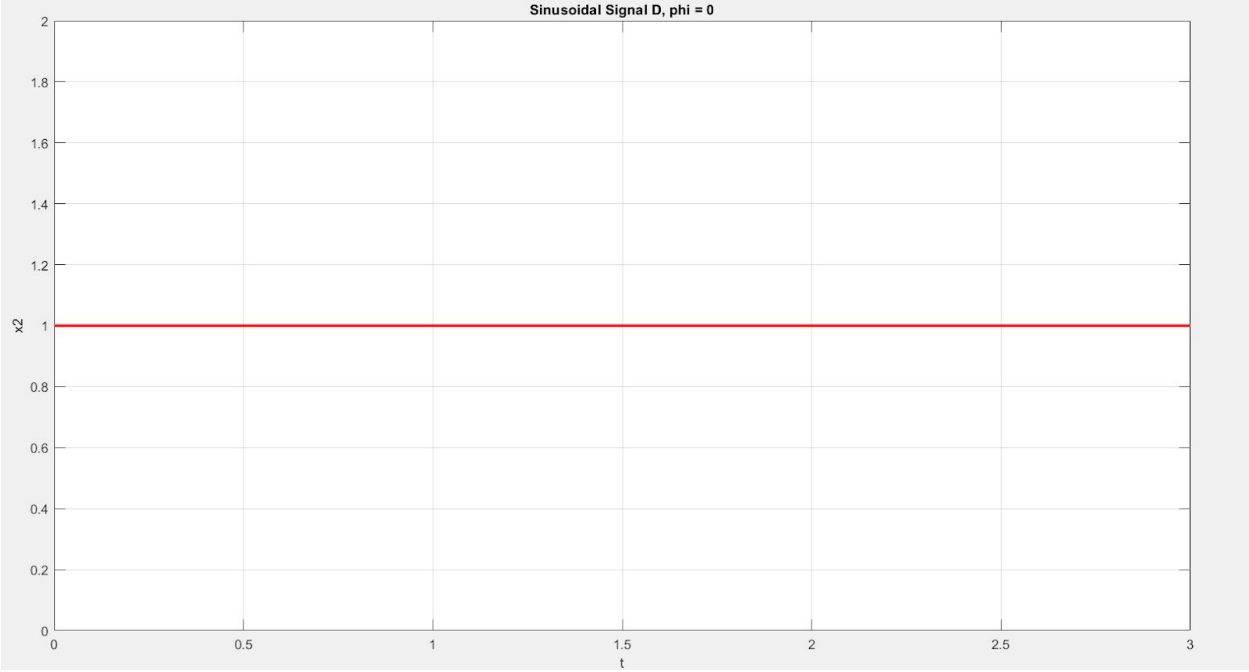
In my code, A major triad is played as a chord. A, C# and E are played at the same time after A4(440 f0) is played before that.

"s(t) = sin(2 π 440 t) + sin(2 π 554 t) + sin(2 π 659 t)"

The special thing about the function is, rather than a single note, we now hear a three-note chord which is A major triad.

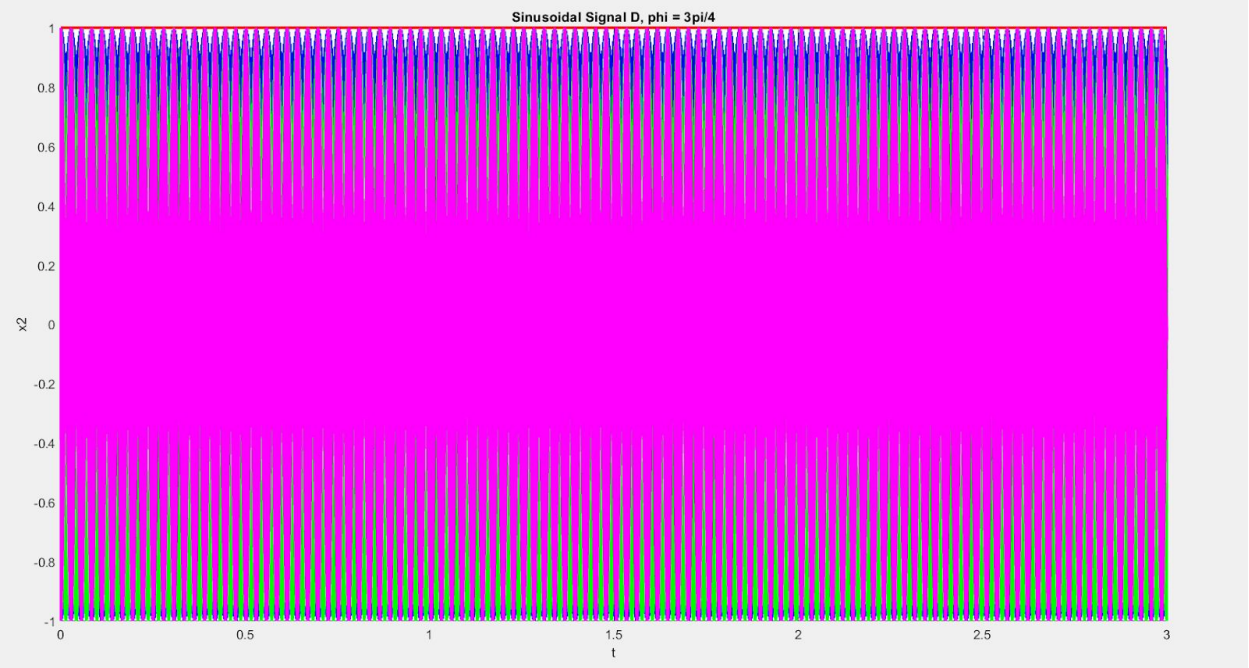
Part 2

a) First settings give a result such as this, which does not have enough vibe in that creates a sound. It is a one shot sound.

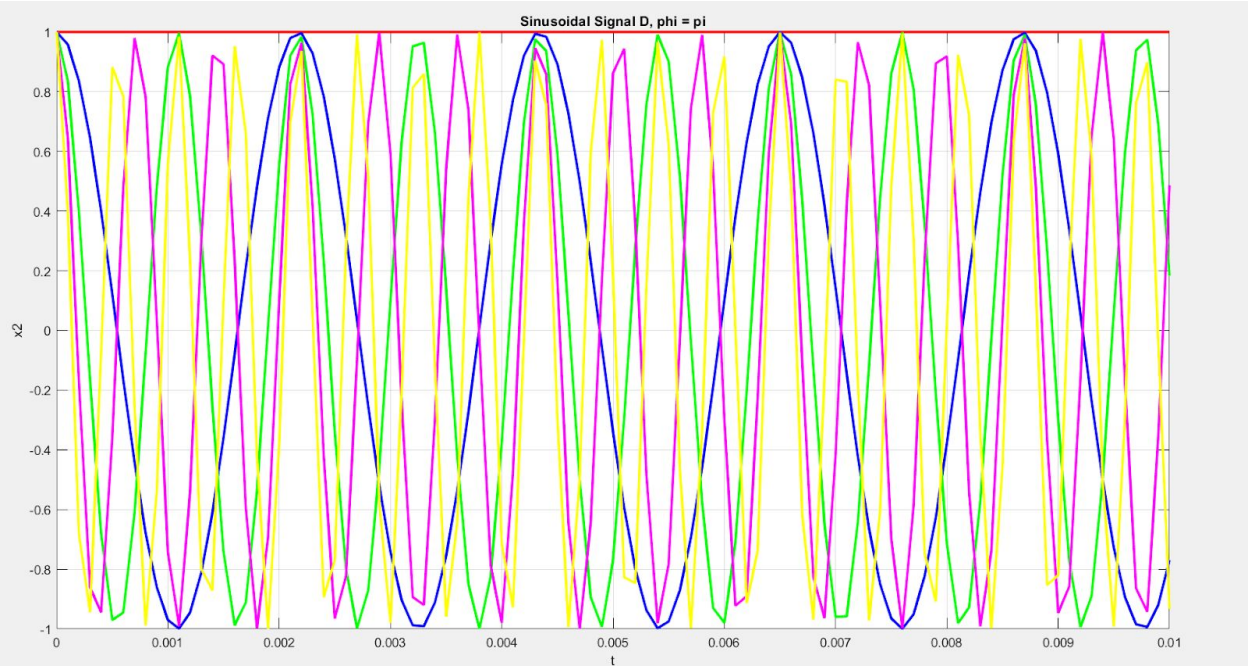


0.01 and 3 second durations are same for this particular plot

b) Changing the phi value gives interesting results by creating distinctively different sounds.



3 second duration waves part 2



0.01 second duration waves part 2

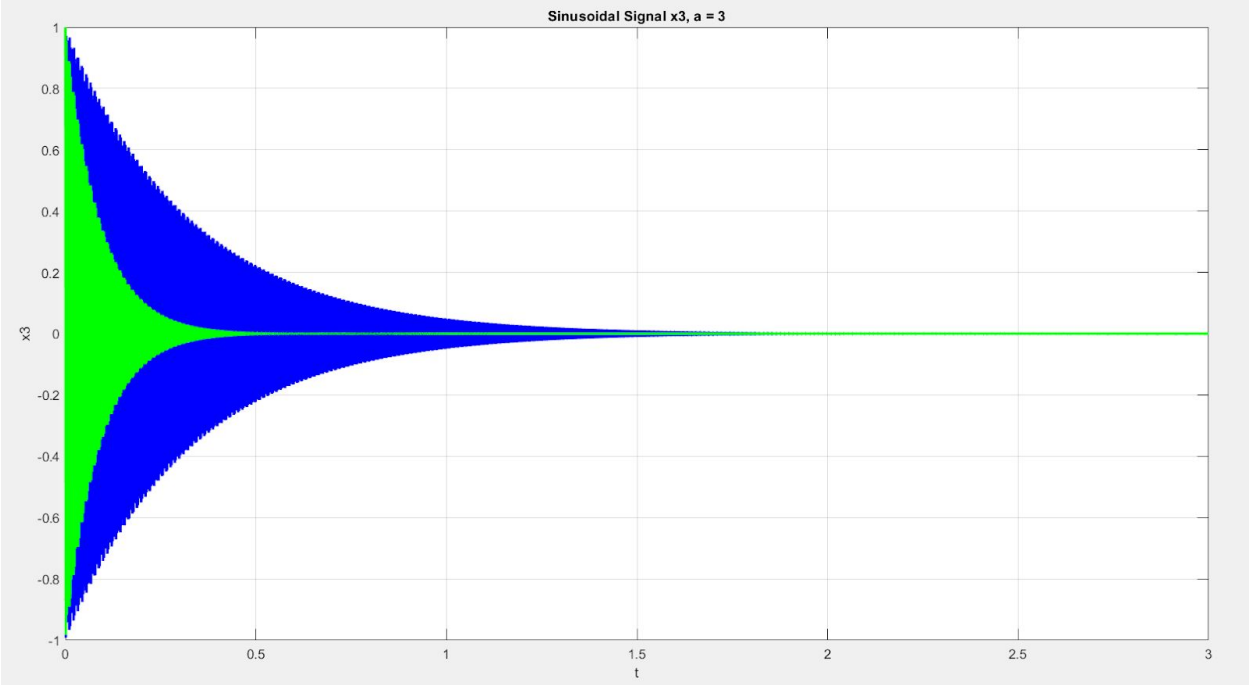
The plot became a lot active. As the phi value went through pi/4, pi/2, 3pi/2, and pi radians, the sound became much higher as the resulting frequency was increasing. They become more like soundwaves.

Part 3

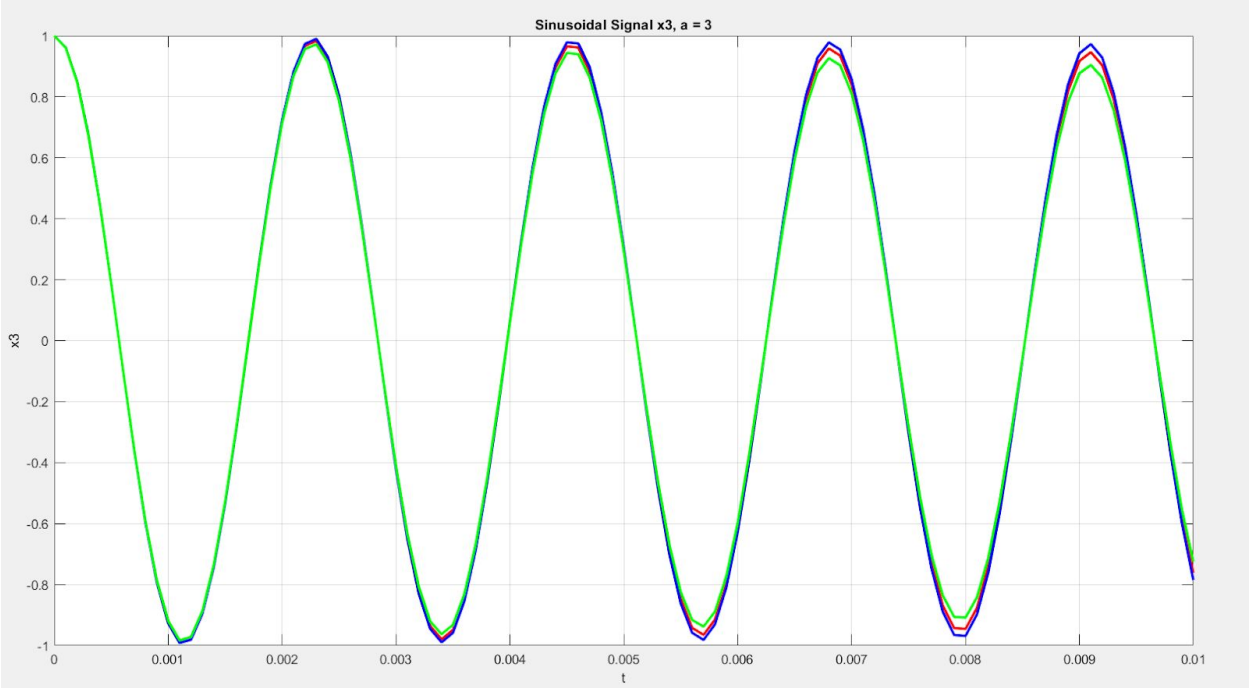
```
x3 = (exp(1).^(-(a.^(2) + 2).*t) ).* (cos(2*pi.*f0.*t));
```

This is the formula that was wanted to be in the report for part 3.

"What is the effect of including the exponential term to the sound that you Hear?"



3 second duration waves part 3



0.01 second duration waves part 3

The exponential term made the frequency to have a curve rather than going in a constant direction. The sound dropped to low frequencies so fast it became snuffed out.

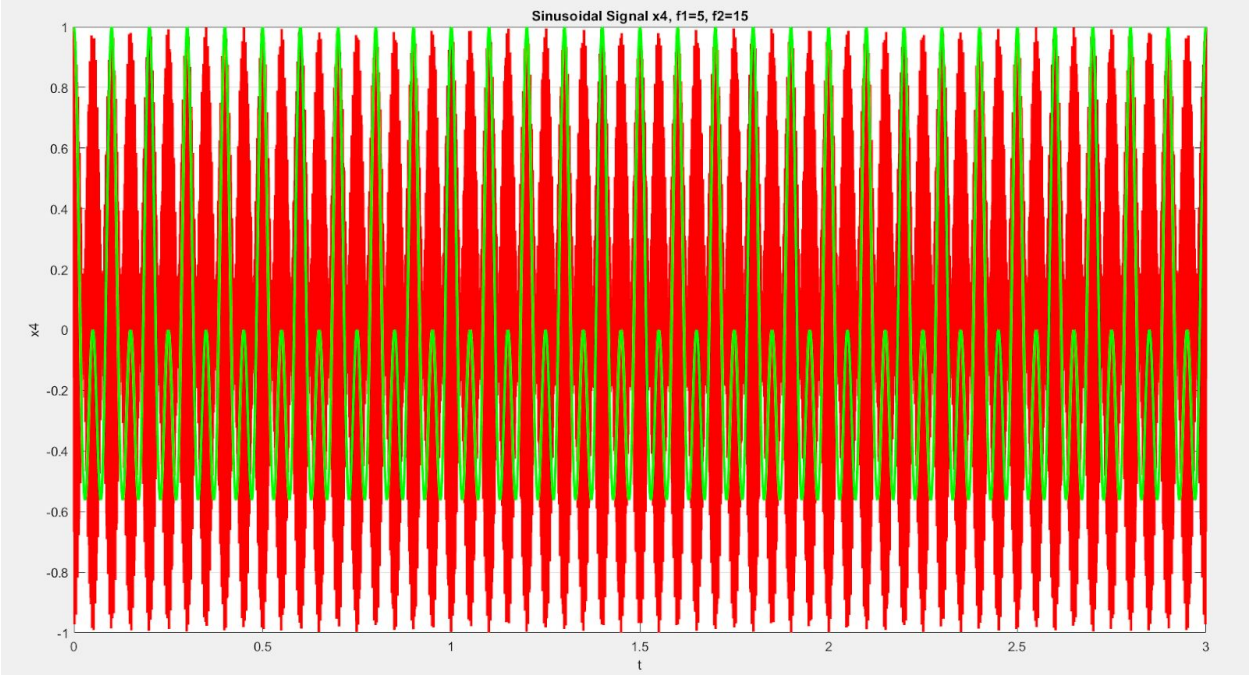
“How does the duration of the sound that you hear change as a increases?”

The higher the a value is, the shorter the sound plays. Being an exponential function the negative exponent for a, made the decreasing rate of the frequency much faster.

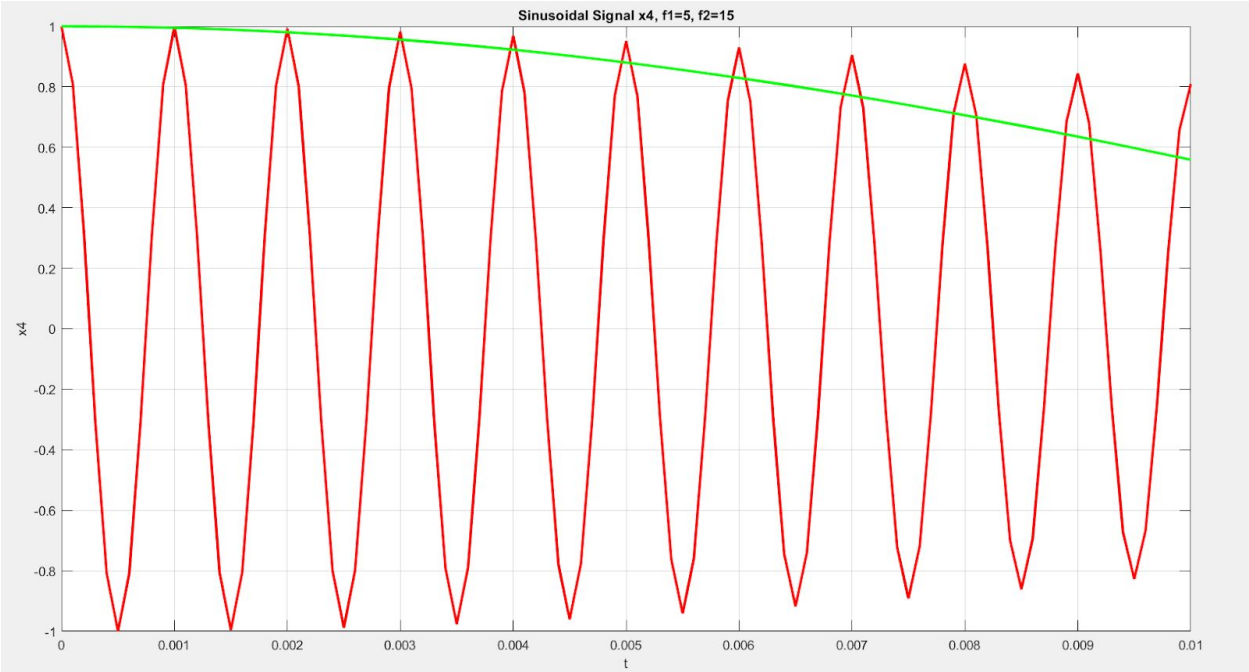
Part 4

“What is the effect of the low-frequency cosine term cos (2 π f1t) on the sound that you hear?“

The effect is as if it is a ringing bell, because of the cosine waves creating a ripple effect.



3 second duration waves part 4



0.01 second duration waves part 4

Part 5

-

Part 6

The code that is given has errors that make it impossible to run. I have yet to figure out the problem, thus I could not create a simple song using the given code.

Complete Code:

```
%% Program runs by itself with each part waiting for the previous one
%% for z seconds.
% set z here
z = 1;
% b sets time between internal parts of parts, set here
b = 1;

%% Part 1 ------------------------------------------------------------
% This is the frequency of A4
f0 = 1*440; % 2*440 would be A5, 4*440 A6 etc...
```

```
Fs = 8192;

% t starts from 0 with an interval of 0.0001 and goes up to 0.01
%t = 0:0.0001:0.01;
t = linspace(0, 24576, 8192)/Fs;

% x1 is the array which uses this function/sinusoid
x1 = sin(2*pi.*f0*t);

% plot the graph
figure('units','normalized','outerposition',[0 0 1 1]) % fullscreen
plot(t, x1,'r', 'LineWidth',2); % red line is x1 or A
hold on % this makes program to keep the existing plot to stay
% and draw the new plot on top of it

% several settings
grid on;
xlabel('t');
ylabel('x1');
title('Sinusoidal Signal A');

sound(x1);

% pause for b sec
pause(b);
% --------- MAJOR TRIAD ------------------------------

eNote = f0 * 3/2;
cSharp = f0 * 5/4;

s = sin(2*pi.*f0*t) + sin(2*pi.*cSharp*t) + sin(2*pi.*eNote*t)

% play sound of A + C# + E / A Major Triad
soundsc(s);

% plot the graph
plot(t, s,'b', 'LineWidth',2); % blue line is sum or Amajor triad

% several settings
grid on;
xlabel('t');
ylabel('s'); % this label replaces the x1 label from y
% axis which should not be a problem.
```

```
title('Sinusoidal Signal A+C#+E');
hold off % simply make plots erasable after this line.

%% Part 2 ------------------------------------------------------------------
pause(z);

% set phi to whatever value
phi = 0;

% set f0 to D frequency
f0 = 587;
% t stays the same in the beginning
x2 = cos(2*pi.*f0*t*phi);

plot(t, x2,'r', 'LineWidth',2); % green line is x2 or D
hold on
% several settings
grid on;
xlabel('t');
ylabel('x2');
title('Sinusoidal Signal D, phi = 0');

% Play D
soundsc(x2);
% --------------------PHI = PI/4----------------------------------
% wait b seconds
pause(b);
% set phi to different value
phi = pi/4;
x2 = cos(2*pi.*f0*t*phi);
soundsc(x2);

plot(t, x2,'b', 'LineWidth',2); % green line is x2 or D

% several settings
grid on;
xlabel('t');
ylabel('x2');
title('Sinusoidal Signal D, phi = pi/4');
% --------------------PHI = PI/2----------------------------------
% wait b seconds
pause(b);
% set phi to different value
```

```
phi = pi/2;
x2 = cos(2*pi.*f0*t*phi);
soundsc(x2);

plot(t, x2,'g', 'LineWidth',2); % green line is x2 or D

% several settings
grid on;
xlabel('t');
ylabel('x2');
title('Sinusoidal Signal D, phi = pi/2');
% --------------------PHI = 3PI/4---------------------------------
% wait b seconds
pause(b);
% set phi to different value
phi = 3*pi/4;
x2 = cos(2*pi.*f0*t*phi);
soundsc(x2);

plot(t, x2,'m', 'LineWidth',2); % green line is x2 or D

% several settings
grid on;
xlabel('t');
ylabel('x2');
title('Sinusoidal Signal D, phi = 3pi/4');
% --------------------PHI = PI-----------------------------------
% wait b seconds
pause(b);
% set phi to different value
phi = pi;
x2 = cos(2*pi.*f0*t*phi);
soundsc(x2);

plot(t, x2,'y', 'LineWidth',2); % green line is x2 or D
hold off
% several settings
grid on;
xlabel('t');
ylabel('x2');
title('Sinusoidal Signal D, phi = pi');
```

```
%% Part 3 ------------------------------------------------------------
% wait z seconds
pause(z);

%------------a = 2----------------------------------------------
f0 = 440;
a = 2;

x3 = (exp(1).^(-(a.^(2) + 2).*t) ).* (cos(2*pi.*f0.*t));

soundsc(x3);

plot(t, x3,'r', 'LineWidth',2);
hold on
% several settings
grid on;
xlabel('t');
ylabel('x3');
title('Sinusoidal Signal x3, a = 2');
%------------a = 1----------------------------------------------
% wait b seconds
pause(b);
a = 1;
x3 = (exp(1).^(-(a.^(2) + 2).*t) ).* (cos(2*pi.*f0.*t));

soundsc(x3);

plot(t, x3,'b', 'LineWidth',2);

% several settings
grid on;
xlabel('t');
ylabel('x3');
title('Sinusoidal Signal x3, a = 1');

%------------a = 3----------------------------------------------
% wait b seconds
pause(b);
a = 3;
x3 = (exp(1).^(-(a.^(2) + 2).*t) ).* (cos(2*pi.*f0.*t));

soundsc(x3);
```

```
plot(t, x3,'g', 'LineWidth',2);
hold off
% several settings
grid on;
xlabel('t');
ylabel('x3');
title('Sinusoidal Signal x3, a = 3');

%% Part 4  ------------------------------------------------------------
% Wait z seconds
pause(z);

%------------f1 = 10-----------------------------------------------
%------------f2 = 1000---------------------------------------------

f1 = 10;
f2 = 1000;

x4 = cos(2* pi.*f1*t).*cos(2*pi.*f2*t);

soundsc(x4);

plot(t, x4,'r', 'LineWidth',2);
hold on
% several settings
grid on;
xlabel('t');
ylabel('x4');
title('Sinusoidal Signal x4, f1=10, f2=1000');

%------------f1 = 5-----------------------------------------------
%------------f2 = 15----------------------------------------------
%pause for b sec
pause(b);

f1 = 5;
f2 = 15;

x4 = cos(2* pi.*f1*t).*cos(2*pi.*f2*t);

soundsc(x4);

plot(t, x4,'g', 'LineWidth',2);
```

```
hold off
% several settings
grid on;
xlabel('t');
ylabel('x4');
title('Sinusoidal Signal x4, f1=5, f2=15');

% ---------- Thetas and Cosines Section ---------------------------------
theta1 = 400;
theta2 = 500;

%x4 =


%% Part 5 ------------------------------------------------------------


%% Part 6 ------------------------------------------------------------
% wait for z seconds
pause(z);

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

% wait for 3 seconds
pause(3);
close;

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