Atahan İyiekici
21801985

part 1

a) first one gives a 1x4 matrix on the other hand second one gives a 4x1 matrix. These matrices are different because of comma's and semicolon

b) There is no difference from part a. Putting spaces in MATLAB inside a matrix gives the same output as putting comma's

c) Semicolon at the end of command line enables not to show output on command window. When semicolon is not used then, it outputs in command window.

d) Incorrect dimensions for matrix multiplication; error is occured. I received this message because dimensions of matrices aren't matched.

e) Adding the dot before multiplication symbol changes the result and outputs [-5 -12 21 -32]. Adding the dot changes the result because when we add dot before multiplication symbol MATLAB understand that is element-wise multiplication, therefore, it gives the output.

f) the result is -28. Matlab has done the matrix multiplication row.

g) the result is $\begin{bmatrix} -5 & 6 & -7 & 8 \\ 10 & -12 & 14 & -16 \\ 15 & -18 & 21 & -24 \\ 20 & -24 & 28 & -32 \end{bmatrix}$, Matlab has done the matrix multiplication for 4x1 and 1x4 matrices.

h) this command creates a 1x51 matrix with values between 11 to 12 and these values are increased by 0.02.

i) it outputs x = 1.0272e+10 which means 1.0272×10^{10}

Spent time is t = 0.000154 seconds

nchoosek command simply does $\frac{n!}{k!(n-k)!}$ this calculation, which is known as combination.

j) Spent time with for loop is t = 0.052478 seconds

k) Spent time with giving memory allocation t = 0.001466

$t_i > t_k > t_j$ Method i is the most efficient one.

l) Matlab does evaluate all cosine values in the array "a". In other words, matlab maps from array "a" to discrete real values, with cosine function $y = \cos(a) + 1$; command adds "1" value to all values of $\cos(a)$ just like a vector summation

m) plot(x) and plot(x,t) plotted as same because Matlab gives t parameter to plot (x) command, this is because of $x = \cos(2\pi t)$ which is a function of t. But in plot(t,x) command, it plots different than other commands. in plot(t,x) the x axis is x variable and y axis is t variable which is the opposite of others.

n) the first command plot (t,x,'-+') fills the empty lines so that the plot seems like it is continuous. The second command plot(t,x,'+') puts "+" on the discrete values.

o) t=[0:0.01:1] has 101 time points.

p) There are 101 time points in part m, thus, t=linspace (0,1,101) can work.

q-r) – nothing to write –

s) t=[0:0.25:1] has 41 time points

t-u) –

v) continuous x(t) is most likely the smallest step size one, because it contains more time points than others.

w) It draws a straight line between two data points.

x) Stem shows discrete points clearly at exact points there are circle's but in plot command there are no circles to show all discrete points

Scanned with CamScanner

Part 2

a) Yes, both commands are appropriate

b) It can be heard in higher frequency, it gets higher in terms of pitch.

* Adding $e^{-\alpha t}$ to $\cos(2\pi f_0 t)$ makes the function decaying sinusoidal.
I hear the voice decaying as its magnitude. $x_1(t)$ has no decay.
$x_1(t)$ resembles more of a flute and $x_2(t)$ resembles more of a piano.
As α increases, fast-decay occurs. This fast-decay causes a short duration.
Therefore, in $\alpha=2$ I hear long duration on the other hand in $\alpha=8$ I hear short duration in compare to other.

* for $x_3(t)$ there are two cosine's that has different frequencies that makes the signal various. $\cos(2\pi f_0 t)$ sounds like a few low magnitude beep sound and $\cos(2\pi f_1 t)$ sound like siren with low magnitude and it makes signal fluctuated. As f_1 gets larger fluctuation becomes less.

* The trigonometric identity $\Rightarrow \cos(a)\cdot\cos(b) = \frac{1}{2}\cos(a-b)+\frac{1}{2}\cos(a+b)$

Thus, we have $a = 8\pi t$, $b = 880\pi t$ for $x_3(t)$

$\Rightarrow \frac{1}{2}\cdot\left(\cos(8\pi t-880\pi t)+\cos(8\pi t+880\pi t)\right)$ which gives very similar sound as $x_3(t)$

Part 3

$x_1(t) = \cos(2\pi f_0 t) \rightarrow \phi(t) = f_0(t)$; Thus $\phi'(t) = f_0$

$x_4(t) = \cos(2\pi \frac{\alpha t^2}{2}) \rightarrow \phi(t) = \frac{\alpha}{2}(t^2)$, $\phi'(t) = \alpha t$ at all t

At $t=0$ frequency of $x_4(t)$ is 0 and in $t=t_0$ is αt_0 (1825)

$\alpha = 1600+5\times45 = 1825$, $x_4(t) = \cos(\pi\cdot 1825t^2)$

$\phi'(t) = 1825t$ frequency range is $[0, 1825]$

As α increases, the function becomes more frequent and fast.

* $x_5(t) = \cos(2\pi(-250t^2+800t+4000))$

$\phi(t) = -250t^2+800t+4000$

$\phi'(t) = -500t+800$

$t=0 \rightarrow F_0 = 800Hz$
$t=1 \rightarrow F_1 = 300Hz$
$t=2 \rightarrow F_2 = -200Hz$ which makes aliasing

As time passes the frequency value is decreasing

Part 4

Volume and pitch does not change because changing phase of the signal only shifts the signal.

Part 5

Atahan İyiekici
21801985
EEE321-Lab report 1

EEE321 Lab Report 1 Codes and Plots

Plots

Part 2

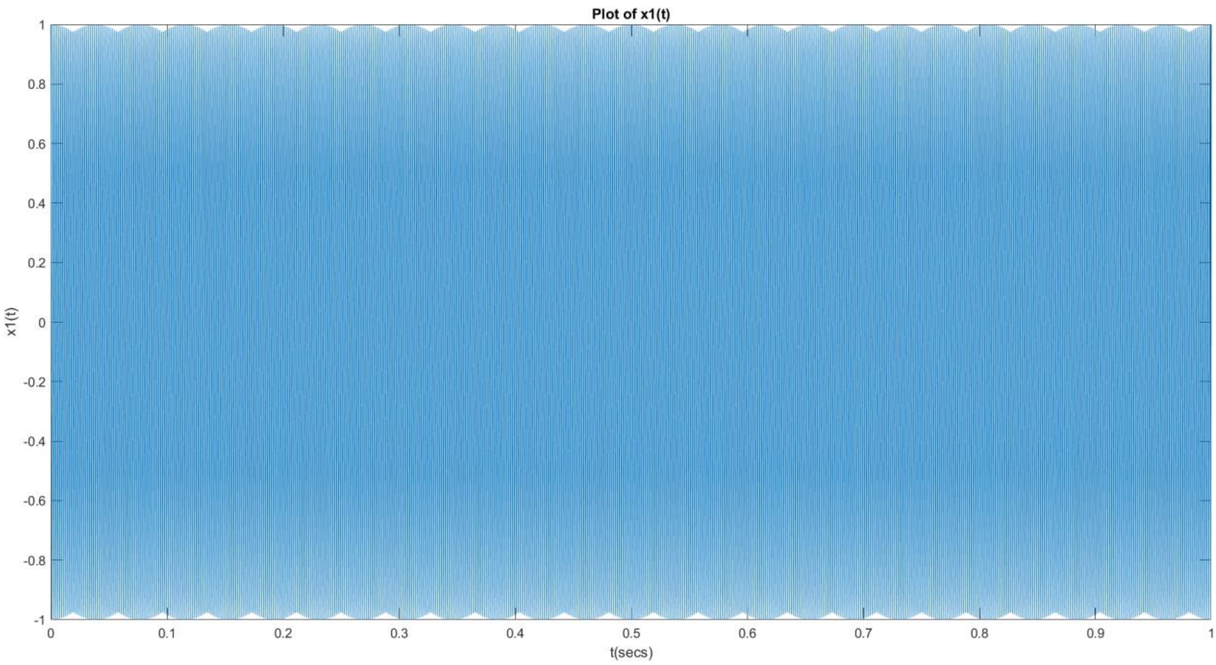

Figure 1: Plot of cos(2π * 440 * t)

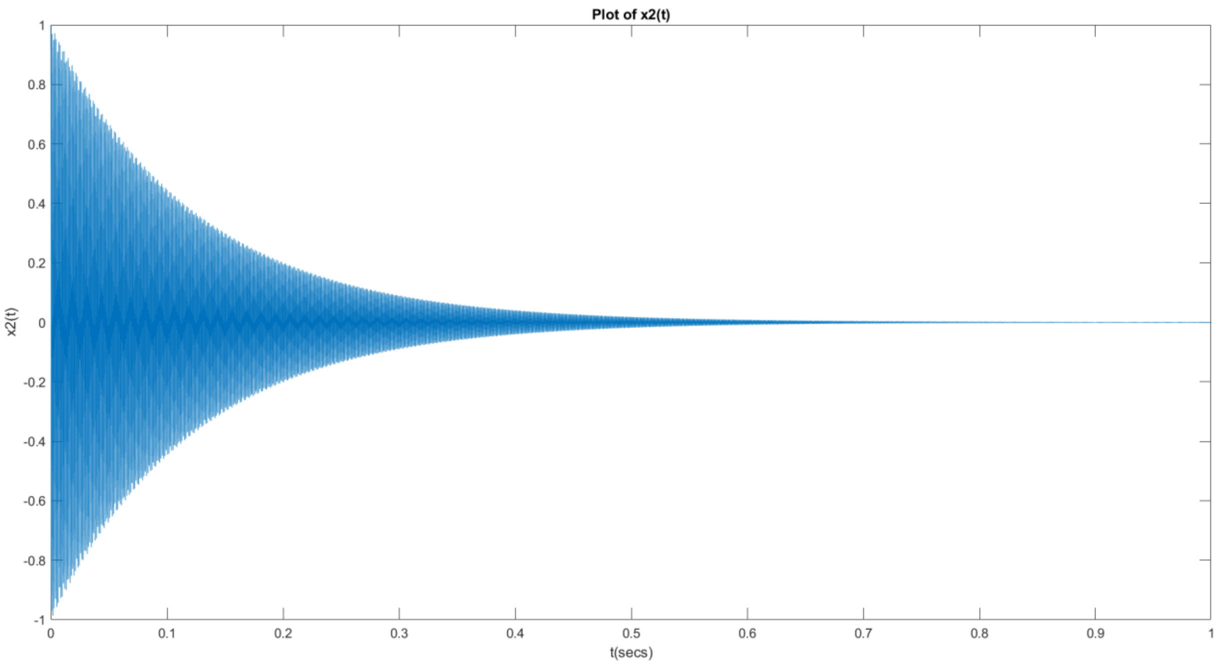

Figure 2: Plot of cos(2π * 880 * t) * e -8t

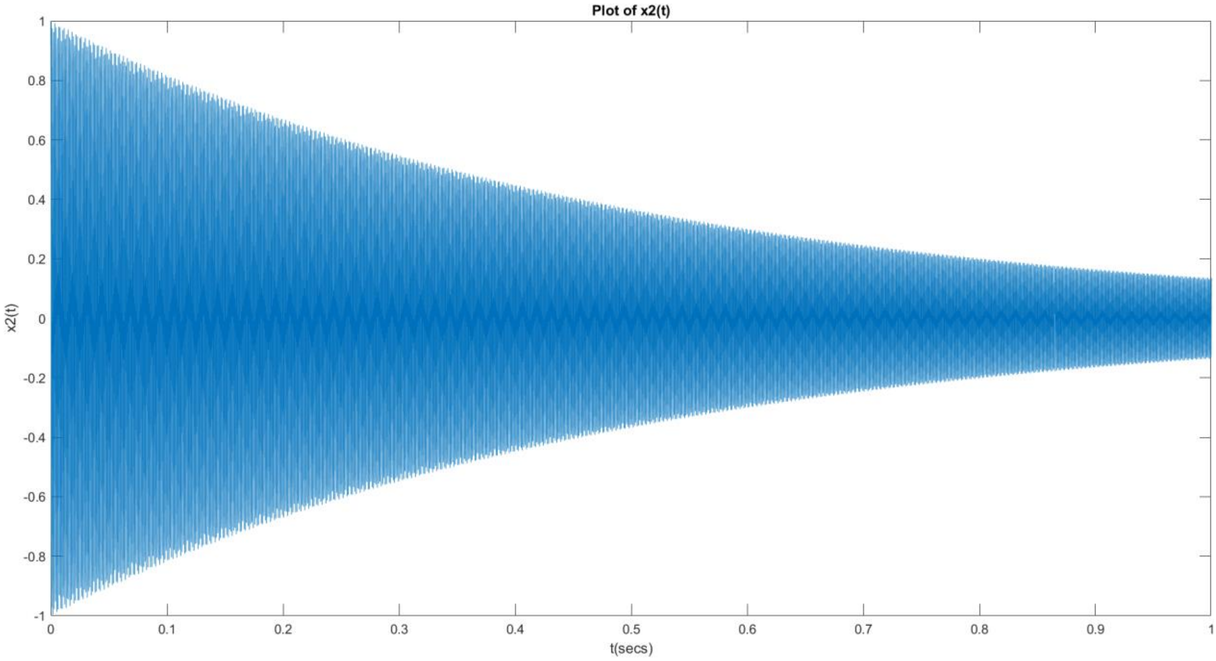

Figure 3: Plot of cos(2π * 880 * t) * e -2t

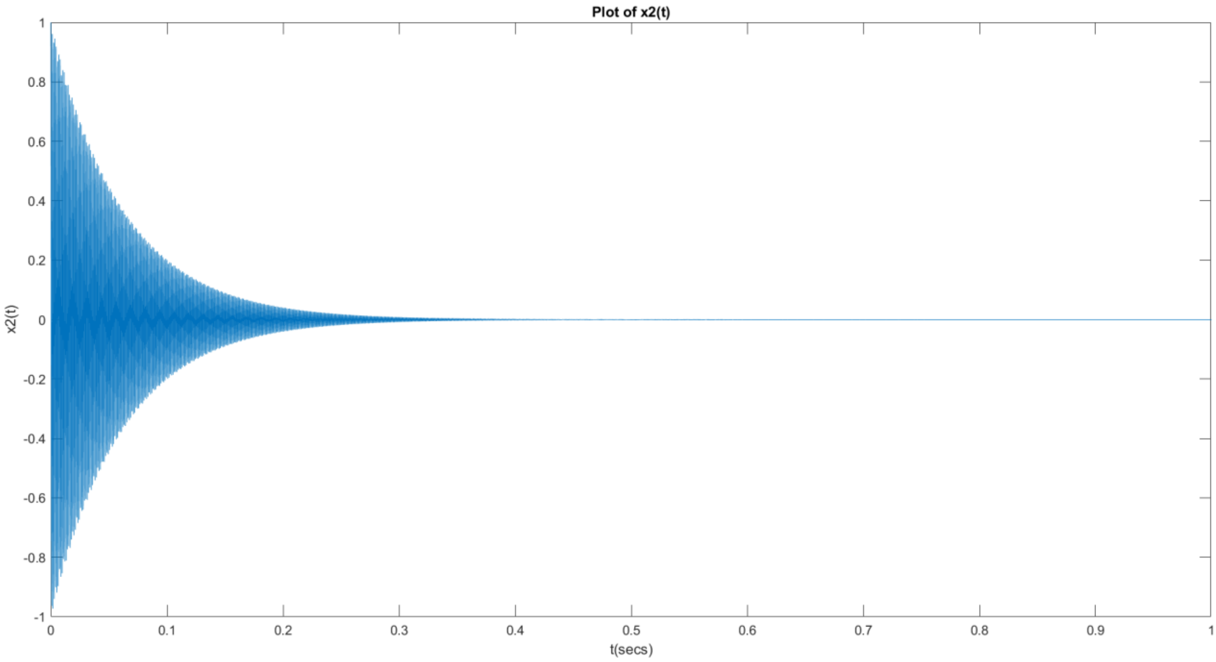

Figure 4: Plot of cos(2π * 880 * t) * e -16t

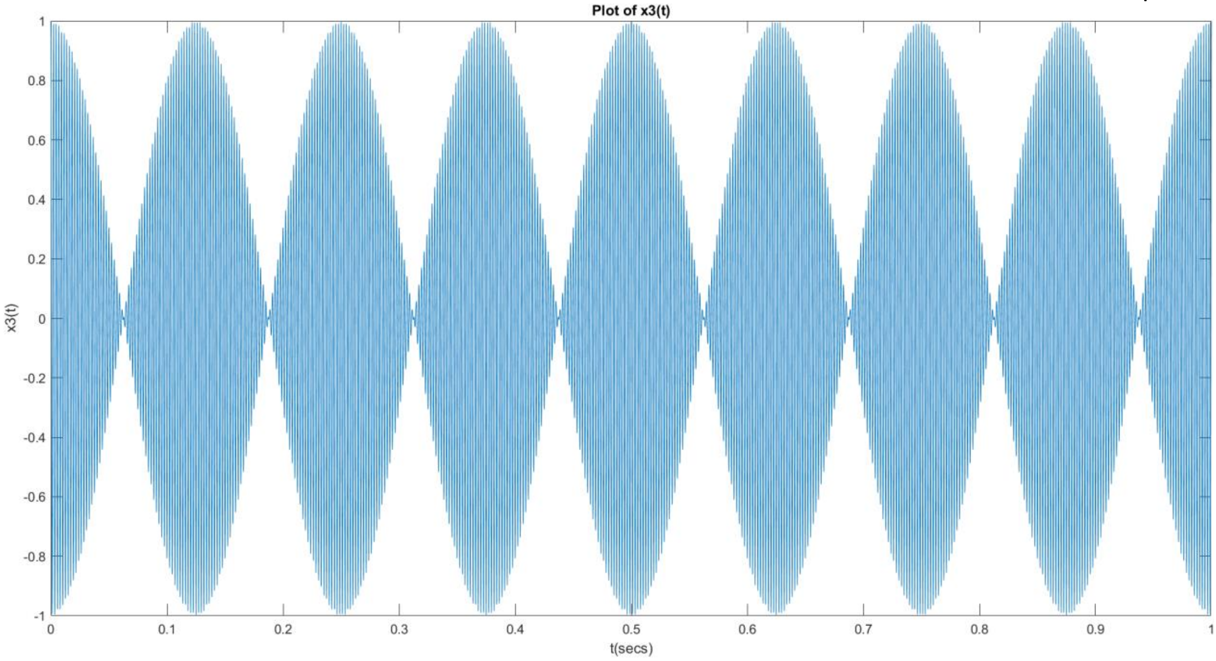

Figure 5: Plot of cos(2 * π * 4 * t) * cos(2 * π * 440 * t)

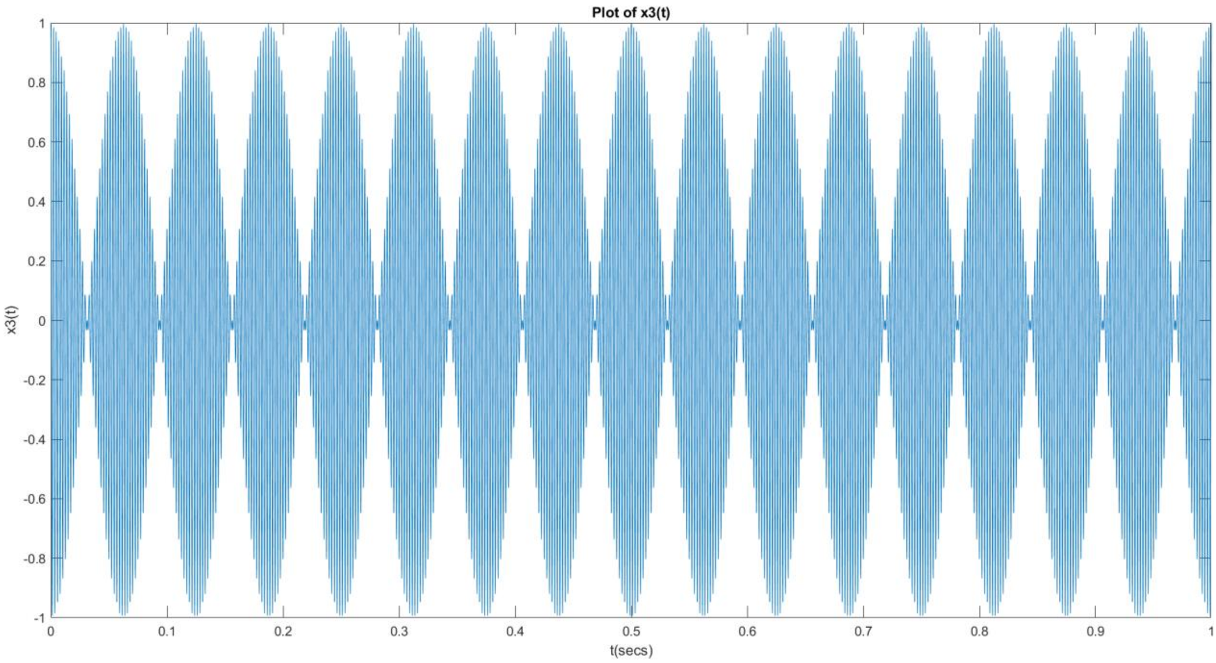

Figure 6: Plot of cos(2 * π * 8 * t) * cos(2 * π * 440 * t)

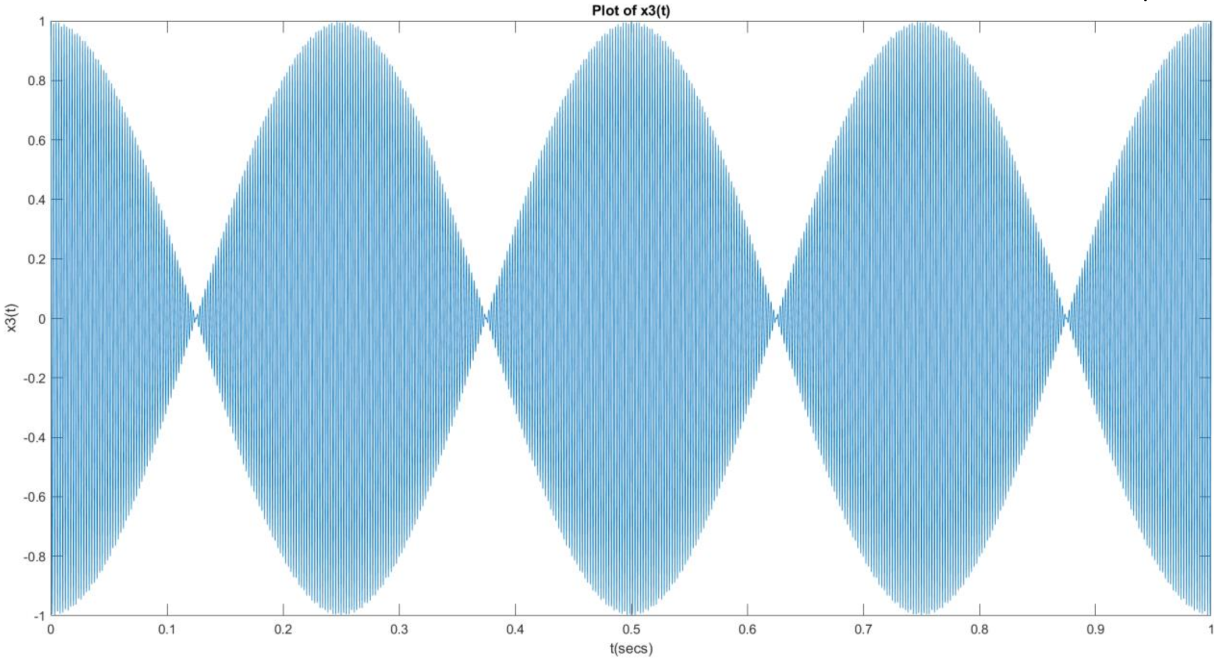

Figure 7: Plot of cos(2 * π * 2 * t) * cos(2 * π * 440 * t)

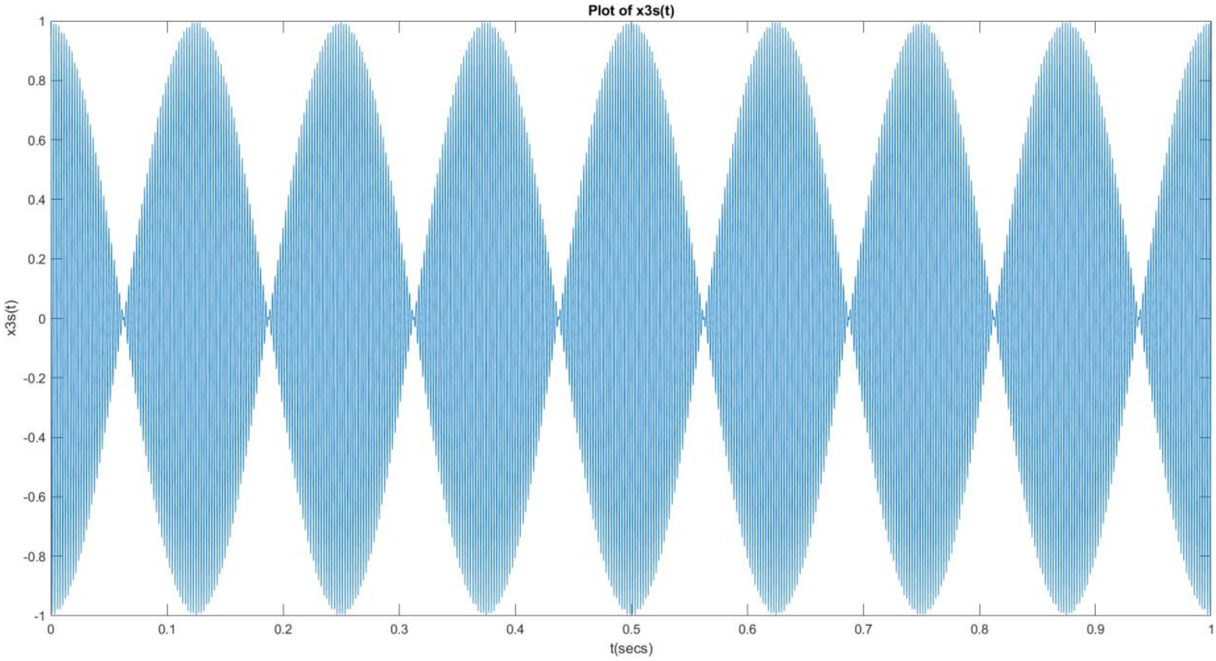

Figure 8: Plot of (1/2) * (cos(2 * π * (4 -440) * t) + cos(2 * π * (4+ 440) * t))

Part 3

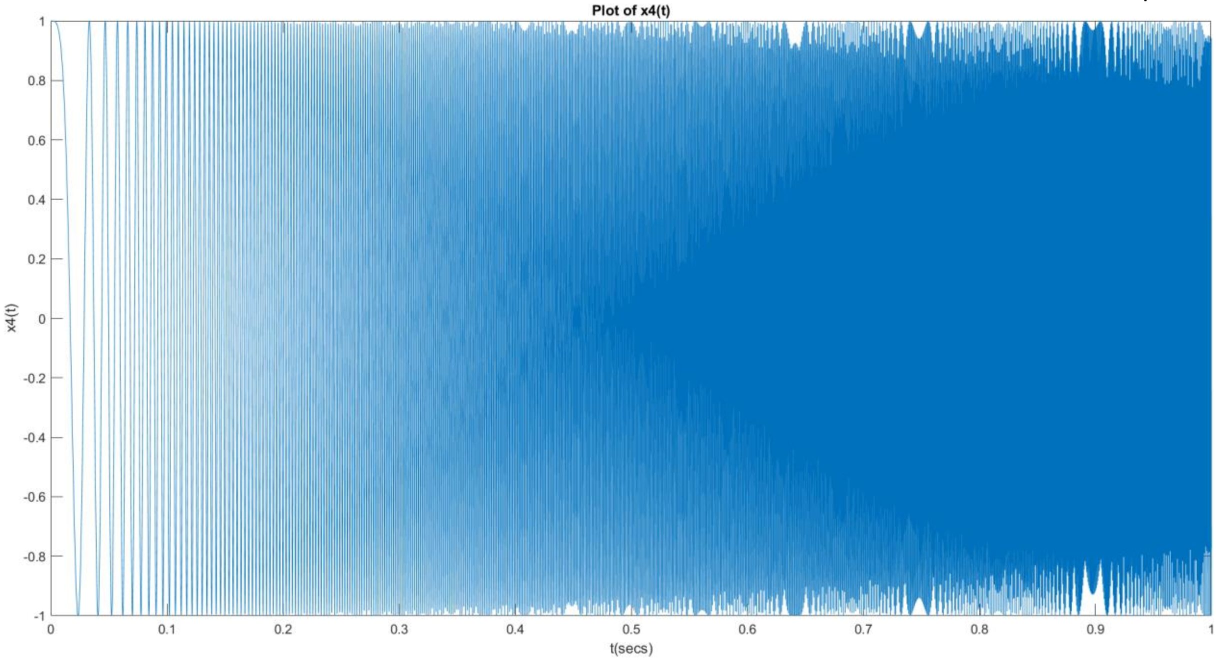

Figure 9: Plot of cos(π * 1825 * (t2))

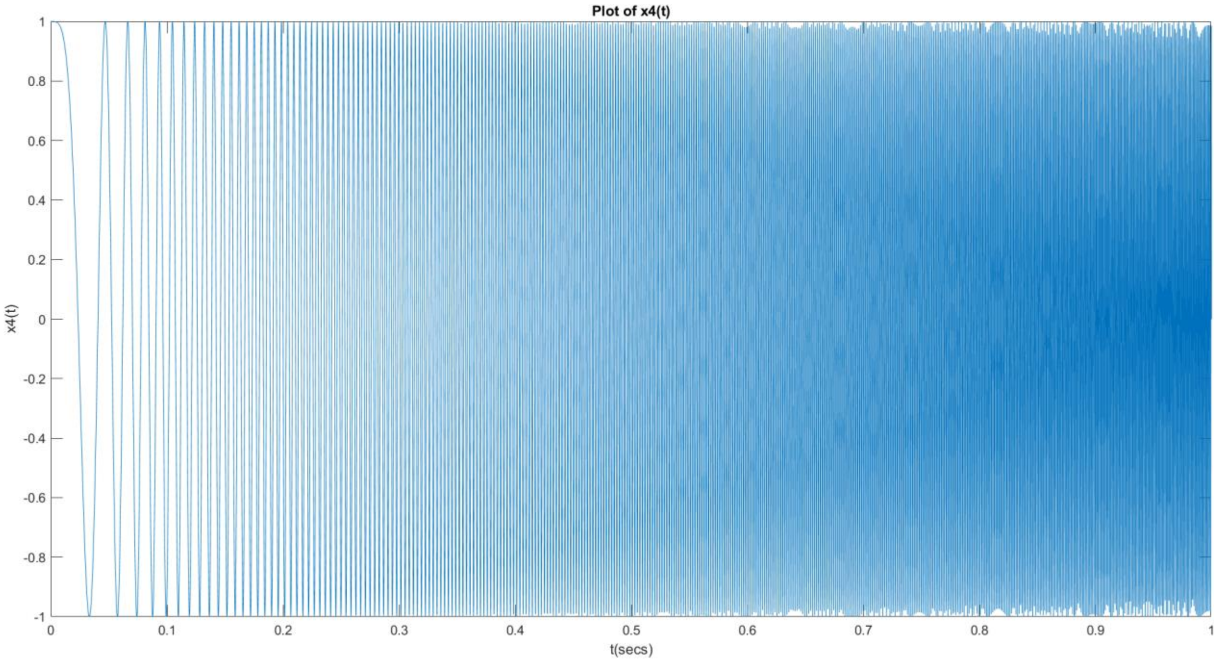

Figure 10: Plot of cos(π * 1825/2 * (t2))

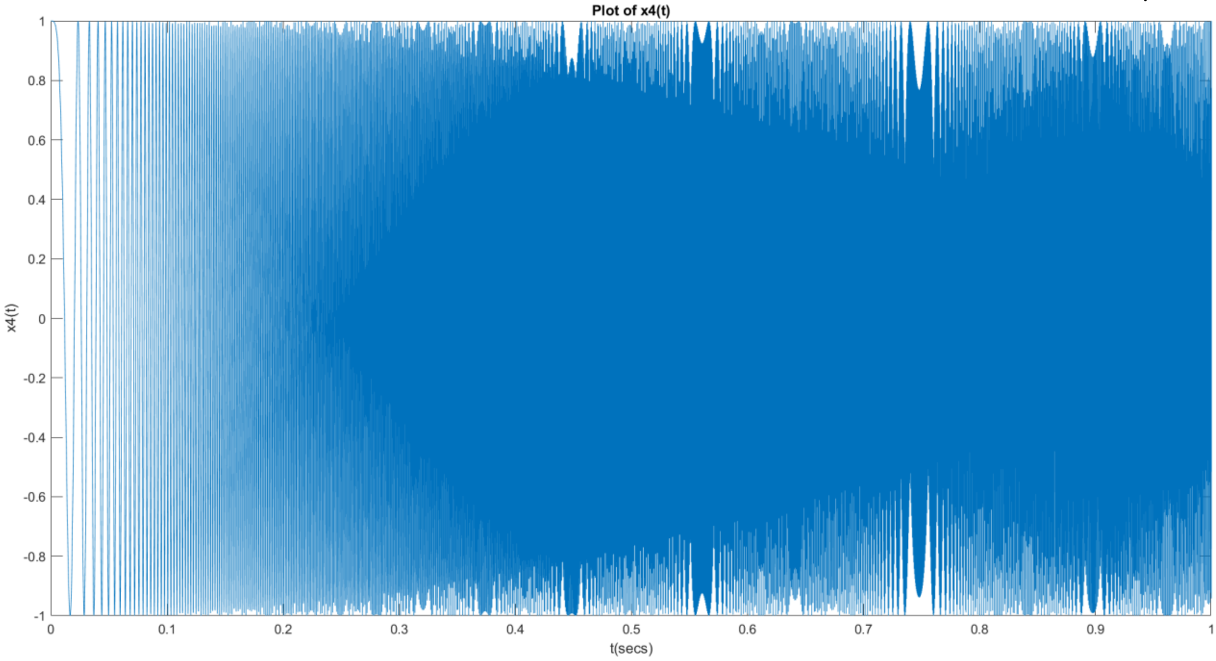

Figure 11: Plot of cos(π * 1825*2 * (t2))

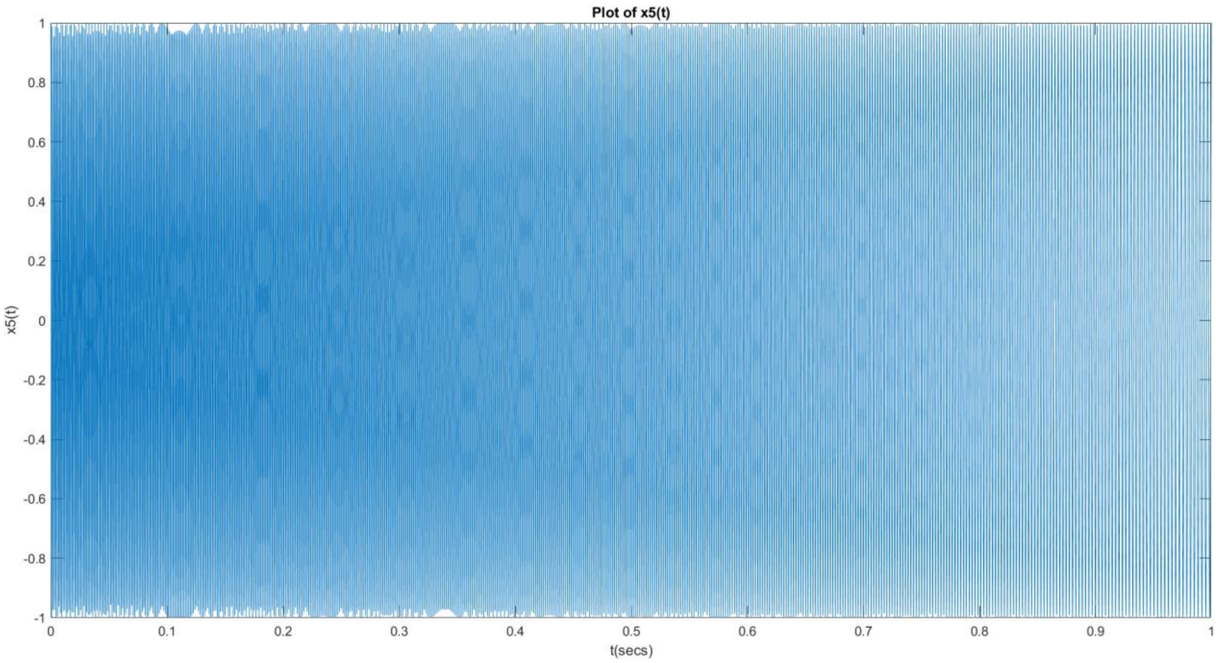

Figure 12: Plot of cos(2 * π * (-250 * (t^2) + 800 * t + 4000))

Part 4

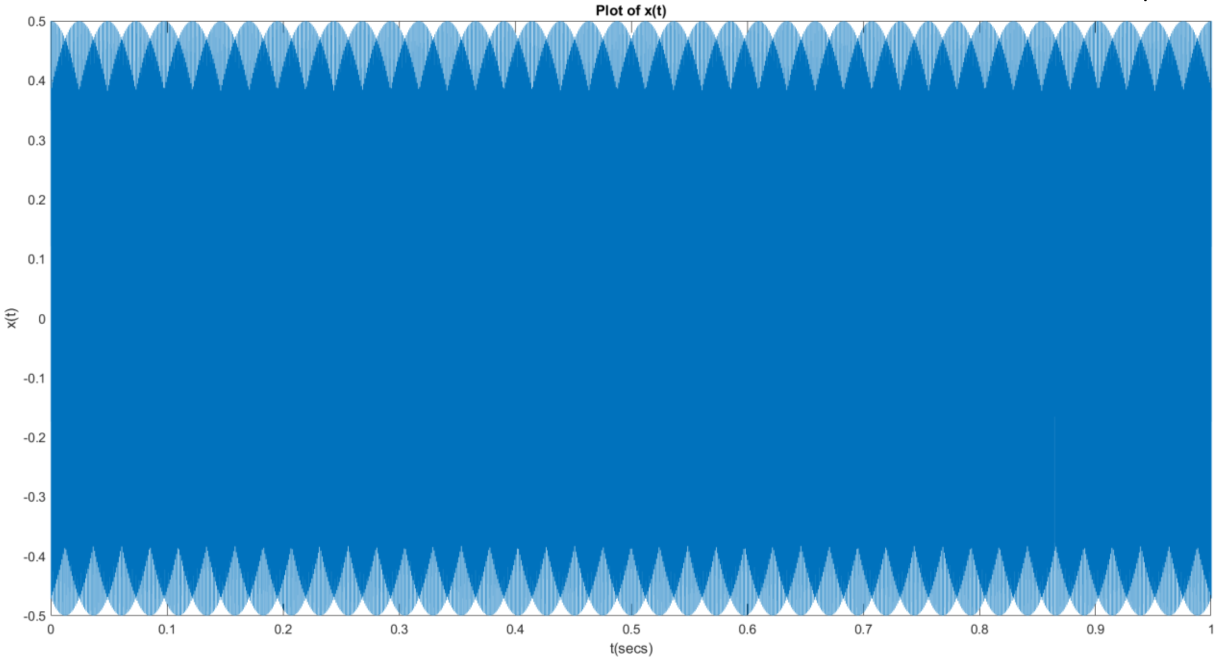

Figure 13: Plot of (1 / 2) * cos(2 * π * 1825 * t)

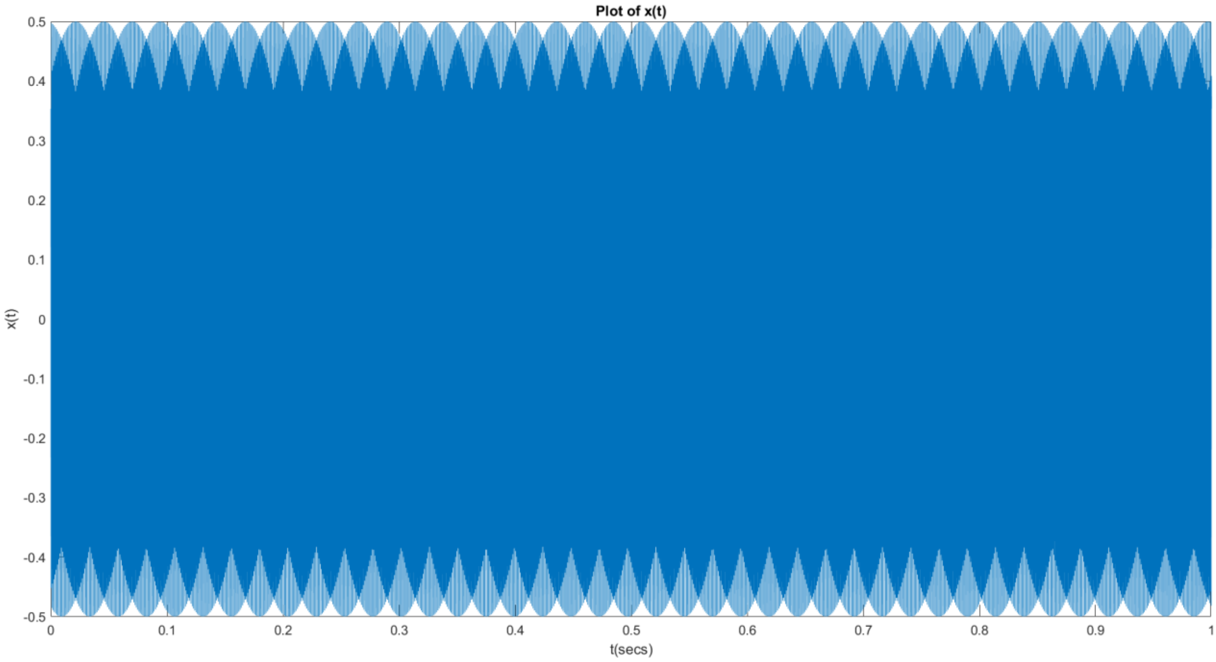

Figure 14: Plot of (1 / 2) * cos(2* π * 1825 * t + π / 4)

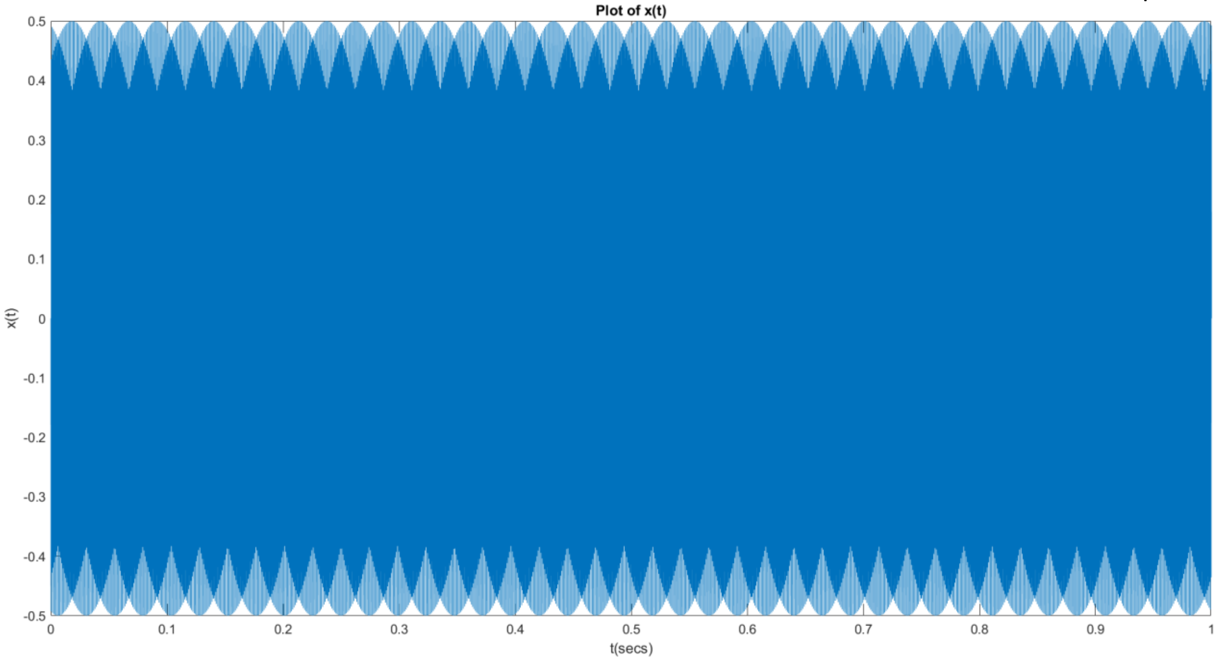

Figure 15: Plot of (1 / 2) * cos(2 * π * 1825 * t + π / 2)

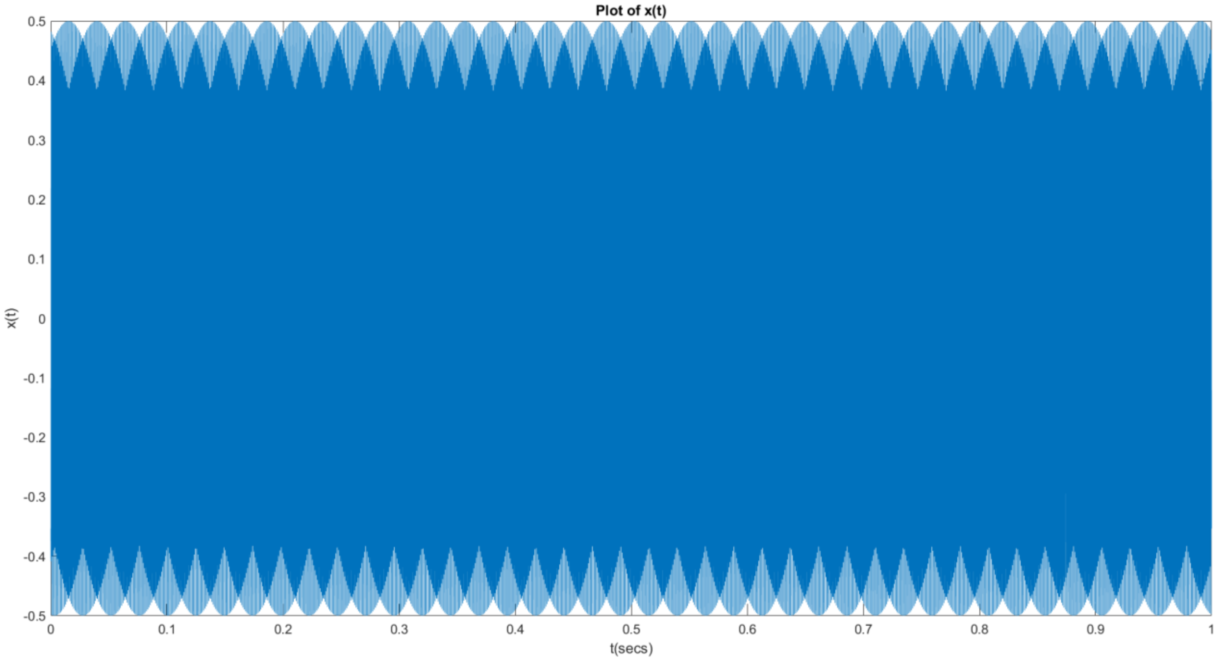

Figure 16: Plot of (1 / 2) * cos(2 * π * 1825 * t + 3 * π / 4)

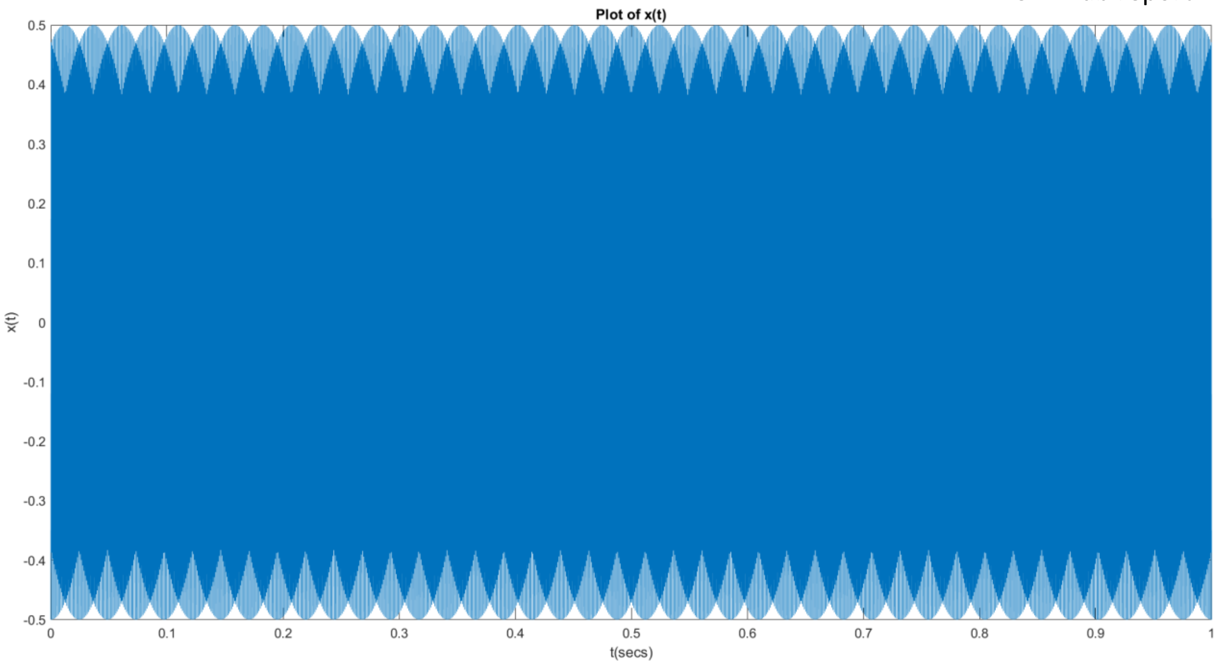

Figure 17: Plot of (1 / 2) * cos(2 * π * 1825 * t + π)

Codes

```
%% Part-2 %%
close all
clear all
t = [0:1/8192:1];
%%f_0 = 440;
f_0 = 587;
%%f_0 = 783;
x1_t = cos(2*pi*t*f_0);
%x1 = 'x1.mat';
%save(x1,"x1_t");

%sound(x1_t);
%soundsc(x1_t);

f_0 == 440;
figure;
plot(t,x1_t);
xlabel('t(secs)');
ylabel('x1(t)');
title('Plot of x1(t)');
%-----------------------------------------------
f_0 = 880;
%a = 2;
%a = 8;
a = 16;
x2_t = exp(-(t)*a).*(cos(2*pi*t*f_0));

figure(2);
plot(t,x2_t);
```

```
xlabel("t(secs)");
ylabel("x2(t)");
title("Plot of x2(t)");

x2 = 'x2.mat';
save(x2,"x2_t");

%sound(x2_t);
%soundsc(x2_t);
%-----------------------------------------------
f_0= 440;
%f_1= 4;
f_1= 2;
%f_1=8;
x3_t=cos(2*pi*f_1*t).*cos(2*pi*f_0*t);
%sound(x3_t)
%x5=cos(2*pi*f_0*t)
%sound(x5)
figure(3);
plot(t,x3_t);
xlabel("t(secs)");
ylabel("x3(t)");
title("Plot of x3(t)");
x3s_t = (1/2) * (cos(2 * pi * (4 -440) * t) + cos(2 * pi * (4+ 440)*t));
figure(9);
plot(t,x3s_t);
xlabel("t(secs)");
ylabel("x3s(t)");
title("Plot of x3s(t)");
%-----------------------------------------------
%% Part3 %%
a=1825 %1600+45*5
%x_4t = cos(pi*a*t.^2);
%x_4t = cos(pi*a/2*t.^2);
x_4t = cos(pi*a*2*t.^2);
t = [0:1/8192:1]
figure(4);
plot(t,x_4t);
xlabel("t(secs)");
ylabel("x4(t)");
title("Plot of x4(t)");
sound(x_4t);
x_5t = cos(2*pi*(-(t.^2)*250+800*t+4000))
t = [0:1/8192:1]
figure(5);
plot(t,x_5t);
xlabel("t(secs)");
ylabel("x5(t)");
title("Plot of x5(t)");
sound(x_5t);
%% Part4 %%
t = [0:1/8192:1];
a=1825
%fi = 0;
%fi = pi/4;
%fi = pi/2;
%fi = 3*(pi/4);
fi = pi;
x_t = (1/2)*(cos(2*pi*a*t + fi));
```

```
figure(100);
plot(t,x_t);
xlabel("t(secs)");
ylabel("x(t)");
title("Plot of x(t)");
sound(x_t);
```