ECS330: EECS Laboratory II

NAME: Sugandh Mittal
ROLL NO: 20278

Brief Description

Y = fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.
Discrete Fourier Transform (DFT) is used to find the Fourier transform of periodic discrete time signals. These yield signal and Fourier coefficients both are periodic and are represented in discrete in both time and frequency domain.

Discrete Fourier Transform	Inverse Discrete Fourier Transform
$X[k] = X\left(e^{j\frac{2\pi}{N}k}\right) = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn} \; for\ k = 0\ to\ N-1$	$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}kn}$

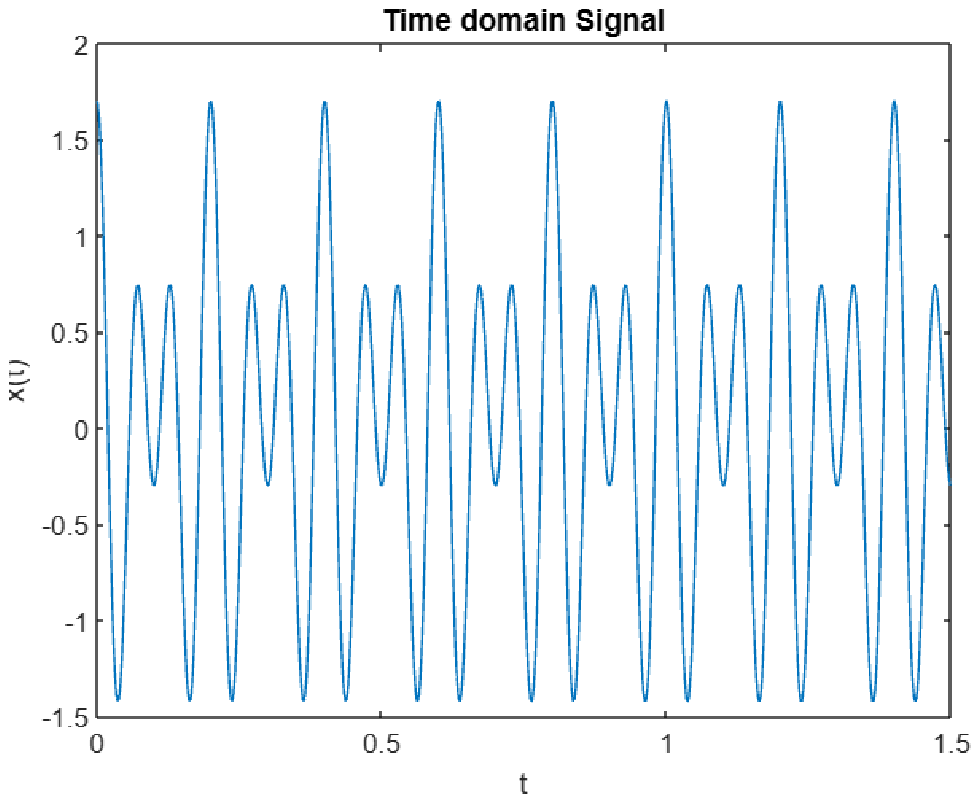
Question:1

a) For a1=0.7, a2=1, f1=10, f2=15
Our sampling frequency is 1000Hz.

i) Code:

```
Ts=1/(1000); %Sampling Time
t=0:Ts:1-Ts;
s=0.7*cos(2*pi*10*t)+cos(2*pi*15*t);
figure(1) %Plotting in time domain
plot(s);
xlabel('t');
ylabel('X(t)');
title('Time domain Signal')
L=length(s);
f = (1/Ts)*(0:(L/2))/L;
S=fft(s);
S_mag=abs(S);
figure(2) %Plotting in frequency domain (bins)
plot(S_mag);
title("Magnitude Spectrum of X(t)")
xlabel("f (Hz)")
ylabel("|X(f)|")
S_phase=angle(S);
figure
plot(S_phase)
xlabel('Frequency (Hz)');
ylabel('Phase Spectrum');
title('Phase Spectrum vs f')
S_phase(50)
S_phase(100)
```

ii) Output

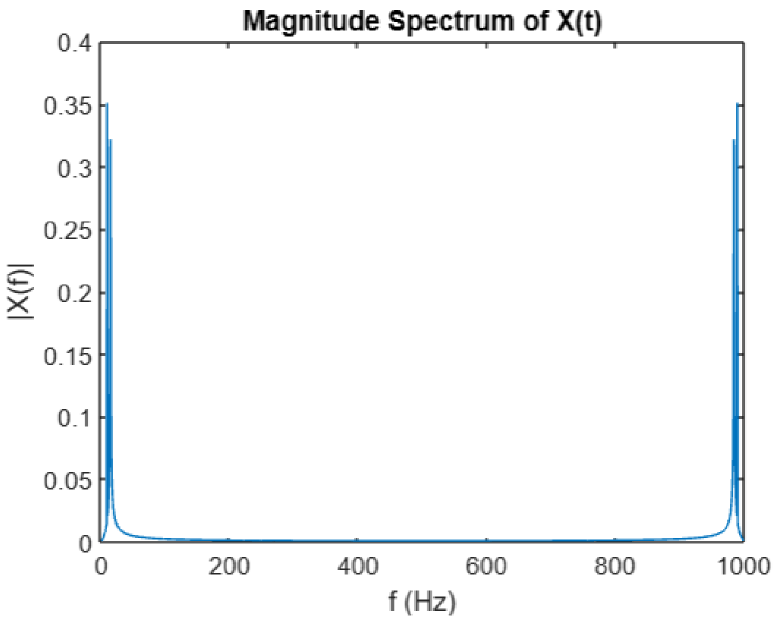


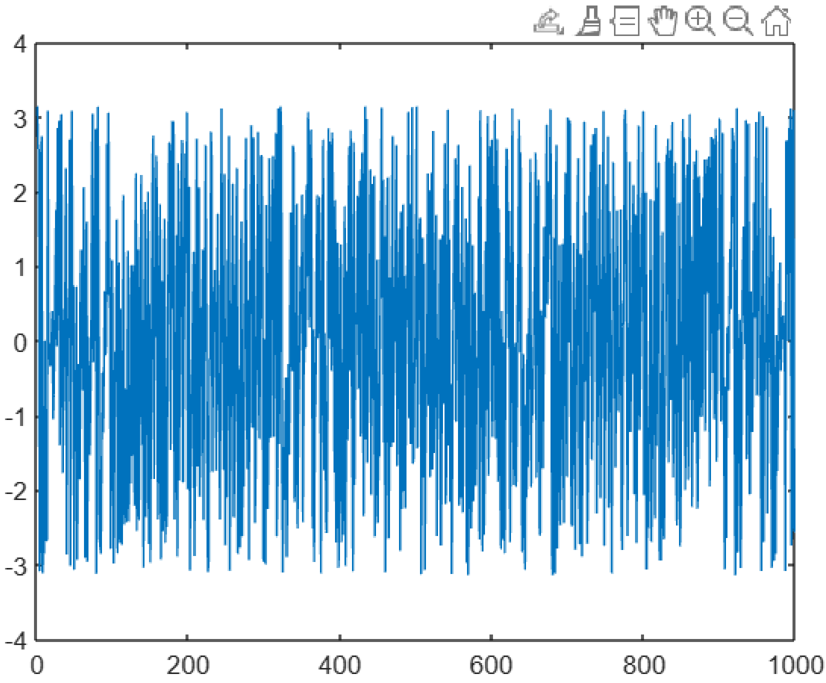
b) Magnitude and Phase Spectrum

i) Code:

```
Written in (a)
```

ii) Output:





Magnitude Spectrum at points other than f1 and f2 are:

a) At f= 50 Hz
   i) Magnitude=`0.008`
   ii) Phase= `-3.061`
b) At f= 500 Hz
   i) Magnitude=`0.0033`
   ii) Phase= `1.0775`

Interpretation:

No, the phase spectrum at points other than f1 and f2 is not very reliable because the phase spectrum is noisy stemming from the fact that using the FFT function computes inverse tangents from the ration of imaginary part to real part of the FFT result, hence there is a chance that even a small decimal error made while rounding off might amplify and manifest as an incorrect phase information.

For $\Theta_1=\Theta_2=0$, the phase is not defined

```
Array indices must be positive integers or logical values.
```

Inference:

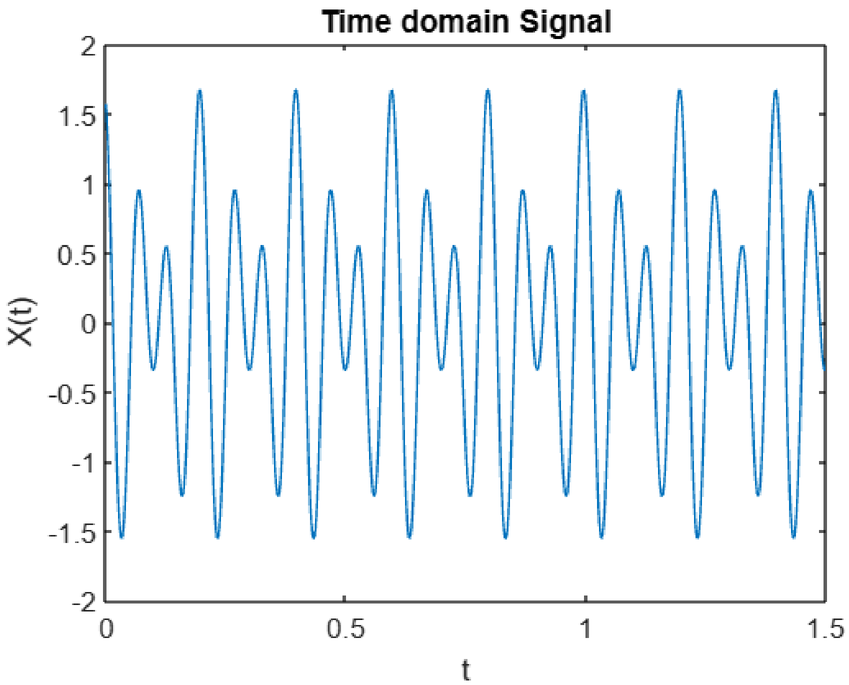
1. In the magnitude spectrum we see spikes both in the left and the right-hand side. Both of them are mirror images of each other. Each spike in the graph indicates the sinusoidal or frequency component of the signal.

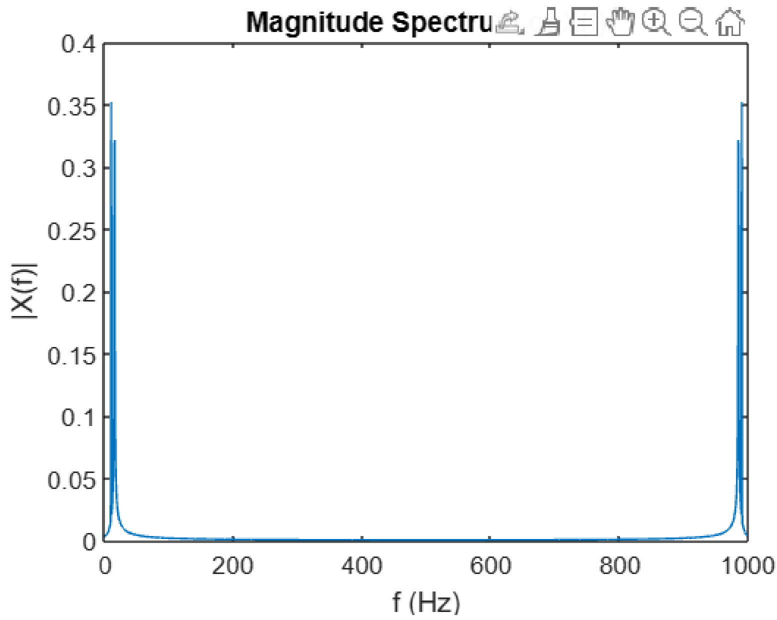
c) If $\Theta_1=0.5$ $\Theta_2=0.3$
   Repeating all the steps done above:

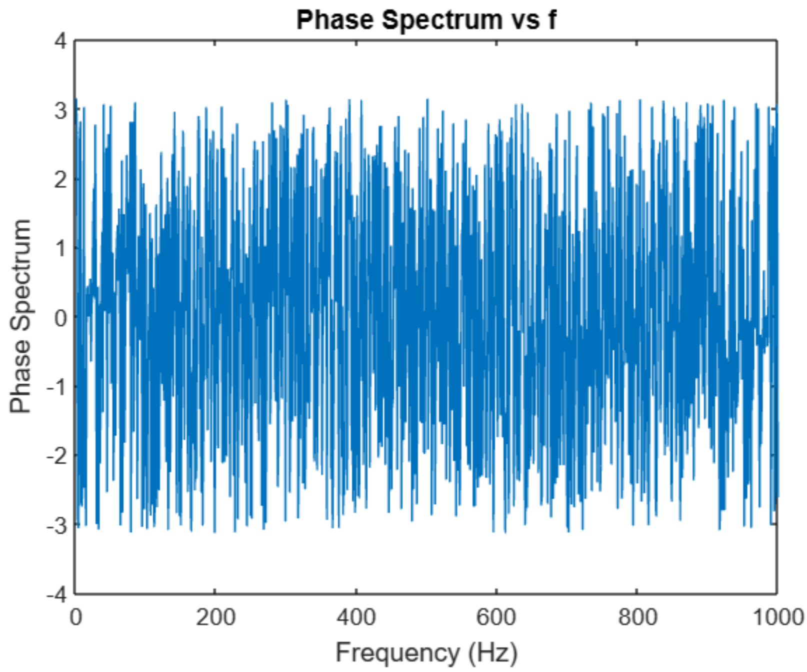
   Code:
```
Ts=1/(1000); %Sampling Time
t=0:Ts:1-Ts;
s=0.7*cos(2*pi*10*t+0.5)+cos(2*pi*15*t+0.3);
figure(1) %Plotting in time domain
plot(s);
```

```
xlabel('t');
ylabel('X(t)');
title('Time domain Signal')
L=length(s);
f = (1/Ts)*(0:(L/2))/L;
S=fft(s);
S_mag=abs(S);
figure(2) %Plotting in frequency domain (bins)
plot(S_mag);
title("Magnitude Spectrum of X(t)")
xlabel("f (Hz)")
ylabel("|X(f)|")
S_phase=angle(S);
figure
plot(S_phase)
xlabel('Frequency (Hz)');
ylabel('Phase Spectrum');
title('Phase Spectrum vs f')
S_phase(50)
S_phase(100)
```

Output:







Magnitude Spectrum at points other than f1 and f2 are:

d) At f= 50 Hz
   i) Magnitude=`0.0082`
   ii) Phase= `3.0375`

e) At f= 100 Hz
   i) Magnitude=`0.0034`
   ii) Phase= `1.6190`

For Θ_1=0.5, Θ_2=0.3, the phase is not defined

```
Array indices must be positive integers or logical values.
```

Question:2

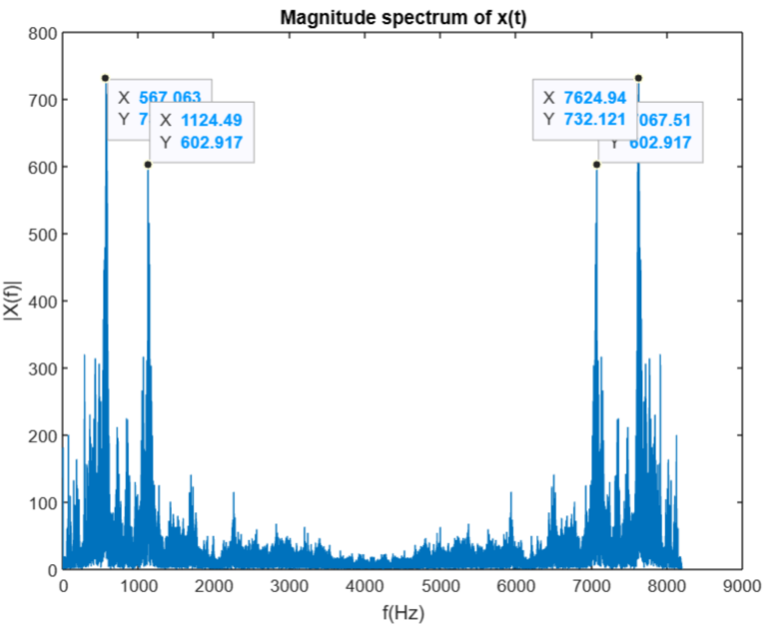
The following are the codes and graphs for the reading and plotting the magnitude spectrum of three audio signals named as audio1, audio2 and audio3.

Code:

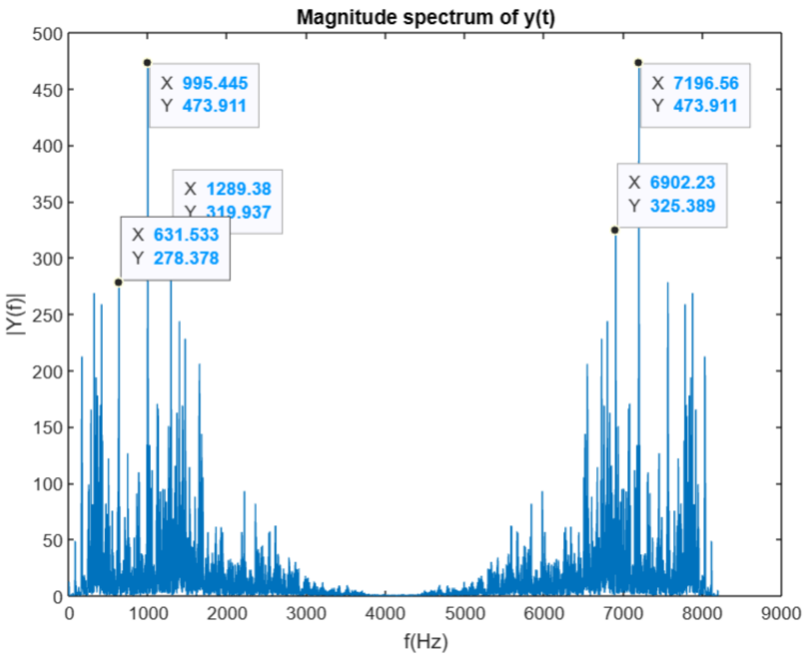
```
load handel.mat
[x,Fs1]=audioread("audio1.wav");
[y,Fs2]=audioread("audio2.wav");
[z,Fs3]=audioread("audio3.wav");
L1=length(x);
t1=0:1/Fs1:(L1-1)/Fs1;
f1 = Fs1*(0:(L1-1))/L1;
L2=length(y);
t2=0:1/Fs3:(L2-1)/Fs2;
f2 = Fs2*(0:(L2-1))/L2;
L3=length(z);
t3=0:1/Fs3:(L3-1)/Fs3;
f3 = Fs3*(0:(L3-1))/L3;
X=fft(x,L1);
X_mag=abs(X);
Y=fft(y,L2);
Y_mag=abs(Y);
Z=fft(z,L3);
Z_mag=abs(Z);
figure(1)
plot(f1,X_mag)
title("Magnitude spectrum of x(t)");
xlabel("f(Hz)");
ylabel("|X(f)|");
figure(2)
plot(f2,Y_mag)
title("Magnitude spectrum of y(t)");
xlabel("f(Hz)");
ylabel("|Y(f)|");
figure(3)
plot(f3,Z_mag)
title("Magnitude spectrum of z(t)");
xlabel("f(Hz)");
ylabel("|Z(f)|");
```

Output:

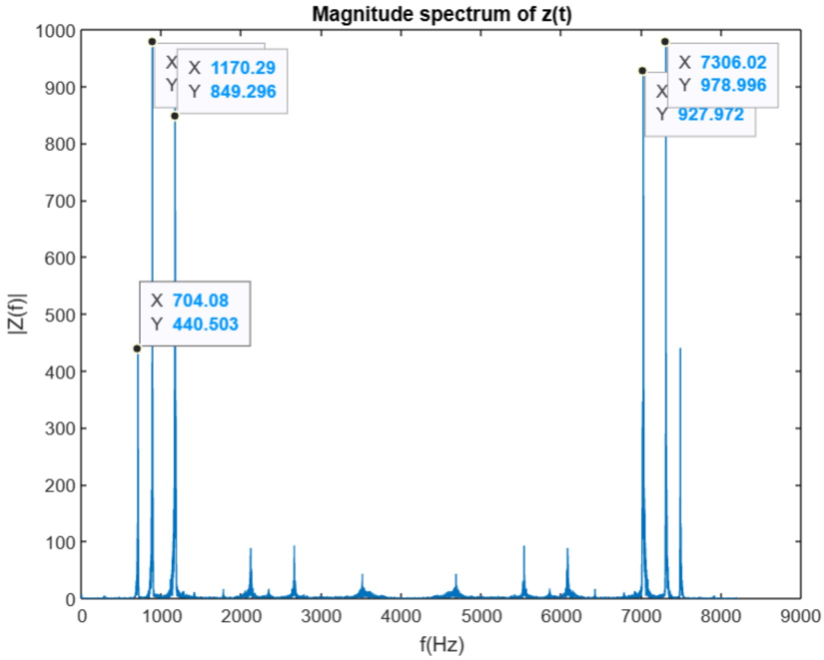
a) For Audio-1



b) For Audio-2



c) For Audio-3



b) The five most prominent frequencies in each of the above audio signals are labeled on the plots (X-coordinates). They are listed below (they correspond to the prominent peaks of the curves)

Audio-1: 1065 , 1125 ,1137, 7067 , 7624

Audio-2:631, 995, 1289, 6902, 7197

Audio-3: 704, 886 , 1170, 7022, 7306

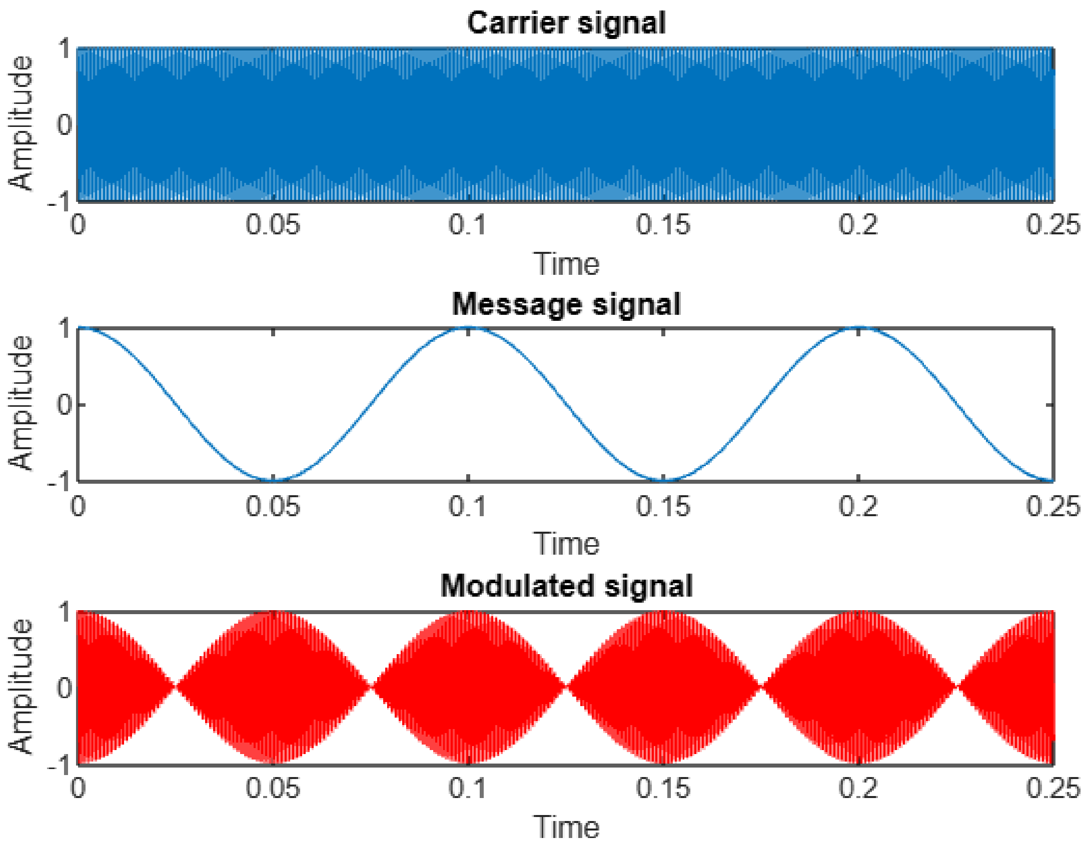
Question:3

a) Message, Carrier and Modulated Signal

i) Code:

```
%% Time specifications:
Fs = 8000;                % samples per second
dt = 1/Fs;                % seconds per sample
StopTime = 0.25;          % seconds
t1 = (0:dt:StopTime-dt)';
%% Equations
y1=cos(2*pi*1000*t1); %Carrier Signal
y2=cos(2*pi*10*t1); %Message Signal
eq=(y2).*(y1); %Amplitude Modulated Signal
%% Plots
subplot(311)
plot(t1,y1)
xlabel('Time')
ylabel('Amplitude')
title('Carrier signal')
subplot(312)
plot(t1,y2)
xlabel('Time')
ylabel('Amplitude')
title('Message signal')
subplot(313)
plot(t1,eq)
plot(t1,eq,'r')
xlabel('Time')
ylabel('Amplitude')
title('Modulated signal')
```

ii) Output



b) Magnitude Spectrum

i) Code:

1) Message Signal

```
%% Time specifications:
Fs = 8000;                % samples per second
dt = 1/Fs;                % seconds per sample
StopTime = 0.25;          % seconds
t1 = (0:dt:StopTime-dt)';
%% Equations
y1=cos(2*pi*1000*t1); %Carrier Signal
y2=cos(2*pi*10*t1); %Message Signal
eq=(y2).*(y1); %Amplitude Modulated Signal
%% Equations
L=length(y2);
Y = fft(y2);
P2 = abs(Y/L);
plot(P2)
title("Magnitude Spectrum of Message Signal")
xlabel("f (Hz)")
ylabel("|y2(f)|")
```

2) Carrier Signal

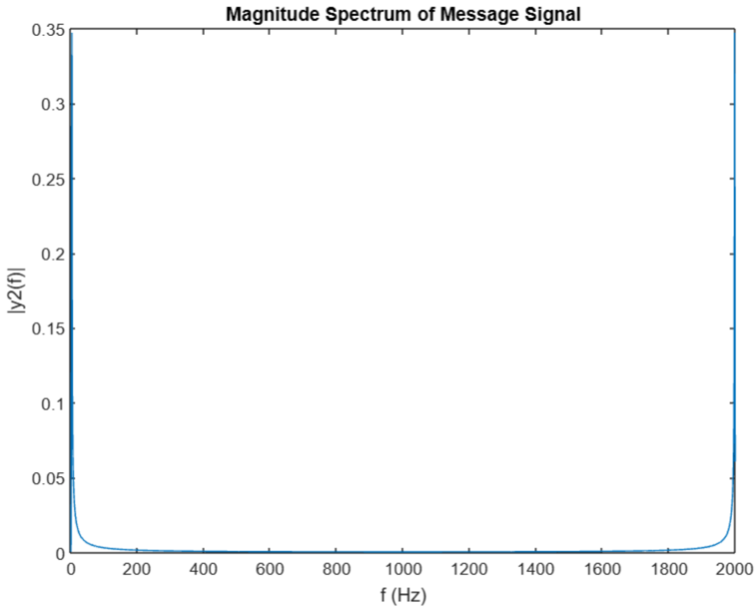
```
%% Equations
L=length(y1);
Y = fft(y1);
P2 = abs(Y/L);
plot(P2)
title("Magnitude Spectrum of Carrier Signal")
xlabel("f (Hz)")
ylabel("|y1(f)|")
```

3) Modulated Signal

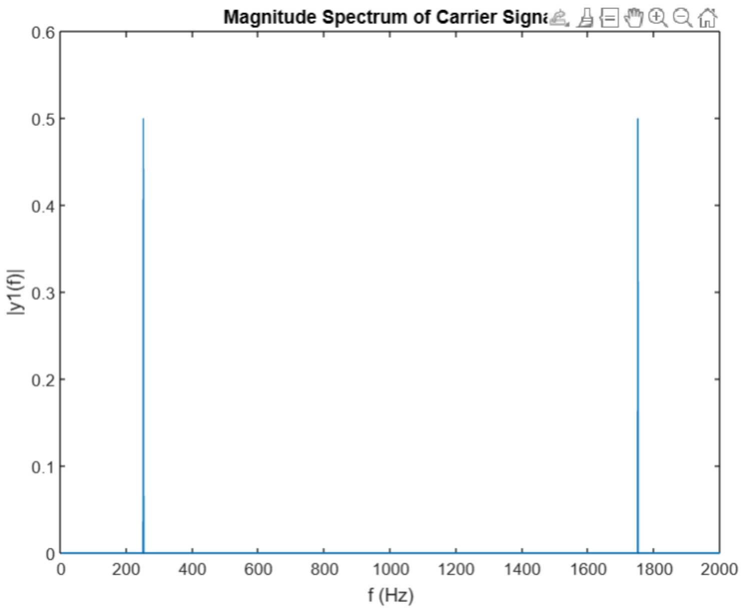
```
%% Equations
L=length(eq);
Y = fft(eq);
P2 = abs(Y/L);
plot(P2)
title("Magnitude Spectrum of Modulated Signal")
xlabel("f (Hz)")
ylabel("|eq(f)|")
```

ii) Output

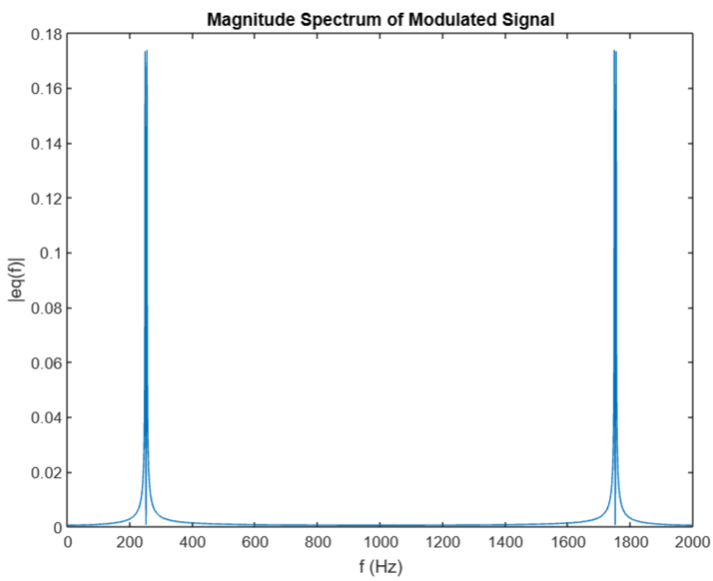
1) Message Signal



2) Carrier Signal



3) Modulated Signal



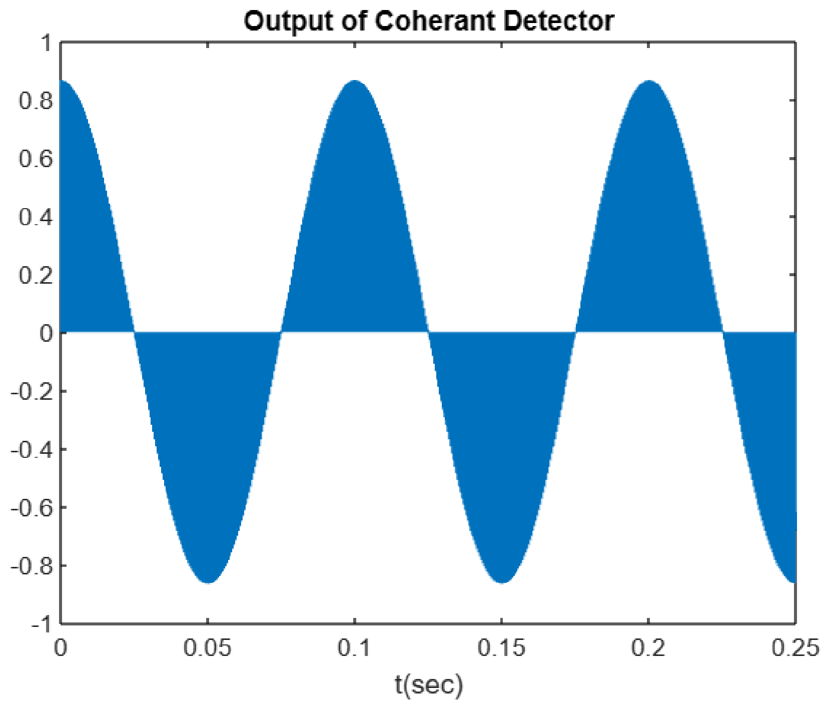
c) Coherent Detector

i) Code

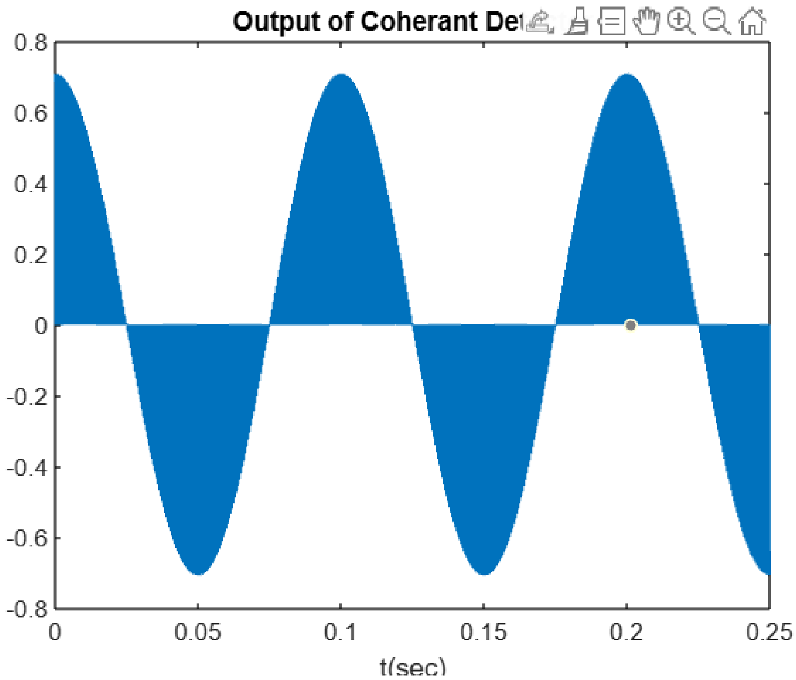
```
%% Time specifications:
Fs = 8000;                  % samples per second
dt = 1/Fs;                  % seconds per sample
StopTime = 0.25;            % seconds
t1 = (0:dt:StopTime-dt)';
%% Equations
y1=cos(2*pi*1000*t1); %Carrier Signal
y2=cos(2*pi*10*t1); %Message Signal
eq=(y2).*(y1); %Amplitude Modulated Signal
lo=cos(2*pi*1000*t1+pi/4);
loo=eq.*lo;
%% Equations
plot(t1,loo)
title("Output of Coherent Detector")
xlabel("t(sec)")
```

ii) Output

1) For the phase of Pi/6 in local oscillator



2) For the phase of Pi/4 in local oscillator



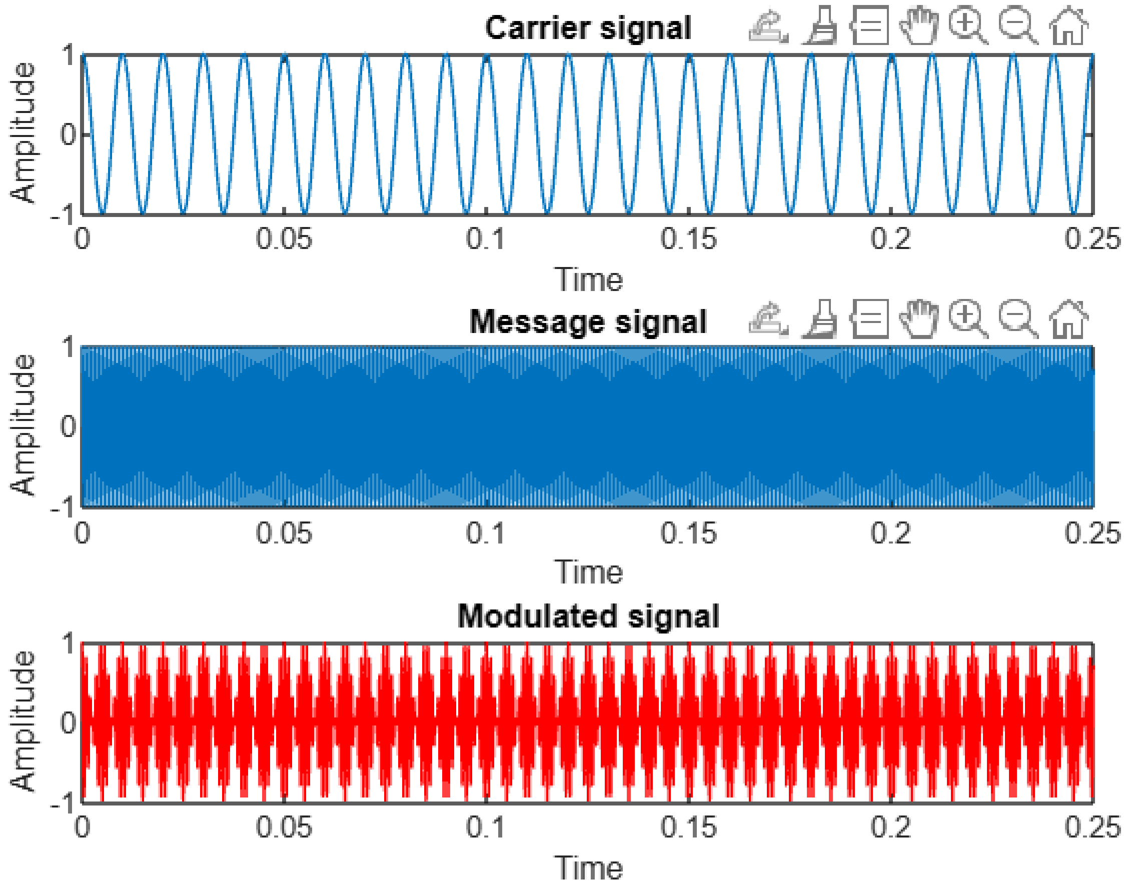
d) Fm=1000, Fc=100 (Fm>Fc)

i) Message, Carrier and Modulated Signal

1) Code:

```
%% Time specifications:
Fs = 8000;                % samples per second
dt = 1/Fs;                % seconds per sample
StopTime = 0.25;          % seconds
t1 = (0:dt:StopTime-dt)';
%% Equations
y1=cos(2*pi*100*t1); %Carrier Signal
y2=cos(2*pi*1000*t1); %Message Signal
eq=(y2).*(y1); %Amplitude Modulated Signal
%% Plots
subplot(311)
plot(t1,y1)
xlabel('Time')
ylabel('Amplitude')
title('Carrier signal')
subplot(312)
plot(t1,y2)
xlabel('Time')
ylabel('Amplitude')
title('Message signal')
subplot(313)
plot(t1,eq)
plot(t1,eq,'r')
xlabel('Time')
ylabel('Amplitude')
title('Modulated signal')
```

ii) Output



ii) Magnitude Spectrum

1) Code:

(a) Message Signal

```
%% Time specifications:
Fs = 8000;                  % samples per second
dt = 1/Fs;                  % seconds per sample
StopTime = 0.25;            % seconds
t1 = (0:dt:StopTime-dt)';
%% Equations
y1=cos(2*pi*100*t1); %Carrier Signal
y2=cos(2*pi*1000*t1); %Message Signal
eq=(y2).*(y1); %Amplitude Modulated Signal
%% Equations
L=length(y2);
Y = fft(y2);
P2 = abs(Y/L);
plot(P2)
title("Magnitude Spectrum of Message Signal")
xlabel("f (Hz)")
ylabel("|y2(f)|")
```

(b) Carrier Signal

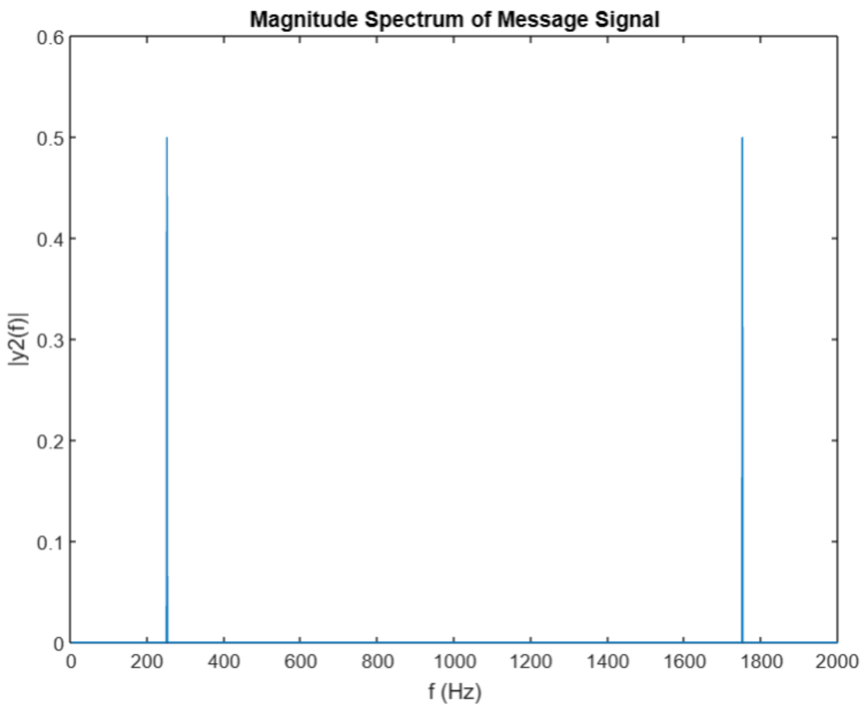
```
L=length(y1);
Y = fft(y1);
P2 = abs(Y/L);
plot(P2)
title("Magnitude Spectrum of Carrier Signal")
xlabel("f (Hz)")
ylabel("|y1(f)|")
```

(c) Modulated Signal

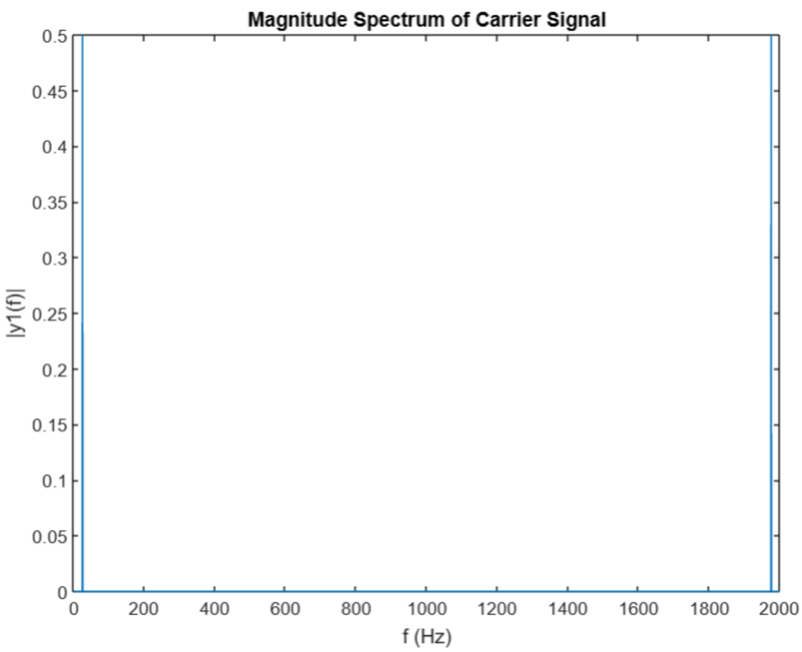
```
L=length(eq);
Y = fft(eq);
P2 = abs(Y/L);
plot(P2)
title("Magnitude Spectrum of Modulated Signal")
xlabel("f (Hz)")
ylabel("|eq(f)|")
```

2) Output

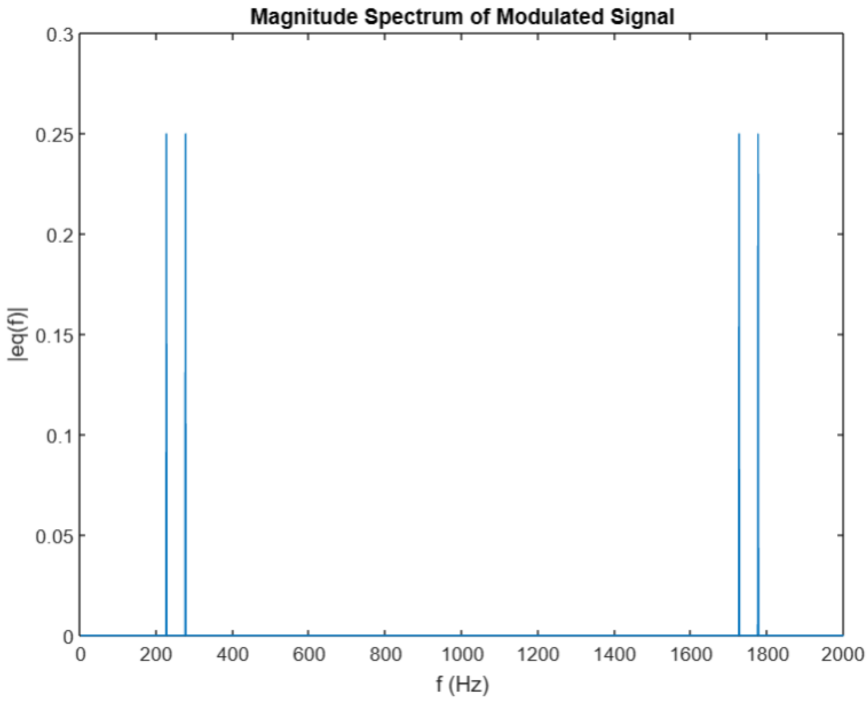
(a) Message Signal



(b) Carrier Signal



(c) Modulated Signal



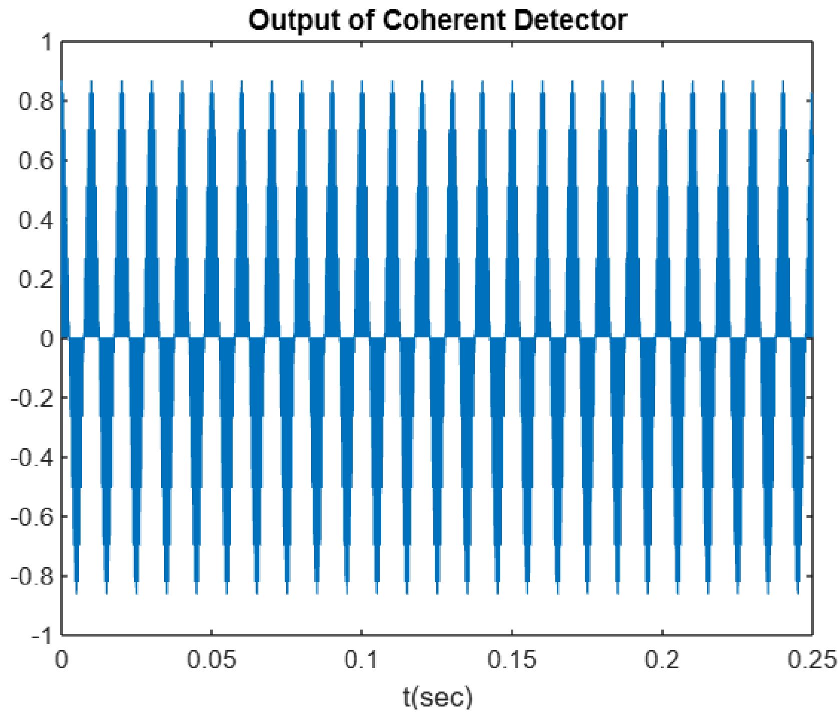
iii) Coherent Detector:
The output signal for this will be a signal from the local oscillator multiplied by the modulated signal since we don’t have to pass this through the low pass filter.

1) Code

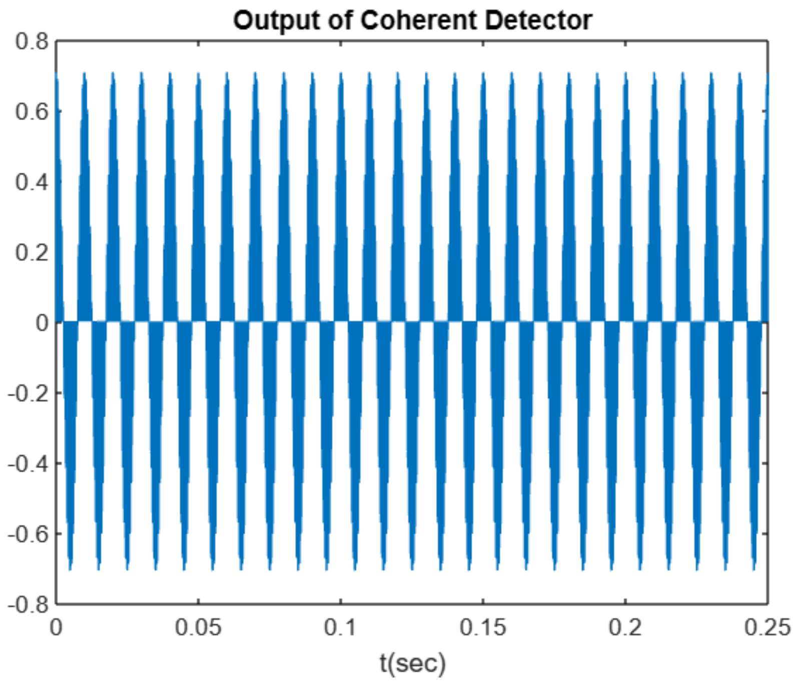
```
%% Time specifications:
Fs = 8000;                   % samples per second
dt = 1/Fs;                   % seconds per sample
StopTime = 0.25;             % seconds
t1 = (0:dt:StopTime-dt)';
%% Equations
y1=cos(2*pi*100*t1); %Carrier Signal
y2=cos(2*pi*1000*t1); %Message Signal
eq=(y2).*(y1); %Amplitude Modulated Signal
lo=cos(2*pi*1000*t1+pi/6);
loo=eq.*lo;
%% Equations
plot(t1,loo)
title("Output of Coherent Detector")
xlabel("t(sec)")
```

2) Output

(a) For the phase of Pi/6 in local oscillator



(b)For the phase of Pi/4 in local oscillator



INFERENCE

Here we see that when fm < fc the signal undergoes amplitude modulation (which is the required criteria) successfully. Also, when the above signal is made DSB-SC and passed through coherent detector we see that demodulation occurs and the coherent detector follows the same phase as the message signal i.e., the envelope of the coherent detector is the same as a message signal implying that the message can be retrieved back with all its contents as it was sent if we further pass it to a low pass filter.

We notice that there is some distortion in the coherent detected signal and original message signal for fm > fc. This is because the message is modulated with less carrier frequency which over modulated and phase reverses and when the DSB-SC of the signal is passed through the coherent detector the signal envelope is similar to carrier frequency which implies that maximum message is been destroyed (or change or distorted) while modulating and hence the output of coherent detector