Emre Can Şen- 21902516

Section-2

Lab-01

Handwritten Solutions

Emre Can Şen - 21902516
Section-2

EEE321-Lab01

Part-1

a-) First vector is 4x1 and second is 1x4

b-) The second vectors of these questions are the same but for the first vector the syntax is different, when there are no symbols between components of vectors, Matlab assumes that vector is added in one row.

c-) ';' is used when we don't care about tracking the outcome of the said code line, so it is not presented in command box. And for the first case where semicolon was not used the generation times were between 0.000345 and 0.000169 seconds. For the second case 0.000060 to 0.00002 seconds. These durations don't matter for this application but when generating a sizable vector, we should add semicolon to make computations faster.

d-) '*' in Matlab is used for regular vector multiplication, so two vectors' dimensions should be compatible with each other. In this case they are not compatible, it would be if the second vector were to be 1x4. So we get an error

e-) This syntax is used for element vise multiplication, so it is enough that dimensions are equal whereas it was not enough for previous case. Hence we get: z = [-5 -12 21 -32].

f-) Now that the two vector's dimensions are compatible, Matlab does a vector multiplacation. Each row element of first vector is multiplied by each column element of second vector and they are added to get '-28'. Which is same as sum of all elements of section e.

g-) Same as f, Matlab takes the vector multiplication of these two vectors. Row elements of the first vector are multiplied by the second vector and they are combined with new rows, creating a 4x4 vector.

h-) This command creates a row vector that starts from '11' and increment the elements by '0.02' until the final column is '12'.

i-) Function returns the operator : $\binom{x}{y}$ Binomial coefficient. For this function elapsed time was 0.000323 seconds.

j-) Elapsed time was 0.000101 seconds

k-) " " " 0.00072 seconds. The elapsed time decreased with each new iteration of the function, and the fastest one was the last one.

l-) Matlab treats the elements of a vector as seperate inputs, and repeats the computation for each element seperately. So all the elements are incremented

m-) For plot(x): x variable is inserted to y-axis and x-axis values are the numbers' of the elements of x-axis. X-axis values are t but instead of displaying values of t, the ranking of the t in the vector is displayed (1st column, 2nd column, etc.) For plot(x, t): x variable is inserted to y-axis and x-axis is the values of t. This is confusing because the variable x is actually the y-axis value of the function in conventional sense and the t is the x-axis value. So for plot(t,x): this is the conventional representation of cosine function.

n-) Although the signal looks continuous(analog), it is discrete(digital). So the exact points that are computed, are highlighted with a '+' sign. For the second case, the continuous representation is changed off.

o-) from 0, to 100, 101 points

p-) t=linspace(0, 1, 101);

q-) $x = \sin(\pi t + \pi/3)$;

• For parts r through v-, graphs and codes are appended at the end.

v-) The more data points of t there are, closer the graph will be to continuous case. So, the sampling frequency is greater and less data is lost.

w-) Plot command connects data points with lines. So when there are more data pts., the graph will be closer to real continuous case.

x-) Plot command connects the data points with lines and makes the graphs look continious (analog). This can be closely observed while looking at the 0.2 interval graph. While stem command displays it just as it is (digital).

Part 2

a-) Everything in the computer is discrete (digital) form, so it should be okay to listen.

b-) As far as I can hear 'soundsc' scales the sound to the maximum without clipping the sound. So there wasn't much audible difference between two sound commands.

c-d) As expected pitch got higher and higher, because the frequency increased.

(2) Adding the decaying exponential term makes the signal exponentially decay over time to zero. So for the piano this is also the case, because when a note is pressed the sound produced is the strongest and it decays over time. But for a flute, a note can be held for a period of time so $x_1(t)$ is related with a flute, while $x_2(t)$ is more closely related with a piano sound.

• As the coefficient of 'a' increases, so does the decaying speed. When 'a' is lower, the sound is more prominent.

(3) $\cos a + \cos b = 2\cos\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right) =$

1- $a+b = 4\pi \cdot 440$, $a-b = 4\pi \cdot 4 \Rightarrow a = 4\pi \cdot 222$ and $b = 4\pi \cdot 218 \Rightarrow \cos(2\pi \cdot 444 \cdot t) + \cos(2\pi \cdot 436 \cdot t)$

2- $a = 4\pi \cdot 221$ and $b = 4\pi \cdot 219 \Rightarrow \cos(2\pi \cdot 442 \cdot t) + \cos(2\pi \cdot 438 \cdot t)$

3- $a = 4\pi \cdot 224$ and $b = 4\pi \cdot 216 \Rightarrow \cos(2\pi \cdot 448 t) + \cos(2\pi \cdot 432 t)$

-So as f_1 increases, the gap between cosine's frequencies increases. This results in higher frequency cosine being more dominant, so frequency and pitch of the signal increases.

Part 3

(5) $x_1(t) = \cos(2\pi\phi(t)) = \cos(2\pi f_0 t) \Rightarrow \phi(t) = f_0 t$, $\frac{d\phi(t)}{dt} = f_0 = f_{ins}$

(6) $\phi(t) = \frac{\alpha t^2}{2} \Rightarrow \frac{d\phi(t)}{dt} = \alpha t = f_{ins}$ • $t=0$, $f_{ins}=0$ • $t=t_0$, $f_{ins}=\alpha t_0$

(7) $\phi(t) = -250t^2 + 800t + 4000$, $\frac{d\phi(t)}{dt} = -500t + 800 = f_{ins}$

• Ins. freq. decreases between $t=0$ & $t=1.6$, then starts to increase again linearly. because cosine is an even function.

$t=0 \Rightarrow f_{ins} = 800\,\frac{1}{s}$ $\quad$ $t=1 \Rightarrow f_{ins} = 300\,\frac{1}{s}$

Part 4

• For each case the phase changes but compared to the frequency of the cosine, change in phase is significantly smaller. So it is not really noticeable but each cosine should be delayed more and more as phase increases.

Part 5

$A_1\cos(2\pi f_0 t + \phi_1) = Re\{A_1 e^{2\pi f_0 t j} \cdot e^{j\phi_1}\} = Re\{A_1 e^{j2\pi f_0 t}(\cos\phi_1 + j\sin\phi_1)\}$

$= Re\{A_1\cos\phi_1 e^{j2\pi f_0 t}\}$ $\quad$ (+) $= Re\{e^{j2\pi f_0 t}\underbrace{(A_1\cos\phi_1 + A_2\cos\phi_2)}_{A_3}\} = A_3\cos(2\pi f_0 t)$

$\Rightarrow Re\{A_2\cos\phi_2 e^{j2\pi f_0 t}\}$

$A_3 = A_1\cos\phi_1 + A_2\cos\phi_2$, $\phi_3 = 0$, $f_3 = f_0$

a-) $\cos x$ takes maximum of 1 & min. of -1 ; $A_1 > A_2 > 0 \Rightarrow A_3 = -A_1 - A_2$

$\Rightarrow \phi_1 = \pi + 2\pi k$ $\quad$ $\phi_2 = \pi + 2\pi k$

b-) $A_3 = A_1 + A_2 \Rightarrow \phi_1 = 0 + 2\pi k$ $\quad$ $\phi_2 = 0 + 2\pi k$ $\quad$ where k's are integers

Part 1

r-

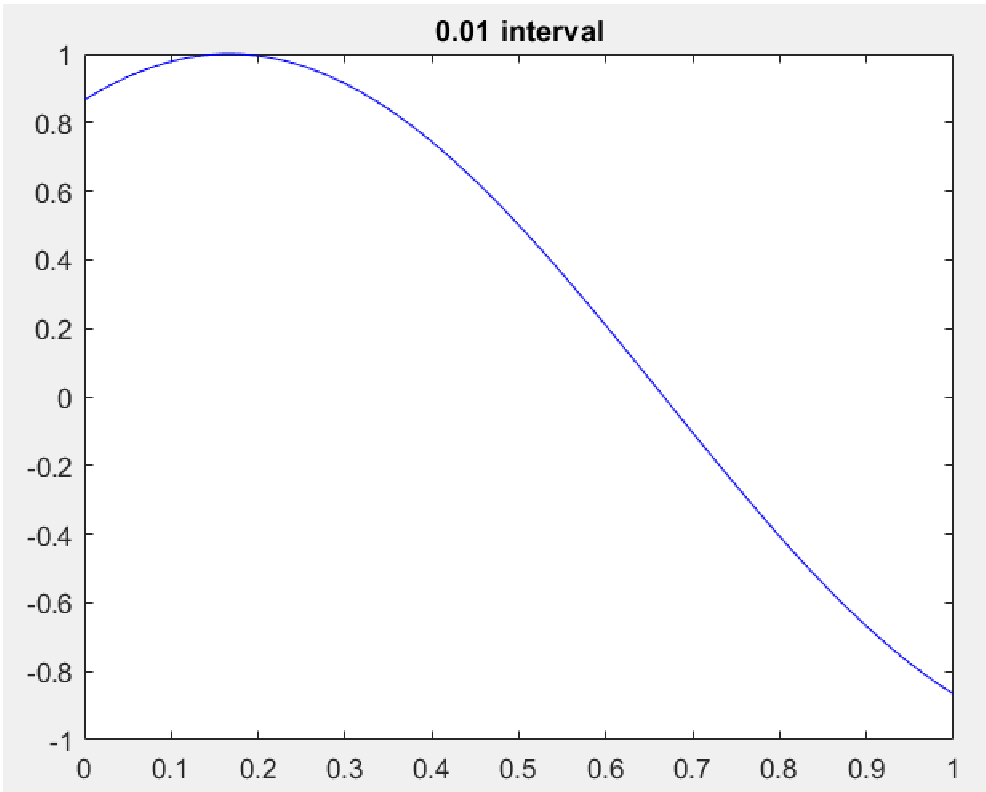

Figure 1- 0.01 Interval Graph

s- t=[0:0.025:1]; or t=linspace(0,1,41);

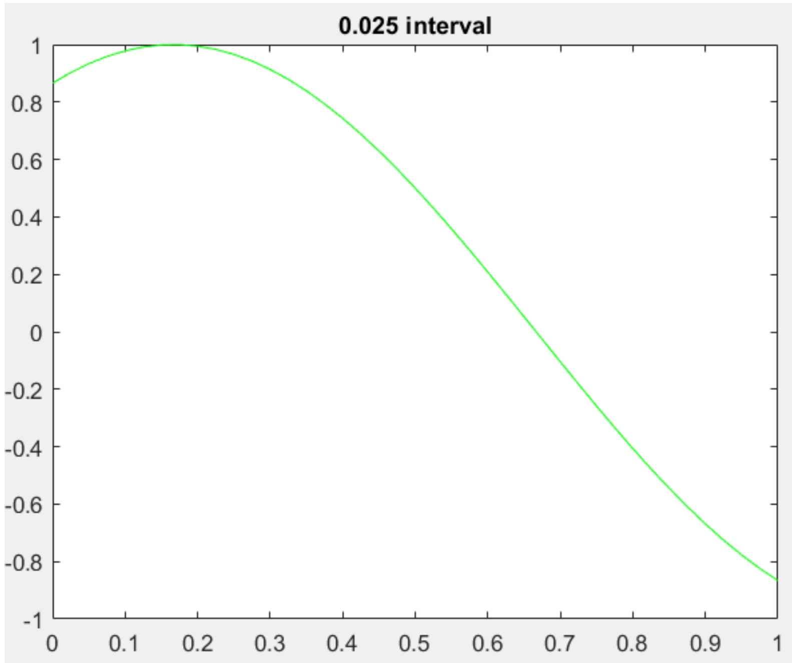

Figure 2- 0.025 Interval Graph

t-

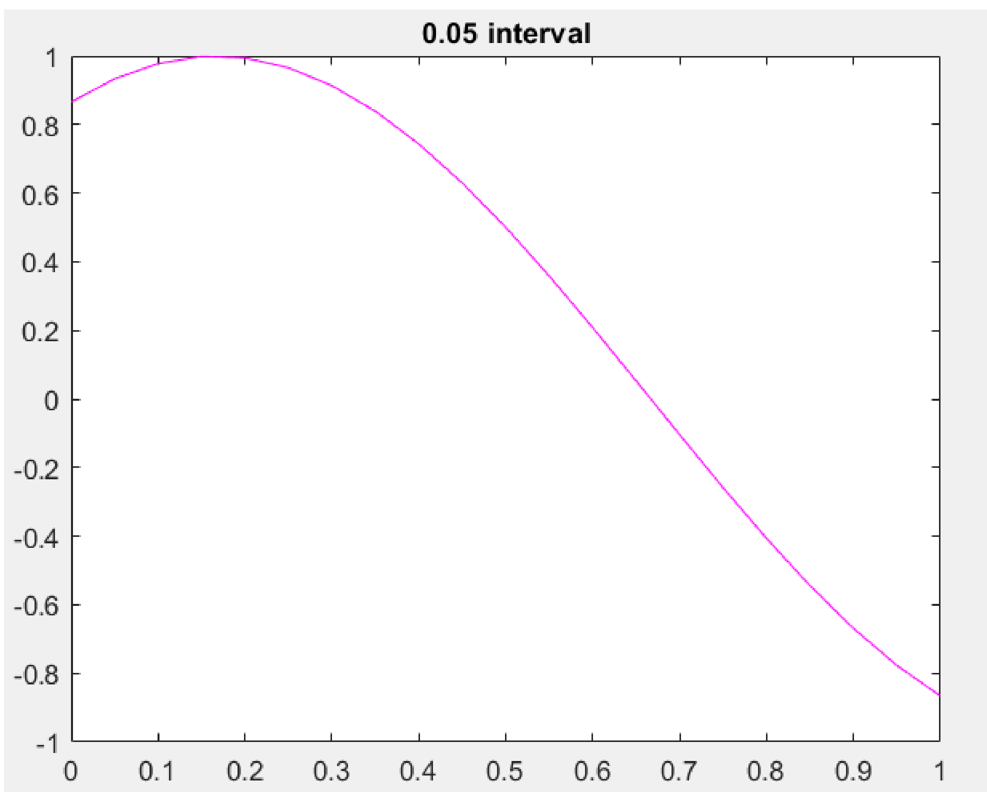

Figure 3- 0.05 Interval Graph

u-

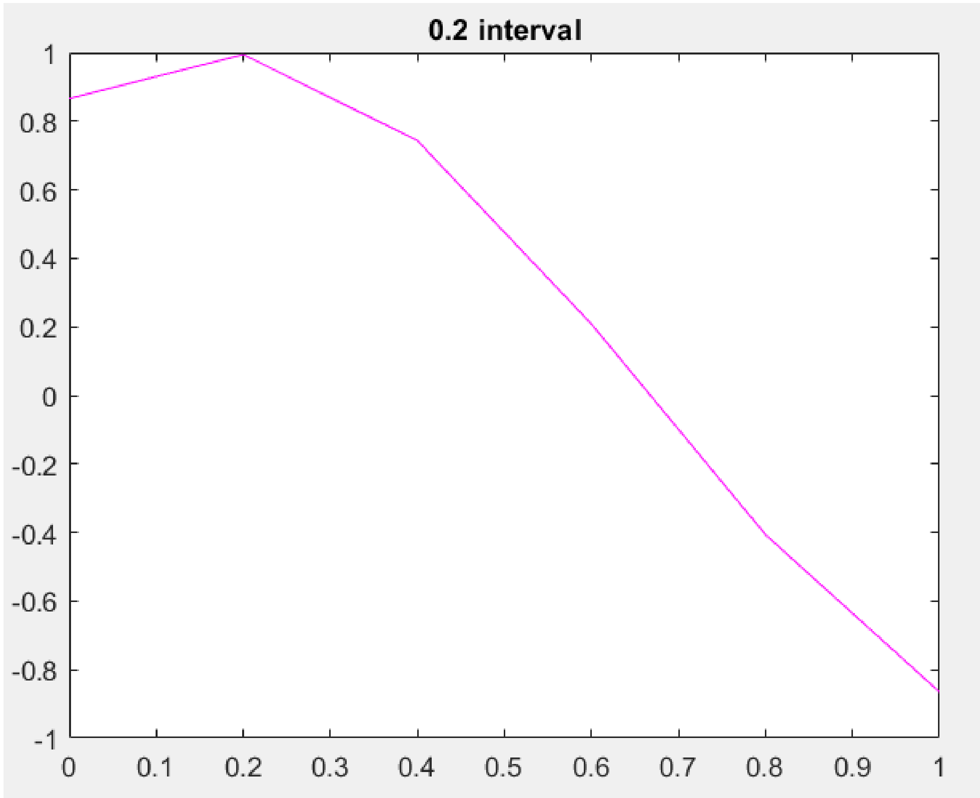

Figure 4- 0.2 Interval Graph

v-

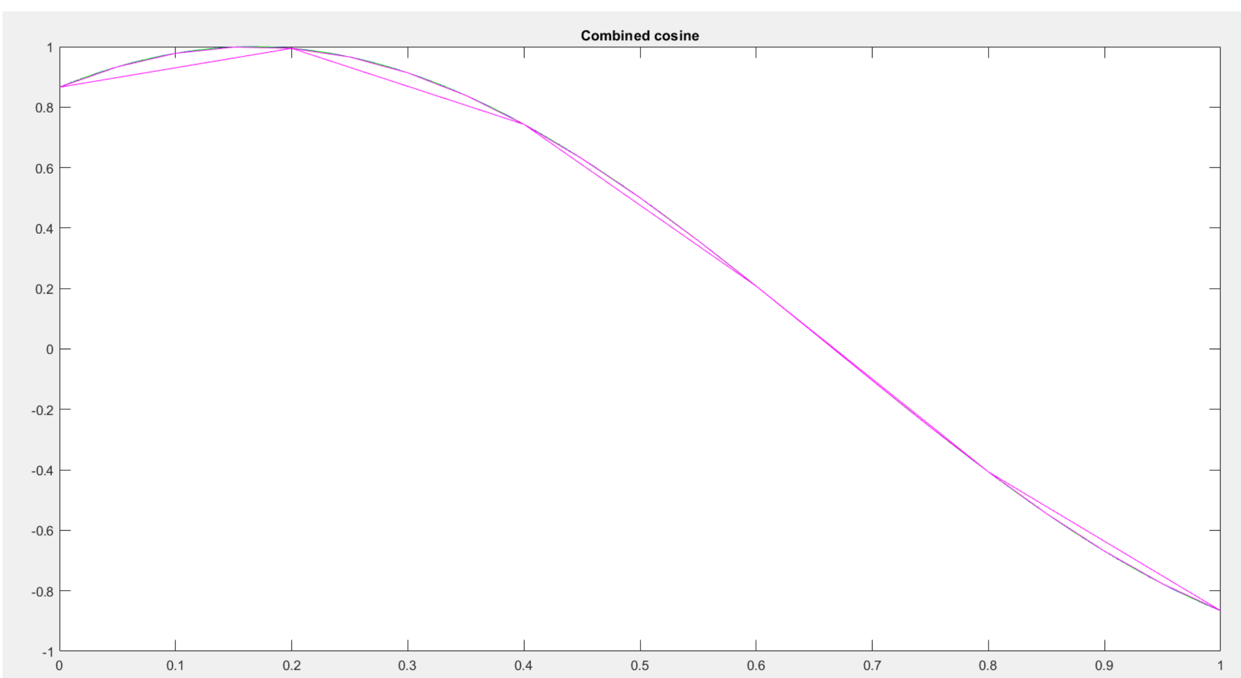

Figure 5- All Graphs Combined

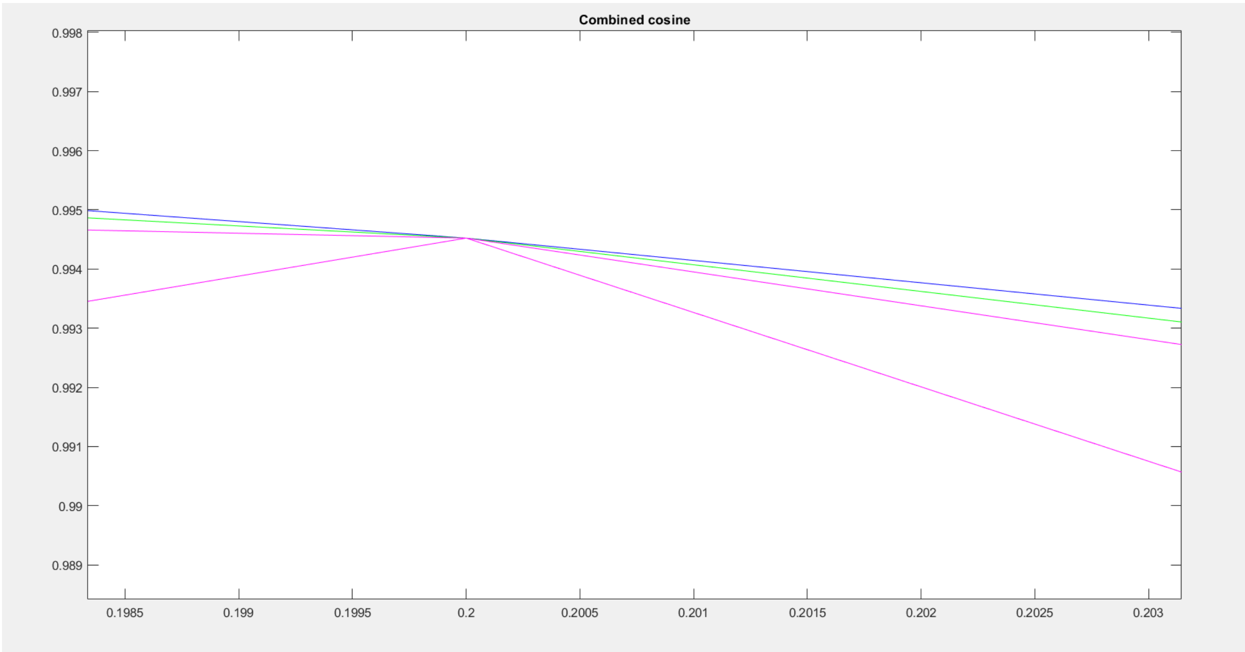

Figure 6- Zoomed Combined Graphs

Part 2

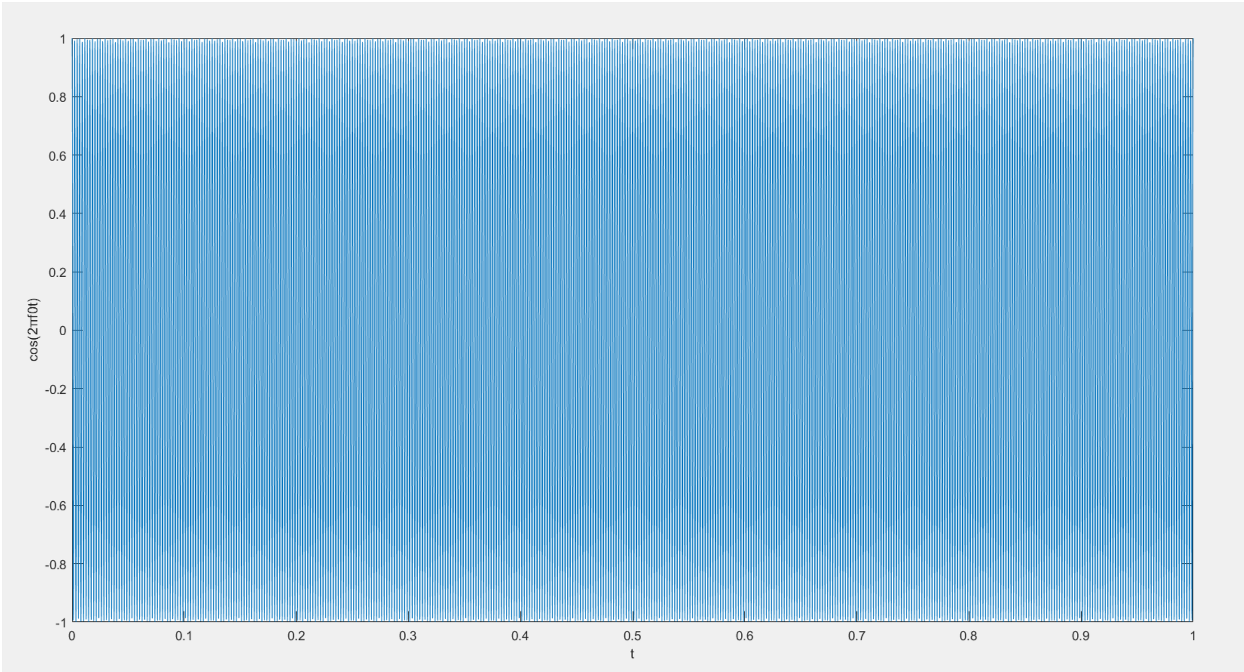

Figure 7- cos(2πf0t) vs time

(2)

```
t=[0:1/8192:1];
a=8;
f=880;
x2= (exp(-a*t)).*cos(2*pi*f*t);

figure
plot(t,x2);
ylabel('e^-at*cos(2?f0t)');
xlabel('t');
sound(x2)
```

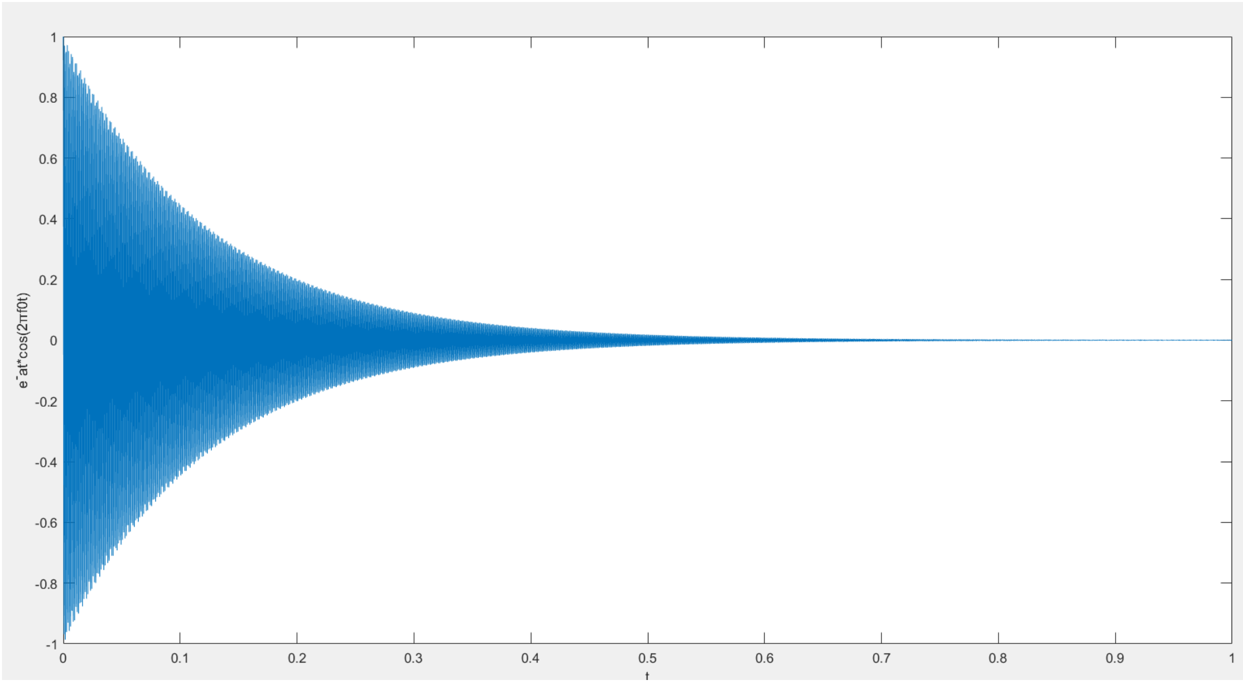

Figure 8- e^−αt*cos(2πf0t) vs time

(3)

<u>Part 3</u>

<u>Part 4</u>

a- It doesn't change???

<u>Part 5</u>

<u>MATLAB Codes</u>

```
%{
tic
x=[-5, 1.2, 1/2, 3];
toc

tic
for i=40
    x=nchoosek(50,i);
```

```
end;
toc

x=zeros(1)
tic
x=nchoosek(50,40)
toc


figure
t=linspace(0,1,101);
x=sin(pi*t+pi/3);
plot(t,x,'b');
title('0.01 interval')

hold on;
t=[0:0.025:1];
x=sin(pi*t+pi/3);
plot(t,x,'g');
title('0.025 interval')

hold on;
t=[0:0.05:1];
x=sin(pi*t+pi/3);
plot(t,x,'m');
title('0.05 interval')
hold on;

t=[0:0.2:1];
x=sin(pi*t+pi/3);
plot(t,x,'m');
title('Combined cosine')
hold on;


t=[0:1/8192:1];
f=783;
x1= cos(2*pi*f*t);

figure
plot(t,x1);
ylabel('cos(2?f0t)');
xlabel('t');
sound(x1)
```

```
t=[0:1/8192:1];
a=16;
f=880;
x2= (exp(-a*t)).*cos(2*pi*f*t);

figure
plot(t,x2);
ylabel('e^-at*cos(2?f0t)');
xlabel('t');
sound(x2)


t=[0:1/8192:1];
f0=440;
f1=8;
x3= (cos(2*pi*f0*t)).*cos(2*pi*f1*t);

figure
plot(t,x3);
title('f1=8')
ylabel('cos(2?f1t)cos(2?f0t)');
xlabel('t');
sound(x3)


t=[0:1/8192:1];
a=1870;
x4=cos(pi*a*t.^2);
sound(x4)

t=[0:1/8192:2];
x5=cos(2*pi*(-250*t.^2+800*t+4000));
sound(x5)
%}

t=[0:1/8192:1];
a=1870;
ps=pi;
x6=1/2*cos(2*pi*a*t+ps);
sound(x6)
```