Assignment-2 (Via MATLAB)

By :- Kwanit Gupta
Roll-No. B19EE046

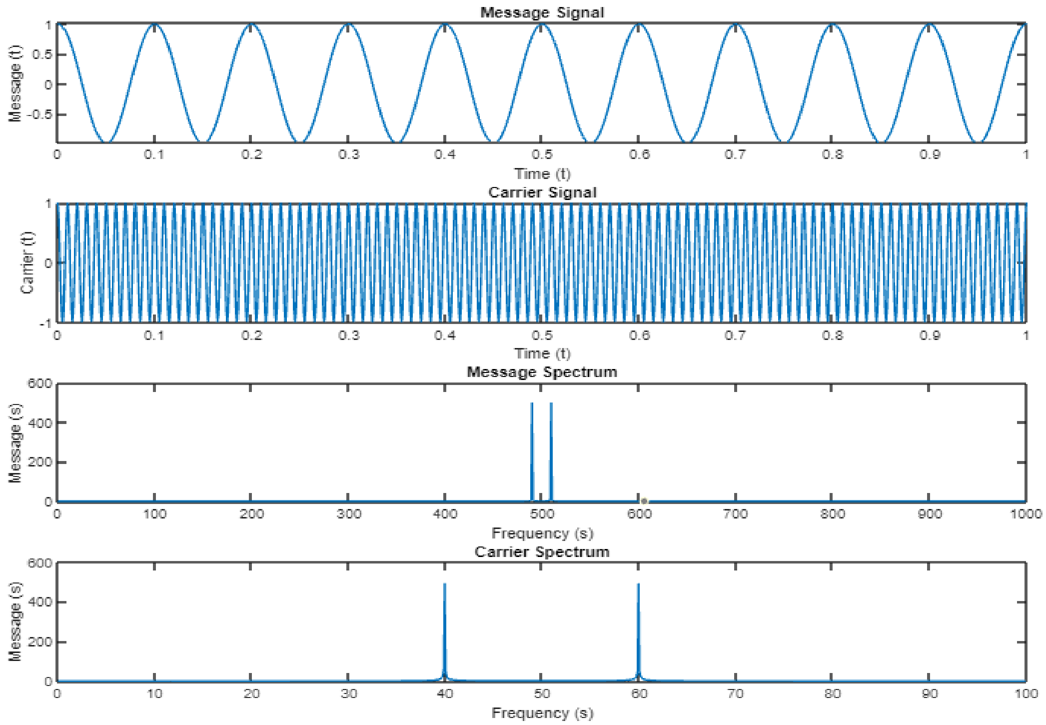
Q.1. Consider the amplitude modulated signal x(t) = Ac*(1 + kam(t))*cos 2πfct. Assume appropriate values of the parameters and a sinusoidal message signal. Show the effect of variation in the modulation index values {0.5, 1, 1.5} on the modulated signal and the demodulated signal in each case. Sketch the time domain waveforms and the spectrum of the message signal, modulated signal and the recovered message signal for each case.

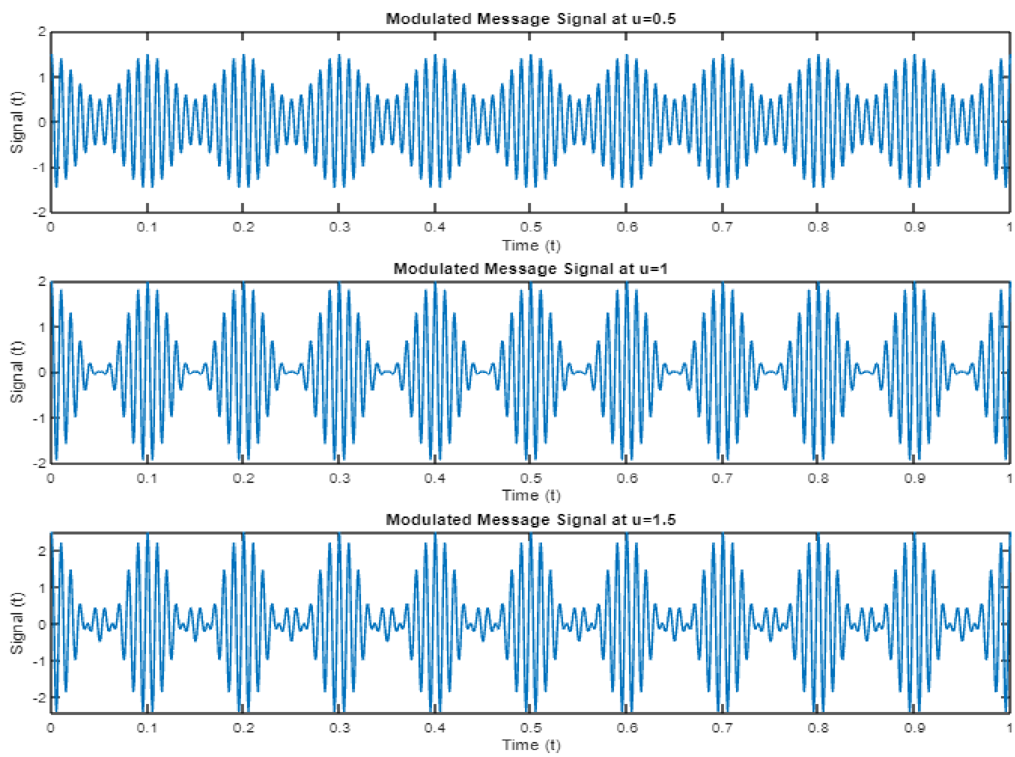
```
% Time and Frequency for sample simulation

Ts = 0.001;
fs = 1/Ts;
t = (0:Ts:1);

% Amplitudes, Frequency and Sensitivity Parameters for Waveforms

Am = 1;                          % Message Amplitude
Ac = 1;                          % Carrier Amplitude
fc = 100;                        % Carrier Frequency
ka=[0.5,1,1.5];                  % Amplitude Sensitivity = Modulation Index
fm = 10;                         % Message Frequency

% Message Waveform and Spectrum Plots

mt = Am*cos(2*pi*fm*t);          % Message Signal
ct = Ac*cos(2*pi*fc*t);          % Carrier Signal
ms = fft(mt);                    % Message Spectrum
ms = fftshift(abs(ms));
fms = (0:length(ms)-1)*fs/length(ms);
cs = fft(ct);                    % Carrier Spectrum
cs = fftshift(abs(cs));
fcs = (0:length(cs)-1)*fc/length(cs);
figure();
subplot(4,1,1);
plot(t,mt);
title("Message Signal")
xlabel("Time (t)")
ylabel("Message (t)")
subplot(4,1,2);
plot(t,ct);
title("Carrier Signal")
xlabel("Time (t)")
ylabel("Carrier (t)")
subplot(4,1,3);
plot(fms,ms);
title("Message Spectrum")
```

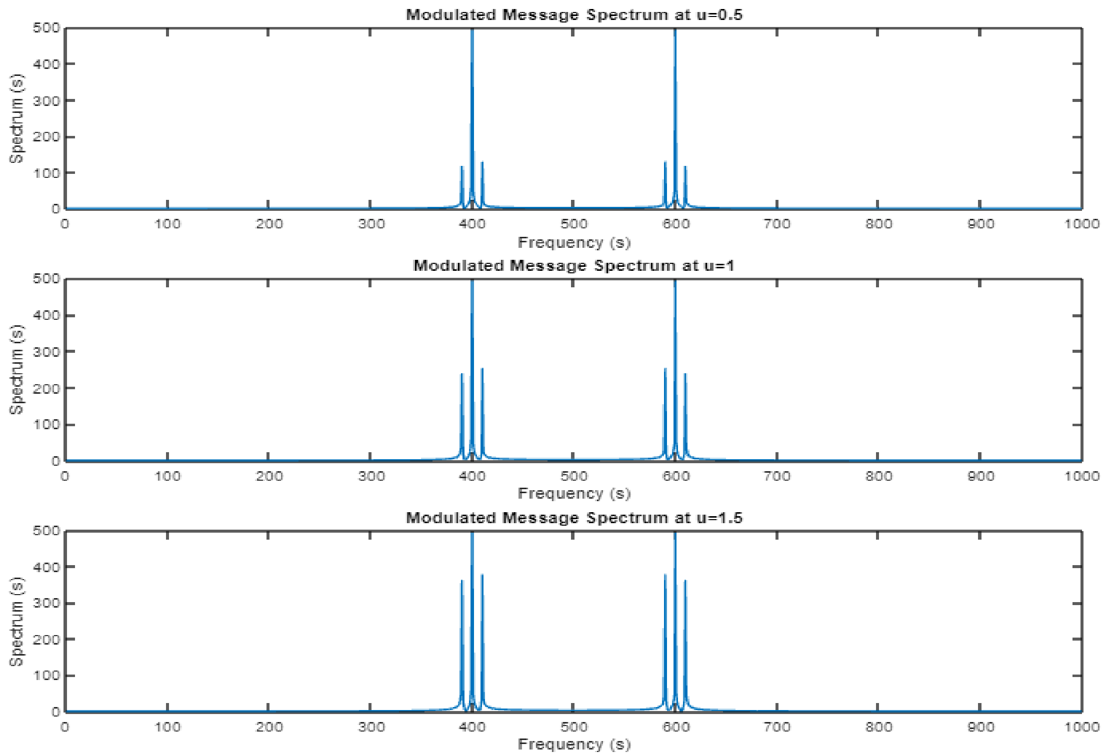
```
xlabel("Frequency (s)")
ylabel("Message (s)")
subplot(4,1,4);
plot(fcs,cs);
title("Carrier Spectrum")
xlabel("Frequency (s)")
ylabel("Message (s)")

% Amplitude Modulation for Resultant Waveforms and their Spectrums

xt1 = Ac*(1+(ka(1)*mt)).*cos(2*pi*fc*t);          % Modulated Signal at modulation index=0.5
xt2 = Ac*(1+(ka(2)*mt)).*cos(2*pi*fc*t);          % Modulated Signal at modulation index=1
xt3 = Ac*(1+(ka(3)*mt)).*cos(2*pi*fc*t);          % Modulated Signal at modulation index=1.5
ms1 = fft(xt1);                                   % Modulated Spectrum at modulation index=0.5
ms2 = fft(xt2);                                   % Modulated Spectrum at modulation index=1
ms3 = fft(xt3);                                   % Modulated Spectrum at modulation index=1.5
ms1 = fftshift(abs(ms1));
ms2 = fftshift(abs(ms2));
ms3 = fftshift(abs(ms3));
fms1 = (0:length(ms1)-1)*fs/length(ms1);
fms2 = (0:length(ms2)-1)*fs/length(ms2);
fms3 = (0:length(ms3)-1)*fs/length(ms3);
figure();
subplot(3,1,1);
plot(t,xt1);
title("Modulated Message Signal at u=0.5")
xlabel("Time (t)")
ylabel("Signal (t)")
subplot(3,1,2);
plot(t,xt2);
title("Modulated Message Signal at u=1")
xlabel("Time (t)")
ylabel("Signal (t)")
subplot(3,1,3);
plot(t,xt3);
title("Modulated Message Signal at u=1.5")
xlabel("Time (t)")
ylabel("Signal (t)")
figure();
subplot(3,1,1);
plot(fms1,ms1);
title("Modulated Message Spectrum at u=0.5")
xlabel("Frequency (s)")
ylabel("Spectrum (s)")
subplot(3,1,2);
plot(fms2,ms2);
title("Modulated Message Spectrum at u=1")
xlabel("Frequency (s)")
ylabel("Spectrum (s)")
```

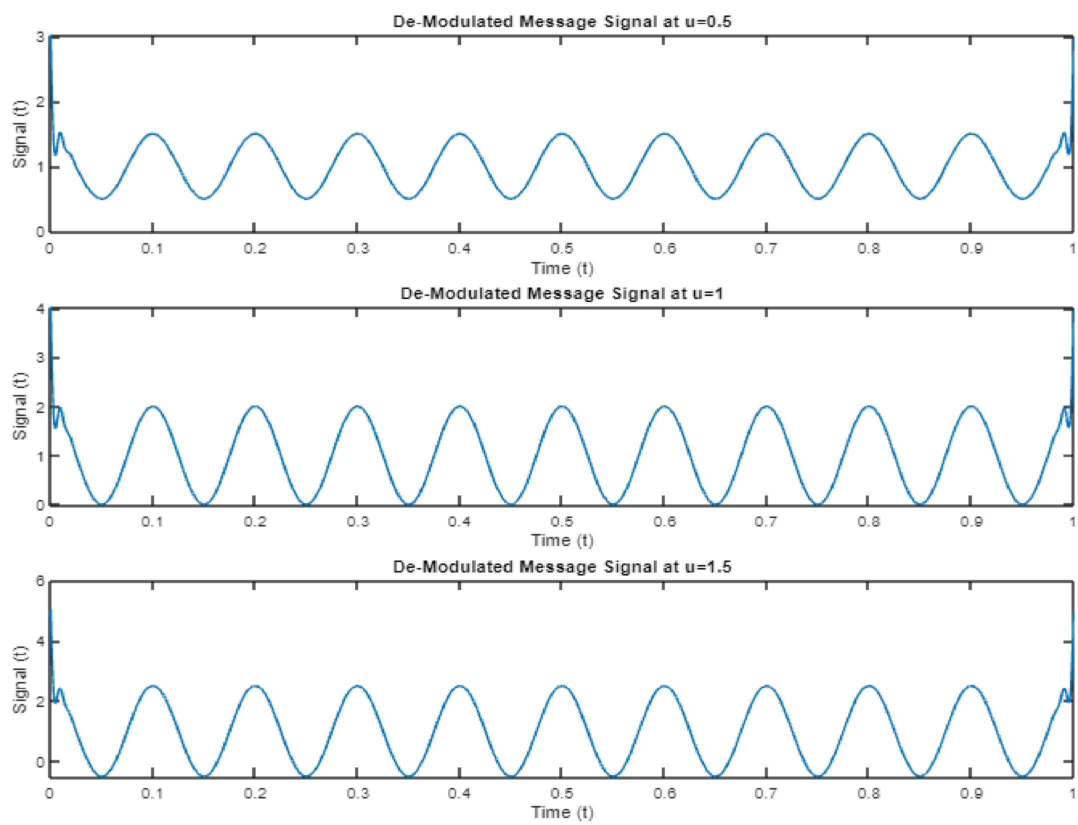
```
subplot(3,1,3);
plot(fms3,ms3);
title("Modulated Message Spectrum at u=1.5")
xlabel("Frequency (s)")
ylabel("Spectrum (s)")

% Amplitude Demodulation Scheme for waveforms and their spectrums

xr1 = amdemod(xt1,fc,fs);                         % Demodulated Signal at modulation index=0.5
xr2 = amdemod(xt2,fc,fs);                         % Demodulated Signal at modulation index=1
xr3 = amdemod(xt3,fc,fs);                         % Demodulated Signal at modulation index=1.5
figure();
subplot(3,1,1);
plot(t,xr1);
title("De-Modulated Message Signal at u=0.5")
xlabel("Time (t)")
ylabel("Signal (t)")
subplot(3,1,2);
plot(t,xr2);
title("De-Modulated Message Signal at u=1")
xlabel("Time (t)")
ylabel("Signal (t)")
subplot(3,1,3);
plot(t,xr3);
title("De-Modulated Message Signal at u=1.5")
xlabel("Time (t)")
ylabel("Signal (t)")
mr1 = fft(xr1);                                   % Demodulated Spectrum at modulation index=0.5
mr2 = fft(xr2);                                   % Demodulated Spectrum at modulation index=1
mr3 = fft(xr3);                                   % Demodulated Spectrum at modulation index=1.5
mr1 = fftshift(abs(mr1));
mr2 = fftshift(abs(mr2));
mr3 = fftshift(abs(mr3));
fmr1 = (0:length(mr1)-1)*fs/length(mr1);
fmr2 = (0:length(mr2)-1)*fs/length(mr2);
fmr3 = (0:length(mr3)-1)*fs/length(mr3);
figure();
subplot(3,1,1);
plot(fmr1,mr1);
title("Demodulated Message Spectrum at u=0.5")
xlabel("Frequency (s)")
ylabel("Spectrum (s)")
subplot(3,1,2);
plot(fmr2,mr2);
title("Demodulated Message Spectrum at u=1")
xlabel("Frequency (s)")
ylabel("Spectrum (s)")
subplot(3,1,3);
plot(fmr3,mr3);
```

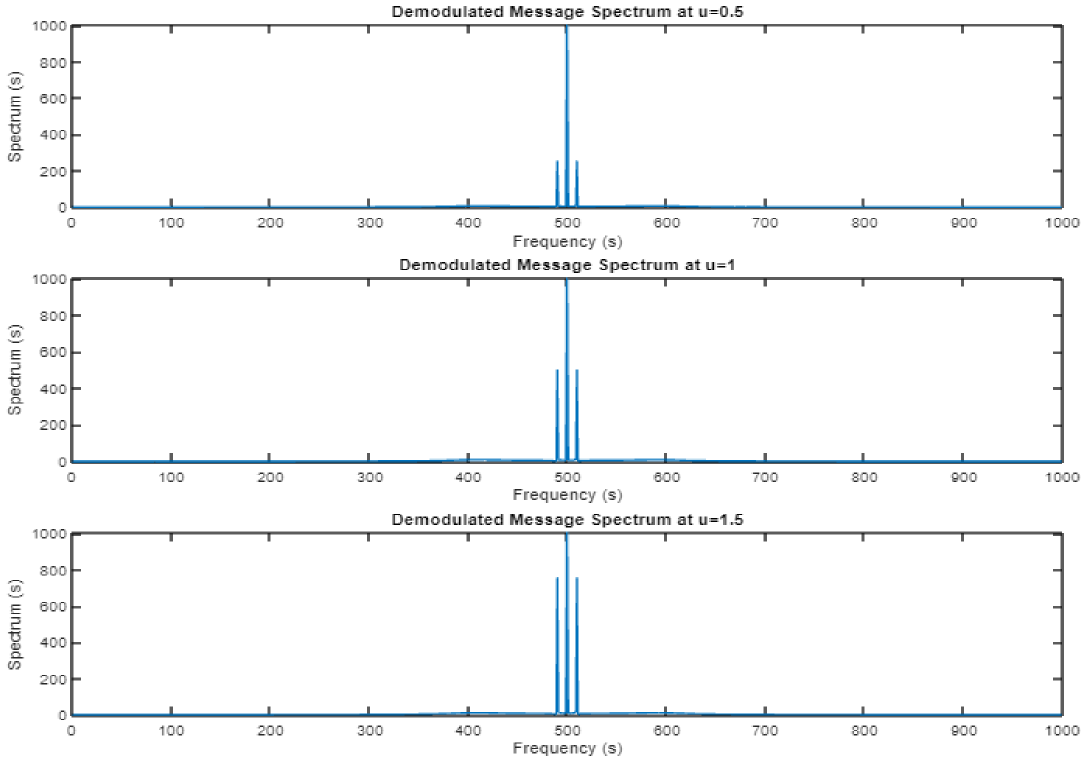
```
title("Demodulated Message Spectrum at u=1.5")
xlabel("Frequency (s)")
ylabel("Spectrum (s)")
```






Modulated Message Spectrum at u=0.5
Spectrum (s)
Frequency (s)
Modulated Message Spectrum at u=1
Spectrum (s)
Frequency (s)
Modulated Message Spectrum at u=1.5
Spectrum (s)
Frequency (s)


De-Modulated Message Signal at u=0.5
Signal (t)
Time (t)
De-Modulated Message Signal at u=1
Signal (t)
Time (t)
De-Modulated Message Signal at u=1.5
Signal (t)
Time (t)



Q.2. Consider a message signal m(t) = 2*cos(100t) + 18*cos(2000πt) and a carrier signal c(t) = 10*cos((10^6)t). Assume that the phase sensitivity parameter is 1 rad/V.

1. Plot the phase modulated signal in MATLAB.
2. Determine the bandwidth of the PM signal (in Hz).

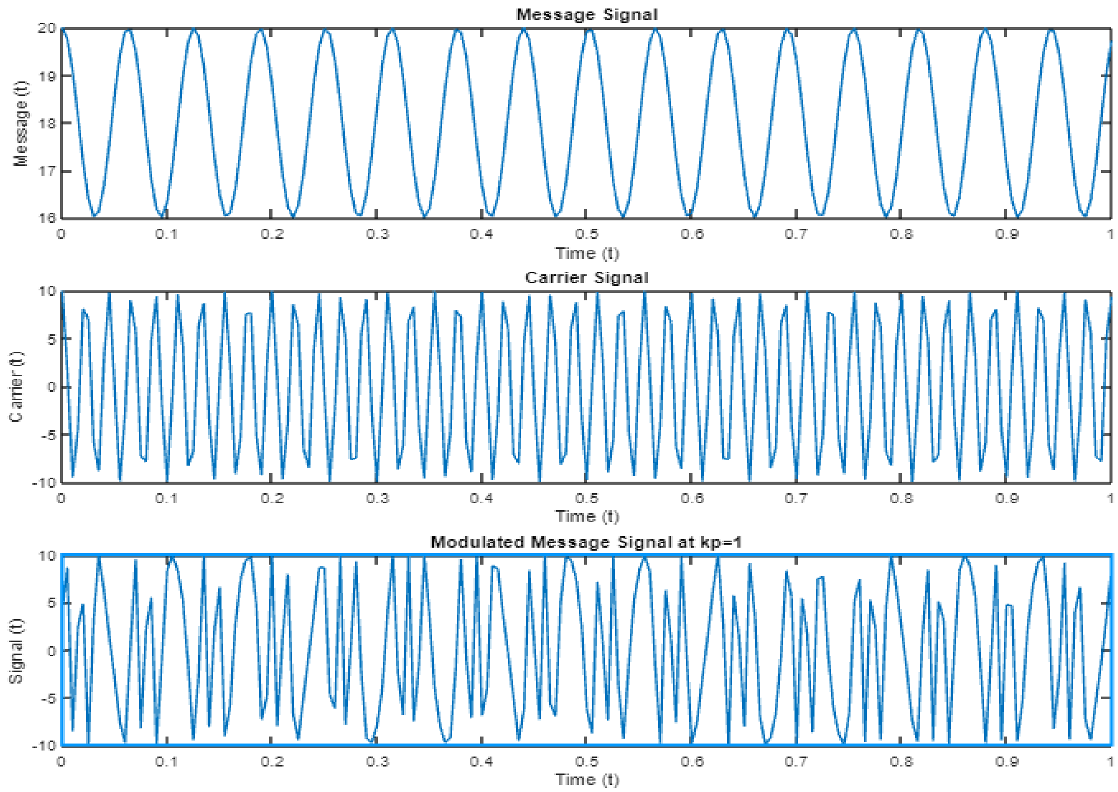
```
% Parameters for Q-2

t = (0:0.005:1);
f=1./t;
wc = 10^6;                              % Carrier Frequency in w
fc = wc/(2*pi);                         % Carrier Frequency in Hz
kp  = 1;                                % Phase Sensitivity Constant

% Message, Carrier and Modulated Signals with their Spectrums

m = 2*cos(100*t) + 18*cos(2000*pi*t);   % Message Signal
c = 10*cos(wc*t);                       % Carrier Signal
ms = fft(m);                            % Message Spectrum
ms = fftshift(abs(ms));
fms = (0:length(ms)-1).*f/length(ms);
cs = fft(c);                            % Carrier Spectrum
cs = fftshift(abs(cs));
fcs = (0:length(cs)-1).*fc/length(cs);
modulated = 10*cos(wc*t + kp*m);        % Modulated Signal
```

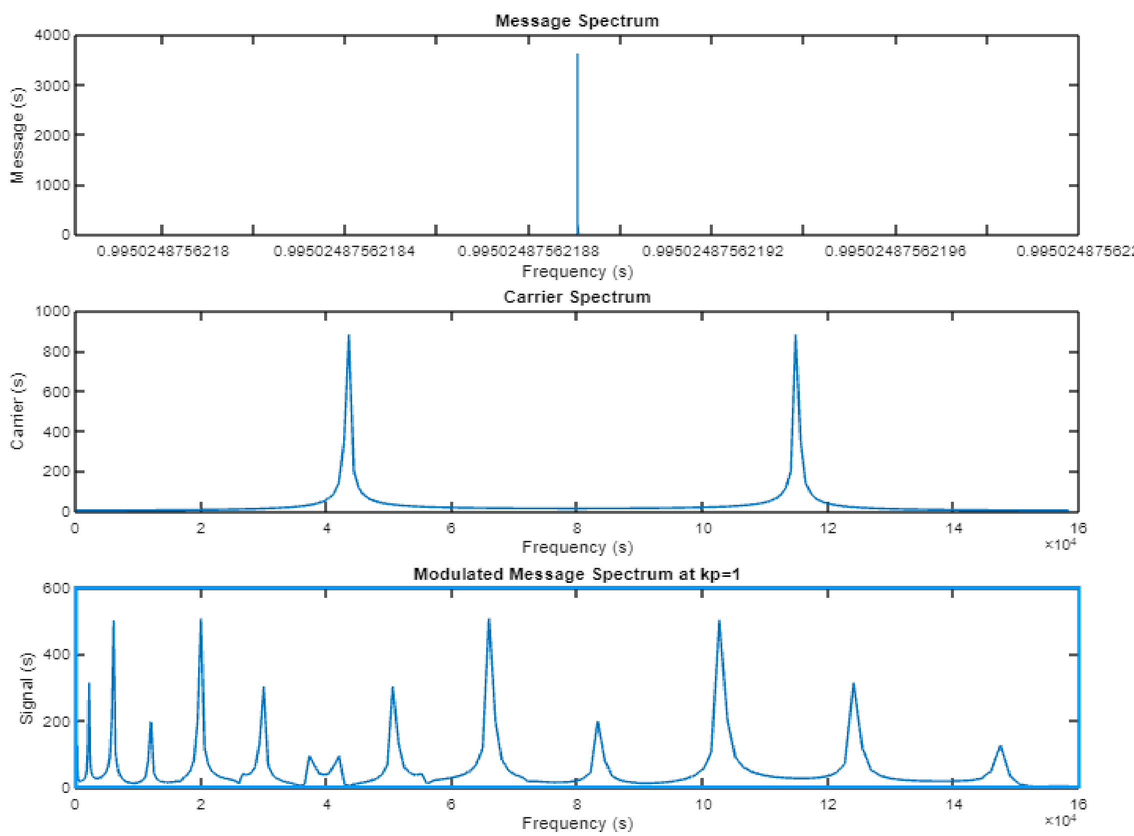
```
wmod = (wc*t + kp*m);
fmod = wmod/(2*pi);
mods = fft(modulated);                         % Modulated Spectrum
mods = fftshift(abs(mods));
fmods = (0:length(mods)-1).*fmod/length(mods);
figure();
subplot(3,1,1);
plot(t,m);
title("Message Signal")
xlabel("Time (t)")
ylabel("Message (t)")
subplot(3,1,2);
plot(t,c);
title("Carrier Signal")
xlabel("Time (t)")
ylabel("Carrier (t)")
subplot(3,1,3);
plot(t,modulated);
title("Modulated Message Signal at kp=1")
xlabel("Time (t)")
ylabel("Signal (t)")
figure();
subplot(3,1,1);
plot(fms,ms);
title("Message Spectrum")
xlabel("Frequency (s)")
ylabel("Message (s)")
subplot(3,1,2);
plot(fcs,cs);
title("Carrier Spectrum")
xlabel("Frequency (s)")
ylabel("Carrier (s)")
subplot(3,1,3);
plot(fmods,mods);
title("Modulated Message Spectrum at kp=1")
xlabel("Frequency (s)")
ylabel("Signal (s)")
```

Bw = 2(kpAm + 1)fm

= 2(1(18) +1)*2000pi

= 38kHz

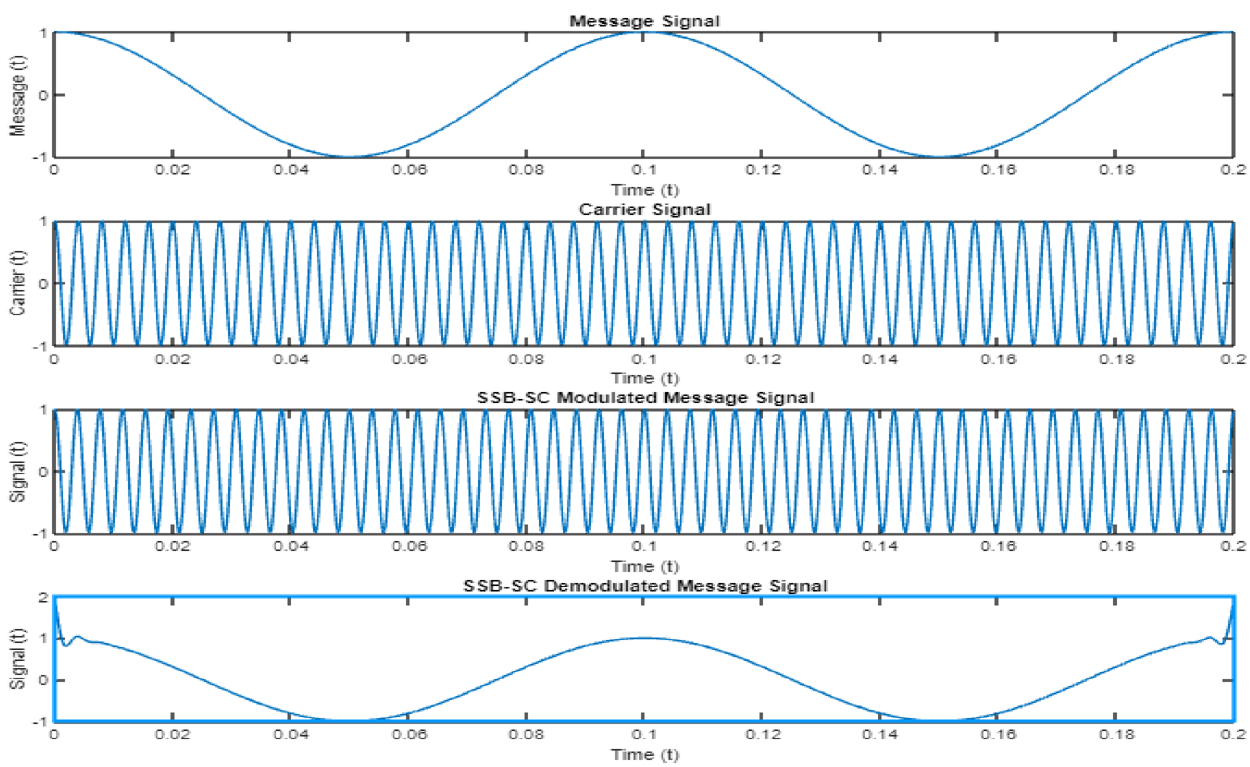

Message Signal
Message (t)
Time (t)
Carrier Signal
Carrier (t)
Time (t)
Modulated Message Signal at kp=1
Signal (t)
Time (t)

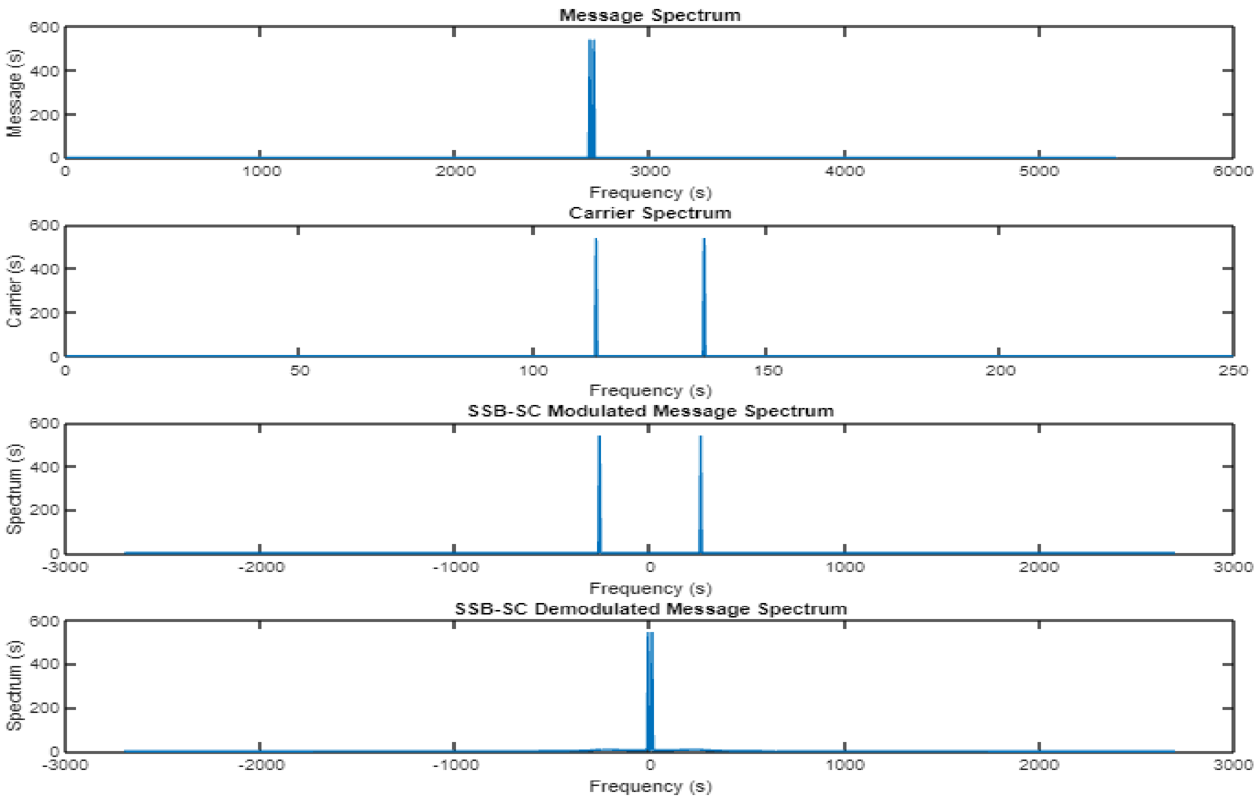

Message Spectrum
Message (s)
0.9950248756218
0.99502487562184
0.99502487562188
0.99502487562192
0.99502487562196
0.995024875622
Frequency (s)
Carrier Spectrum
Carrier (s)
Frequency (s)
×10^4
Modulated Message Spectrum at kp=1
Signal (s)
Frequency (s)
×10^4

Q.3. Illustrate the SSB-SC modulation and demodulation using a sinusoidal message signal in MATLAB. Assume that the upper side band is desired and coherent demodulation is used. Sketch the spectrum of the modulated and the recovered signals.

```
% Parameters for Q-3

Am=1;                                % Amplitude of Message Signal
Ac=1;                                % Amplitude of Carrier Signal
N = 1080;                            % Number of Samples
fm = 10;                             % Frequency of Message Signal
fc = 250;                            % Frequency of Carrier Signal
fs = 2*(fc + 2*fm)*10;
f= -fs/2:fs/N:fs/2-fs/N;
t=0:1/fs:(2/fm-1/fs);

% Message, Carrier and Modulated, Demodulated Signals with their Spectrums

m = Am*cos(2*pi*fm*t);               % Message Signal
ms = fft(m);                         % Message Spectrum
ms = fftshift(abs(ms));
fms = (0:length(ms)-1)*fs/length(ms);
c = Ac*cos(2*pi*fc*t);               % Carrier Signal
cs = fft(c);                         % Carrier Spectrum
cs = fftshift(abs(cs));
fcs = (0:length(cs)-1)*fc/length(cs);
s_usb = ssbmod(m,fc,fs,0,"upper");   % SSB-SC Modulated Signal
y_hat = ssbdemod(s_usb,fc,fs,0);     % SSB-SC Demodulated Signal
ft_usb = fft(s_usb);                 % SSB-SC Modulated Spectrum
ft_usb = fftshift(abs(ft_usb));
ft_dem = fft(y_hat);                 % SSB-SC Demodulated Spectrum
ft_dem = fftshift(abs(ft_dem));
figure();
subplot(4,1,1)
plot(t,m);
title("Message Signal")
xlabel("Time (t)")
ylabel("Message (t)")
subplot(4,1,2)
plot(t,c);
title("Carrier Signal")
xlabel("Time (t)")
ylabel("Carrier (t)")
subplot(4,1,3)
plot(t,s_usb);
title("SSB-SC Modulated Message Signal")
xlabel("Time (t)")
ylabel("Signal (t)")
subplot(4,1,4)
plot(t,y_hat);
```

```
title("SSB-SC Demodulated Message Signal")
xlabel("Time (t)")
ylabel("Signal (t)")
figure();
subplot(4,1,1)
plot(fms,ms);
title("Message Spectrum")
xlabel("Frequency (s)")
ylabel("Message (s)")
subplot(4,1,2)
plot(fcs,cs);
title("Carrier Spectrum")
xlabel("Frequency (s)")
ylabel("Carrier (s)")
subplot(4,1,3)
plot(f,ft_usb);
title("SSB-SC Modulated Message Spectrum")
xlabel("Frequency (s)")
ylabel("Spectrum (s)")
subplot(4,1,4)
plot(f,ft_dem);
title("SSB-SC Demodulated Message Spectrum")
xlabel("Frequency (s)")
ylabel("Spectrum (s)")
```



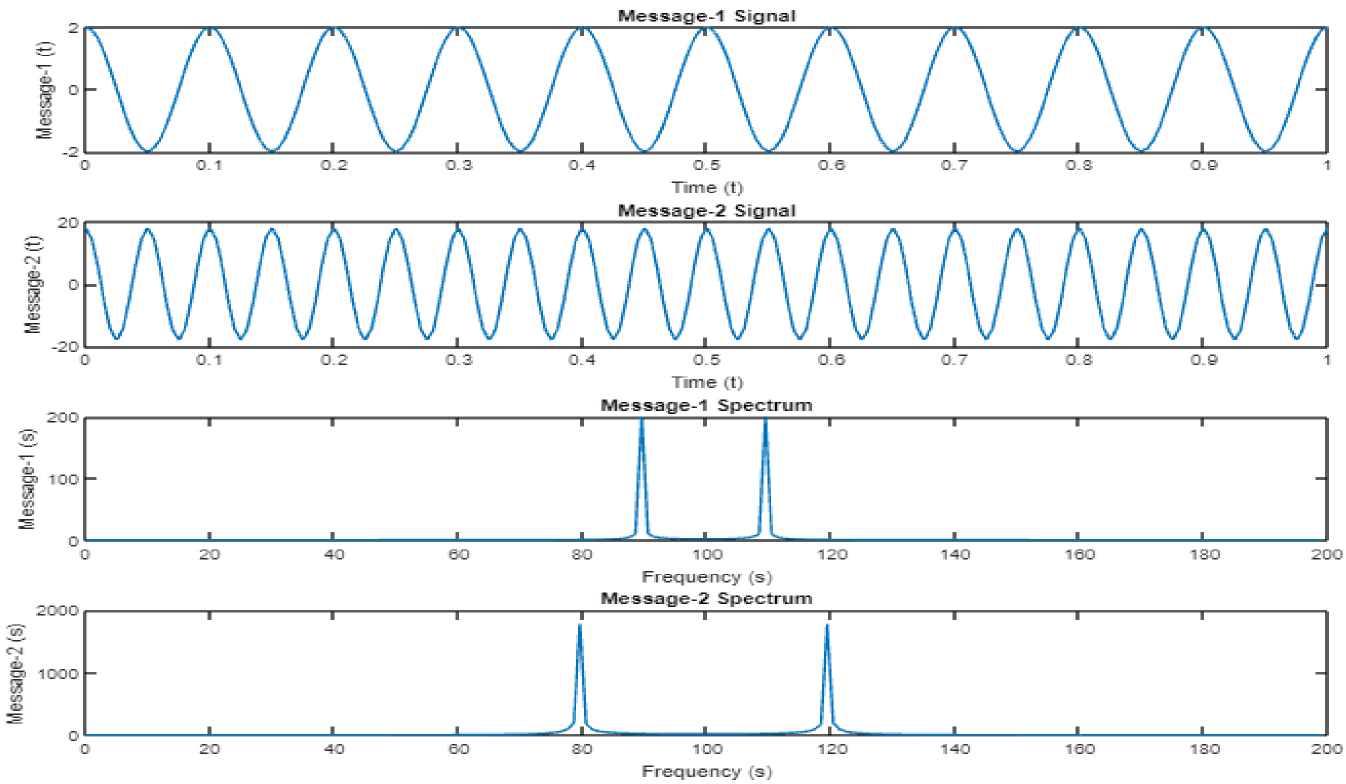


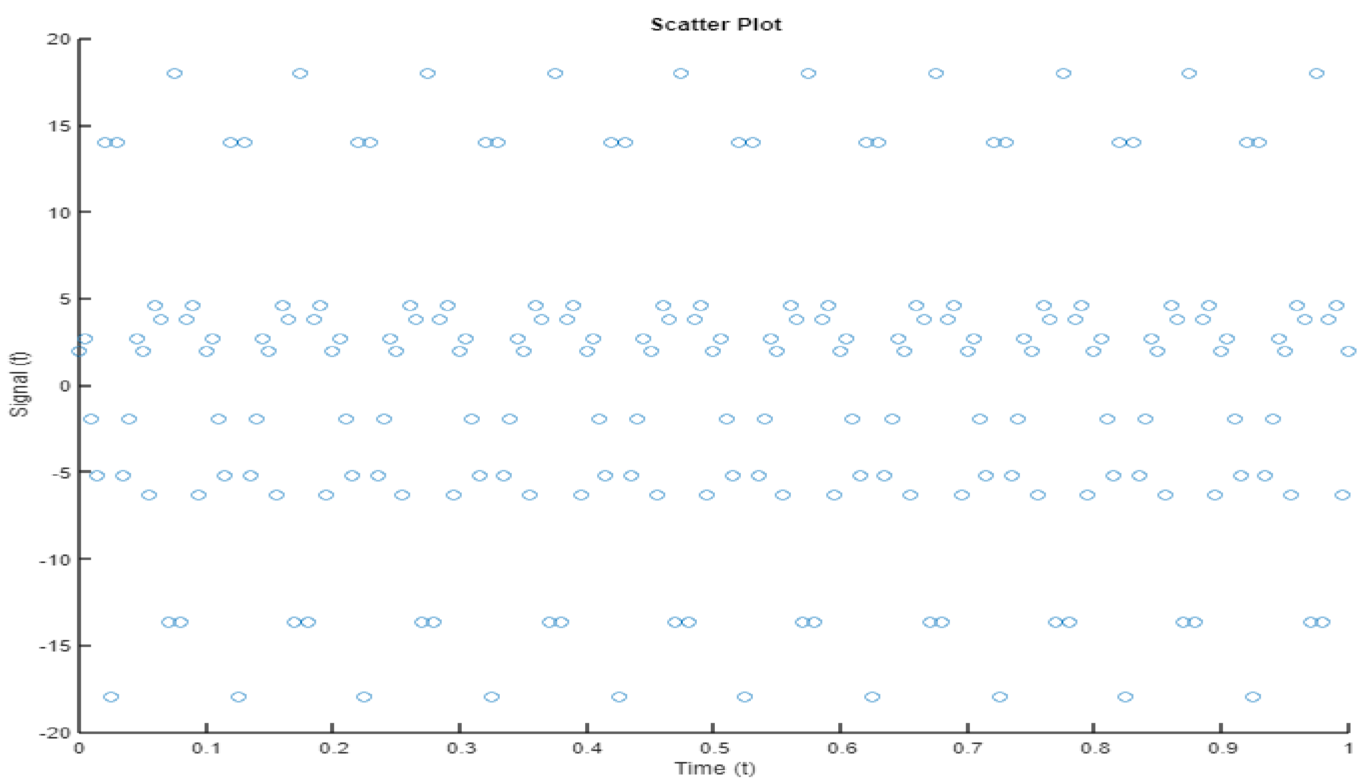
Q.4. Consider two sinusoidal message signals that have different frequencies. These are modulated using QAM. Sketch the time domain waveform and the spectrum of the modulated signal. Suppose there is a phase offset of θ in the local carrier used for demodulation with respect to the carrier on the transmitter side. Show the effect of different values of θ on the recovered signals.

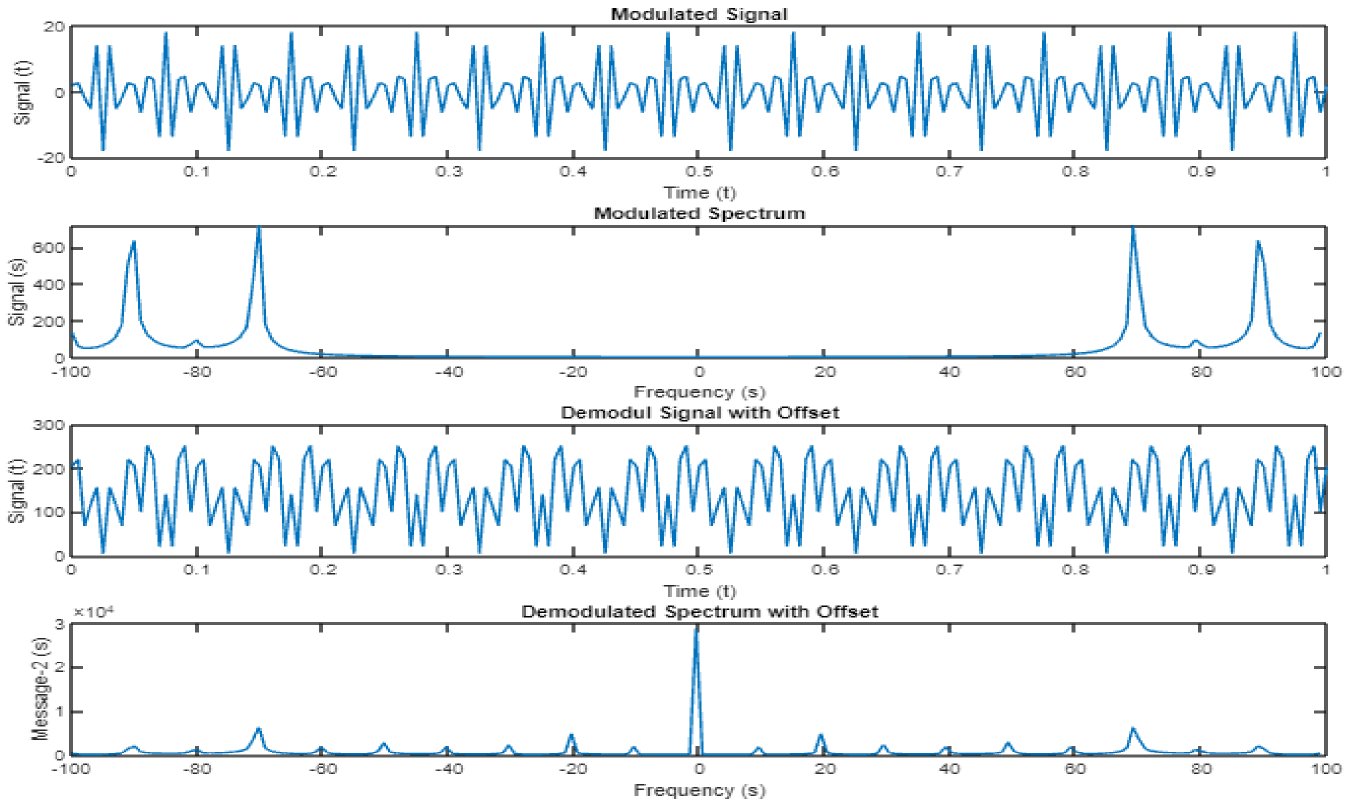
```
% Parameters for Q-4
fs = 200;
t = 0:1/fs:1;

% Message-1,2 and their Modulated, Demodulated Signals, Spectrums

m1 = 2*cos(2*pi*10*t);                      % Message-1 Signal
m1s = fft(m1);                              % Message-1 Spectrum
m1s = fftshift(abs(m1s));
fm1s = (0:length(m1s)-1)*fs/length(m1s);
m2 = 18*cos(2*pi*20*t);                     % Message-2 Signal
m2s = fft(m2);                              % Message-2 Spectrum
m2s = fftshift(abs(m2s));
fm2s = (0:length(m2s)-1)*fs/length(m2s);
figure();
subplot(4,1,1);
plot(t,m1);
title("Message-1 Signal")
xlabel("Time (t)")
```

```
ylabel("Message-1 (t)")
subplot(4,1,2);
plot(t,m2);
title("Message-2 Signal")
xlabel("Time (t)")
ylabel("Message-2 (t)")
subplot(4,1,3);
plot(fm1s,m1s);
title("Message-1 Spectrum")
xlabel("Frequency (s)")
ylabel("Message-1 (s)")
subplot(4,1,4);
plot(fm2s,m2s);
title("Message-2 Spectrum")
xlabel("Frequency (s)")
ylabel("Message-2 (s)")
y = modulate(m1, 90, fs, 'qam', m2);        % Modulated Signal
fftSignal = fft(y);                         % Modulated Spectrum
fftSignal = fftshift(abs(fftSignal));
n = length(t);
f= -fs/2:fs/n:fs/2-fs/n;
z = qamdemod(y.*exp(-1i*pi/32),256);        % Demodulated Signal
fft_demodulated_Signal = fft(z);            % Demodulated Spectrum
fft_demodulated_Signal = fftshift(abs(fft_demodulated_Signal));
figure();
scatter(t, y);
title("Scatter Plot")
xlabel("Time (t)")
ylabel("Signal (t)")
figure();
subplot(4,1,1);
plot(t, y);
title("Modulated Signal")
xlabel("Time (t)")
ylabel("Signal (t)")
subplot(4,1,2);
plot(f, fftSignal);
title("Modulated Spectrum")
xlabel("Frequency (s)")
ylabel("Signal (s)")
subplot(4,1,3);
plot(t, z);
title("Demodul Signal with Offset")
xlabel("Time (t)")
ylabel("Signal (t)")
subplot(4,1,4);
plot(f, fft_demodulated_Signal);
title("Demodulated Spectrum with Offset")
xlabel("Frequency (s)")
```

```
ylabel("Message-2 (s)")
```







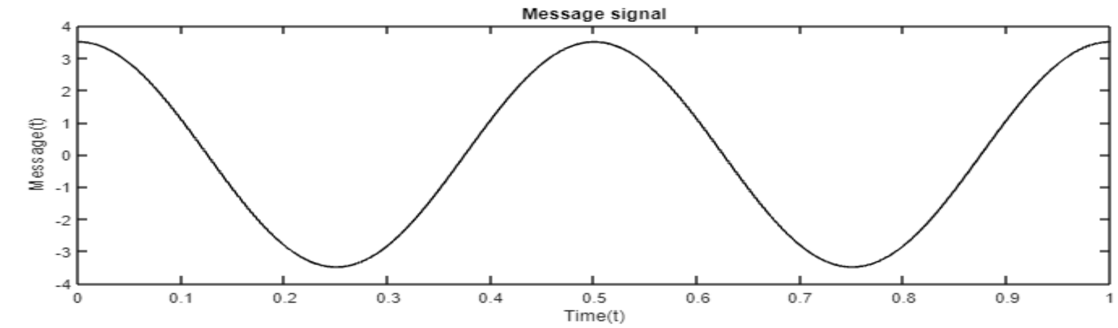
Q.5. Consider a sinusoidal message signal. The number of samples in a period needs to be taken as an input from the user. Plot the sampled signal. Now perform uniform quantization by taking the number of bits to represent each sample as an input from the user. Encode the signal and use Polar NRZ to map the encoded sequence. Plot the recovered signal at the receiver side. Repeat the exercise by increasing the number of bits to represent each sample and comment on the results.

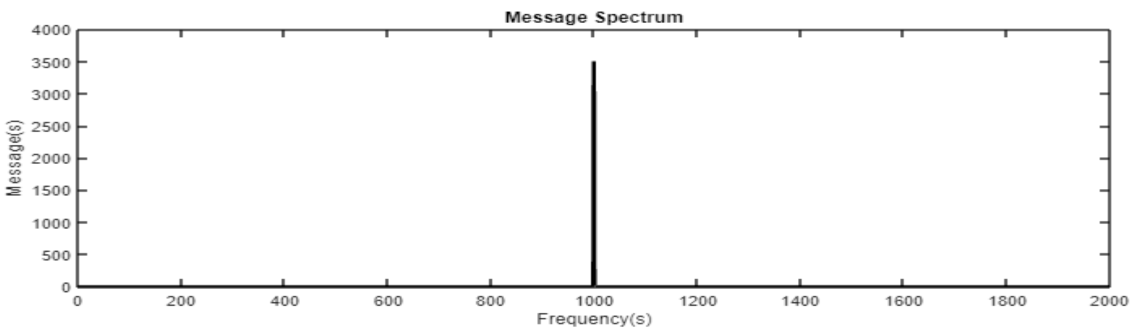
```
% Parameters and Inputs for Q-5

fm = input("Enter the Frequency of Message Signal ==> ");
Am = input("Enter the Amplitude of Message Signal ==> ");
bd = input("Enter the Number of bits ==> ");
fs = 1000*fm;
t = 0:1/fs:1;
m = Am*cos(2*pi*fm*t);                        % Message Signal
a = round(min(m));                            % Lowest Message Value
b = round(max(m));                            % Largest Message Value
delta = (b-a)/2.^bd;                          % Quant Size
v_max = b - delta/2;                          % Largest Quant Value
v_min = a + delta/2;                          % Lowest Quant Value
N = length(m);

% Signals and Spectrums Plotting

ms = fft(m);                                  % Message Spectrum
```

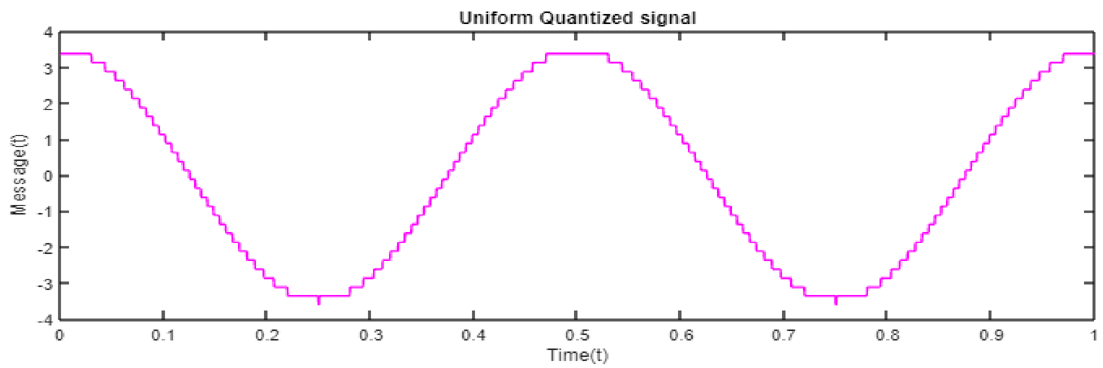
```
ms = fftshift(abs(ms));
fms = (0:length(ms)-1)*fs/length(ms);
figure()
subplot(2,1,1)
plot(t,m,'k-');
title('Message signal');
xlabel("Time(t)");
ylabel("Message(t)");
subplot(2,1,2)
plot(fms,ms,'k-');
title('Message Spectrum');
xlabel("Frequency(s)");
ylabel("Message(s)");

% Quantizer and Encoder

enc = [];
for i= v_min:delta:v_max
   for j=1:N
       if ((i-delta/2) < m(j)) && (m(j) <= (i+delta/2))
           idx = round((m(j) - v_min)/delta);
           e = de2bi(idx,bd,'left-msb');
           enc = [enc e];
           Xq(j) = i;
       end
   end
end

% Quantized Signal and Spectrum Plotting

xs = fft(Xq);                                    % Quantized Spectrum
xs = fftshift(abs(xs));
figure();
subplot(2,1,1);
plot(t,Xq, 'm-')
title('Uniform Quantized signal')
xlabel("Time(t)");
ylabel("Message(t)");
subplot(2,1,2);
plot(fms,xs, 'm-')
title('Uniform Quantized Spectrum')
xlabel("Frequency(s)");
ylabel("Message(s)");

% Quantized Error Plot

figure()
plot(m,m-Xq,'r-')
title('Quantization error')
```

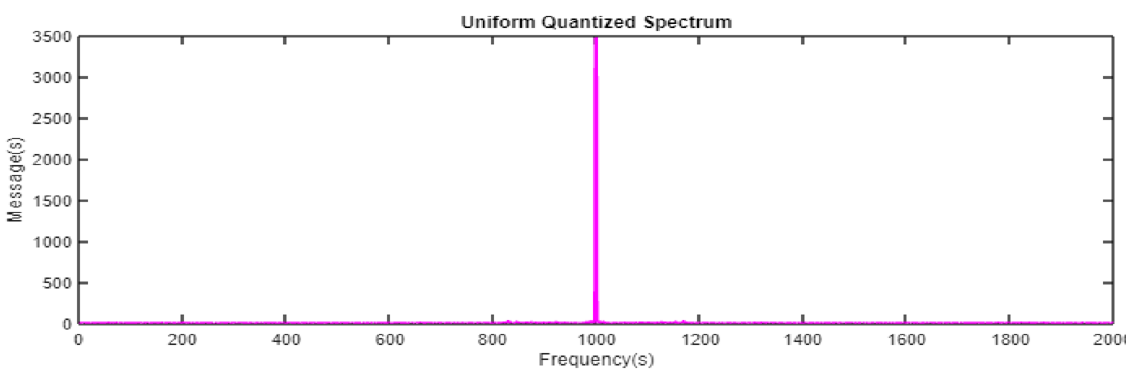
```
xlabel("Message (V)");
ylabel("Message Error (V)");

% Polar NRZ : 1 --> +1, 0 --> -1

tb = 0:.1:100;
fb = 1./tb;
for m=1:length(enc)
   if enc(m) == 1
       enc_mapped(m) =1;
   else
       enc_mapped(m) = -1;
   end
end
i=1;
for j=1:length(tb)
   if tb(j) <= i
       p_nrz(j) = enc_mapped(i);
   else
       i = i + 1;
   end
end

% Signal and Spectrum Plotting, after Polar NRZ Scheme

ps = fft(p_nrz);                                % Polarized NRZ Spectrum
ps = fftshift(abs(ps));
fps = (0:length(ps)-1).*fb/length(ps);
figure()
subplot(2,1,1);
plot(tb,p_nrz)
title('Polar No-Return to Zero Scheme Signal plot')
xlabel("Clock Time (t)");
ylabel("Clock Message (V)");
subplot(2,1,2);
plot(fps,ps)
title('Polar No-Return to Zero Scheme Spectrum plot')
xlabel("Clock Frequency (s)");
ylabel("Clock Message (s)");
```

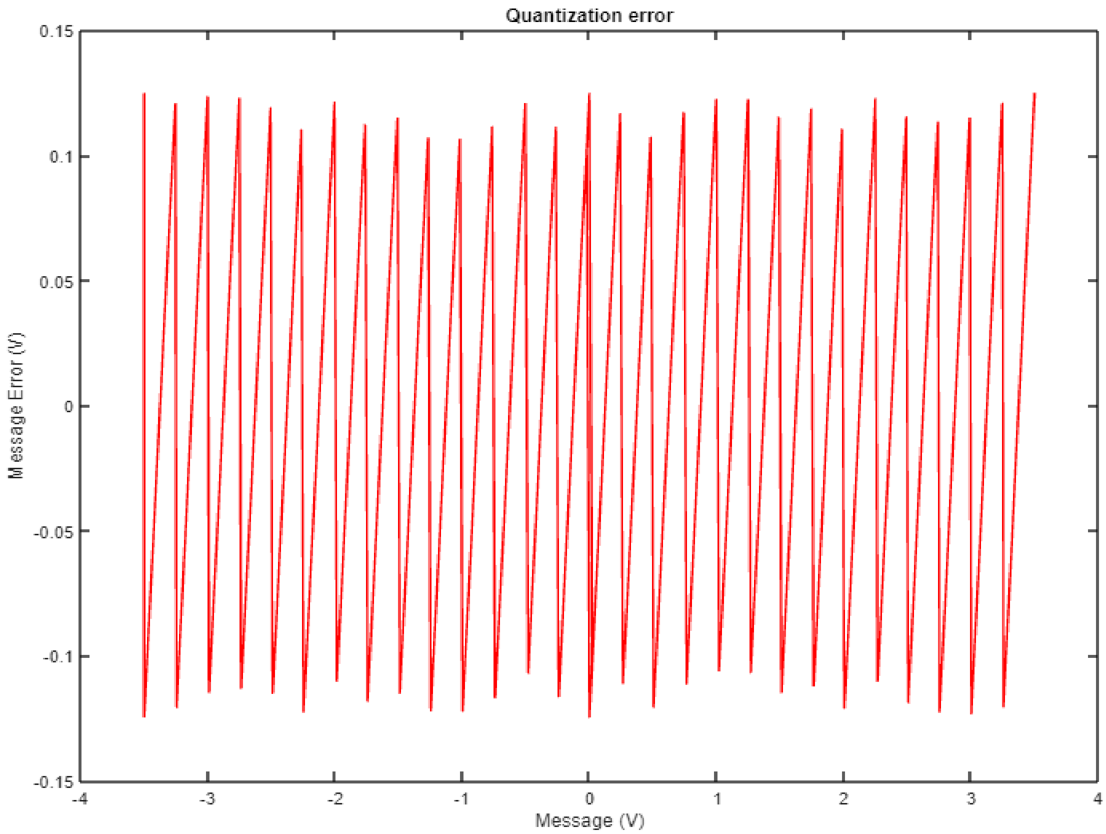
```
Enter the Frequency of Message Signal ==>
2
Enter the Amplitude of Message Signal ==>
3.5
Enter the Number of bits ==>
10
```

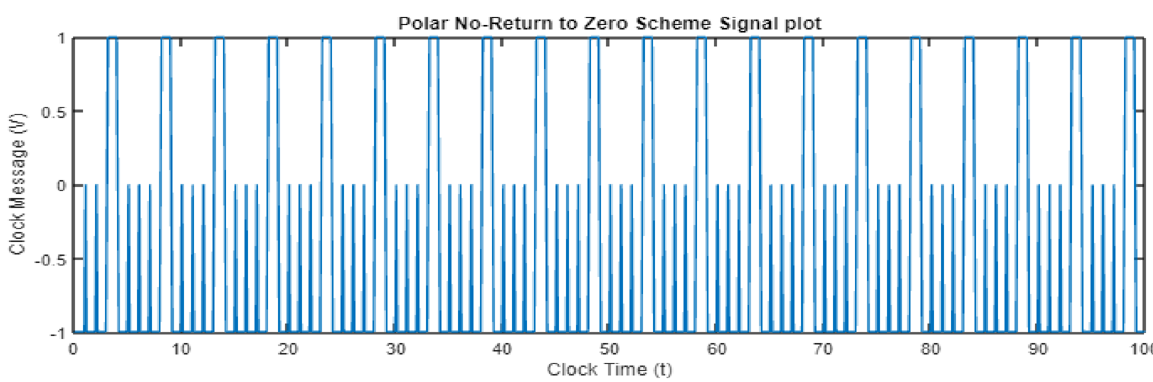

Message signal
Message(t)
Time(t)

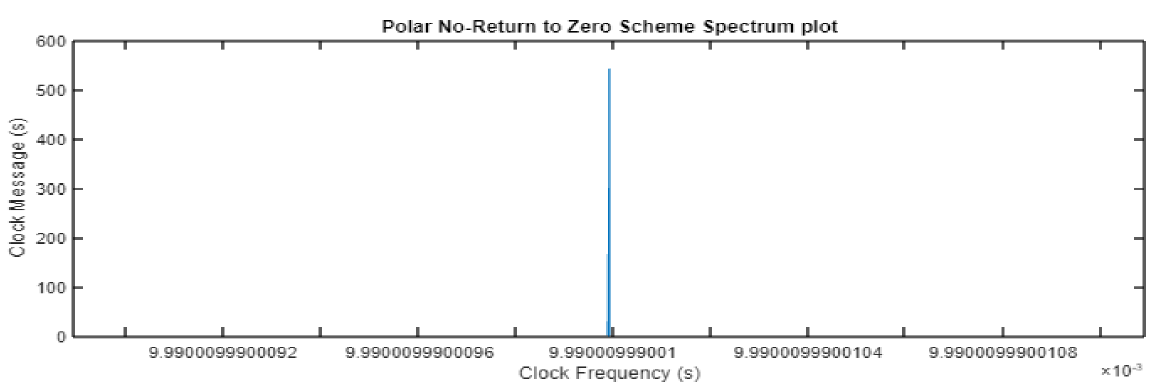

Message Spectrum
Message(s)
Frequency(s)


Uniform Quantized signal
Message(t)
Time(t)


Uniform Quantized Spectrum
Message(s)
Frequency(s)


Quantization error
Message Error (V)
Message (V)
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-4
-3
-2
-1
0
1
2
3
4


Polar No-Return to Zero Scheme Signal plot
Clock Message (V)
Clock Time (t)
1
0.5
0
-0.5
-1
0
10
20
30
40
50
60
70
80
90
100


Polar No-Return to Zero Scheme Spectrum plot
Clock Message (s)
Clock Frequency (s)
600
500
400
300
200
100
0
9.9900099900092
9.9900099900096
9.99000999001
9.9900099900104
9.9900099900108
×10^-3