Index for Wireless Communication

Problem No	Problem Name
01	To write a MATLAB program to calculate the median path loss for Hata model for outdoor propagation.
02	To write a MATLAB program to calculate the median path loss for Okumura model for outdoor propagation.
03	To write a MATLAB program to calculate the link budget for satellite communication.
04	To determine the free space loss and the power received using MATLAB program.
05	Write a program to Pulse Code Modulation Techniques By Using MATLAB
06	Write a MATLAB program to simulate Delta Modulation.
07	Write a MATLAB program to perform ASK, FSK &PSK using MATLAB and to obtain its output waveforms.
08	Write a MATLAB program to plot the constellation diagram of digital modulation system BPSK,QPSK&QAM using MATLAB.
09	Write a MATLAB program for OFDM spectrum.
10	Simulation of OFDM Transceiver Using OFDM.
11	Write a MATLAB program to simulate Least Mean Square (LMS) algorithm to adaptively adjust the coefficients of an FIR filter.
12	Write a MATLAB program to simulate the Zero Forcing Equalizer using MATLAB.

13	Write a MATLAB code for Frequency Division Multiplexing.
14	Write a MATLAB code for Time Division Multiplexing.
15	Write a MATLAB program to Simulation of 2 × 2, 4 × 4, 8 × 8 MIMO.
16	Cyclic Code Generation Using MATLAB
17	Implementation Of Linear Block Codes Using MATLAB
18	To obtain a Direct Sequence Spread Spectrum (DSSS) for a given input signal by simulating in MATLAB.
19	To perform the simulation of convolutional coding scheme and hamming code scheme using MATLAB.
20	To perform the simulation of DPSK modulation and demodulation using MATLAB.

Problem 01: To write a MATLAB program to calculate the median path loss for Hata model for outdoor propagation.

code:

```
clc;
clear all
```

```
close all;
f=input('enter the frequency of transmisson in mhz:');
Hb=input('enter the height of base station Antenna in meter:');
Hm=input('enter the height of mobile station Antenna in
meter:');
d=input('enter the distance between the base and mobile
stations:');
n=input('enter 0 for small or median city and 1 for large
city:');
if n==0
ch=0.8+(1.1*log10(f)-0.7)*Hm-1.56*log10(f);
else
if f>=150 && f<=200
ch=8.29*(log10(1.54*Hm))^.2-1.1;
else
if f>200 && f<=1500
ch=3.82*(log10(11.75*Hm))^.2-4.97;
end;
end;
end;
Lu=69.55+26.26*log10(f)-13.82*log10(Hb)-ch+(44.9-6.55*log10(Hb))
*log10(d);
disp(sprintf('%s %f %s','Path loss in Urban Areas=',Lu,'db'));
display(ch);
```

Input:

enter the frequency of transmisson in mhz:200

enter the height of base station Antenna in meter:50

enter the height of mobile station Antenna in meter:5

enter the distance between the base and mobile stations:5

enter 0 for small or median city and 1 for large city:0

Output:

Path loss in Urban Areas= 123.734662 db

Problem 02. To write a MATLAB program to calculate the median path loss for Okumura model for outdoor propagation.

Code

```
clc;
clear all;
close all;
Hte=30:1:1000; % Base Station Antenna Height
Hre=input('Enter the receiver antenna height 3m<hre<10m : ');
d =input('Enter distance from base station 1Km<d<100Km : ');
f=input('Enter the frequency 150Mhz<f<1920Mhz : ');
c=3*10^8;
lamda=(c)/(f*10^6);
Lf = 10*log((lamda^2)/((4*pi)^2)*d^2);
Amu = 35;
Garea = 9;
Ghte = 20*log(Hte/200);
if(Hre>3)
Ghre = 20*log(Hre/3);
else
Ghre = 10*log(Hre/3);
end
%   Propagation Path Loss
L50 = Lf+Amu-Ghte-Ghre-Garea;
display('Propagation pathloss is : ');
disp(L50);
plot(Hte,L50);
title('Okumura Model Analysis');
xlabel('Transmitter antenna Height (Km)');
ylabel('Propagation Path loss(dB) at 50 Km');
grid on;
```

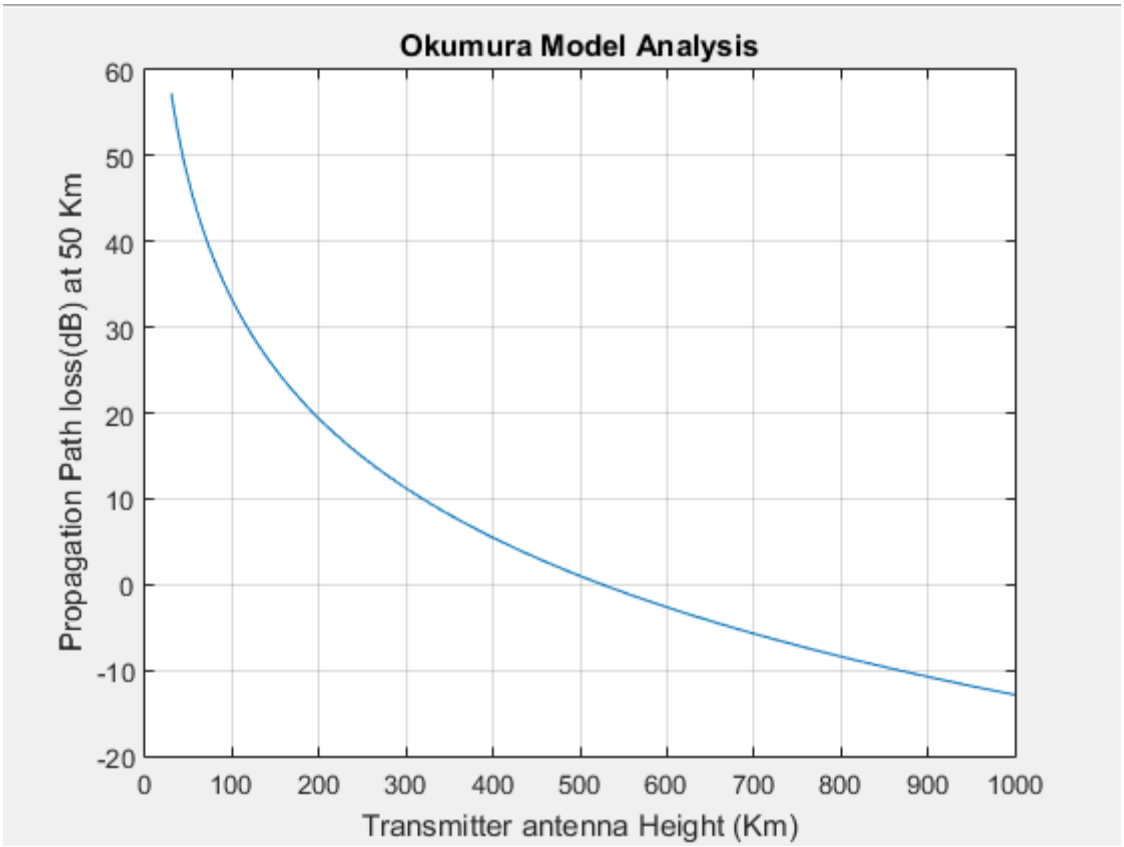
Input:

Enter the receiver antenna height 3m<hre<10m : 5

Enter distance from base station 1Km<d<100Km : 10

Enter the frequency 150Mhz<f<1920Mhz : 200

Output:



Problem 03: To write a MATLAB program to calculate the link budget for satellite communication.

Code:

```
clc;
close all;
clear all;
pt=input('enter the input power in watts:');
Pt=10*log10(pt) %calculating transmitted power in db
gt=input('enter the transmitting antenna gain in db:');
gs=input('enter the recieving antenna gain in db:');
EIRP=Pt+gt %calculating EIRP
d=input('enter the distance in km:');
f=input('enter the frequency in mhz:');
fsl=32.4+20*log10(d)+20*log10(f) %calculating path loss
rfl=input('enter the reciever feeder loss in db:');
aa=input('enter the atmospheric absorption in db:');
aml=input('enter the antenna misalignment loss in db:');
pl=input('enter the polarization loss in db:');
```

```
losses=fsl+rfl+aa+aml+pl; %calculating total losses
disp(sprintf('%s %f %s','total loss',losses,'db'));
P=EIRP+gs-losses; %calculating power recieved
disp(sprintf('%s %f %s','Total recieved power =',P,'db'));
```

Input:

enter the input power in watts:25
enter the transmitting antenna gain in db:1
enter the recieving antenna gain in db:2
enter the distance in km:1000
enter the frequency in mhz:900
enter the reciever feeder loss in db:.1
enter the atmospheric absorption in db:2
enter the antenna misalignment loss in db:4.5
enter the polarization loss in db:5

Output:

total loss 163.084850 db

Total recieved power = -146.105450 db

Problem 04: To determine the free space loss and the power received using MATLAB program.

Code:

```
clc;
close all;
clear all
f=input('enter the frequency in Mhz: ');
L=300/f; %calculating wavelength
disp('thus the wavelength is: ');
 %displaying wavelength
d=input('enter the distance in km: ');
%d=1:1:10000;
Gt=input('enter the transmitting antenna gain in db: ');
Gr=input('enter the receiving antenna gain in db: ');
Wt=input('enter the transmitted power in db: ');
```

```
ls=32.44+20*log10(d)+20*log10(f); %calculating path loss
disp(sprintf('%s %d %s','the path loss
is:',ls,'db'));%displaying path loss
%plot(d,ls);
Wr=Wt+Gt+Gr-ls; %calculating recieved power in db
disp(sprintf('%s %d %s','the recieved power is:',Wr,'db'));
wr=10^(Wr/10); %calculating recieved power in watts
disp(sprintf('%s %d %s','the recieved power is:',wr,'watts'));
%displaying recieved power in watts
```

Input:

enter the frequency in Mhz: 900

thus the wavelength is:

enter the distance in km: 5

enter the transmitting antenna gain in db: 1

enter the receiving antenna gain in db: 2

enter the transmitted power in db: 25

Output:

the path loss is: 1.055043e+02 db

the recieved power is: -7.750425e+01 db

the recieved power is: 1.776540e-08 watts

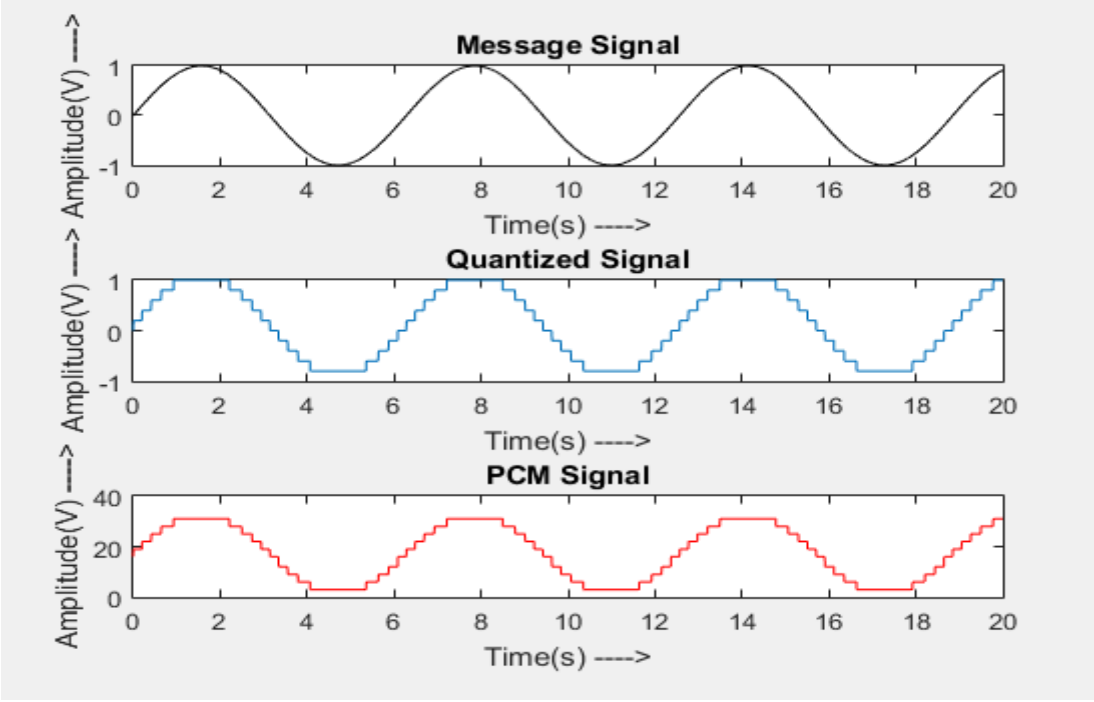
Problem 05. Write a program to Pulse Modulation Techniques By Using MATLAB

Code:

```
clc;
clear all;
t = 0:0.005:20;
% Representation of the Message Signal
x = sin(t);
subplot(3,1,1);
```

```
plot(t,x,'black');
title('Message Signal');
xlabel('Time(s) ---->')
ylabel('Amplitude(V) ---->')
% Representation of the Quantized Signal
partition = -1:0.2:2;
codebook = -1:0.2:2.2;
[index,quants] = quantiz(x,partition,codebook);
subplot(3,1,2);
plot(t,quants);
title('Quantized Signal');
xlabel('Time(s) ---->')
ylabel('Amplitude(V) ---->')
% Representation of the PCM Signal
y = uencode(quants,5);
subplot(3,1,3);
plot(t,y,'red');
title('PCM Signal');
xlabel('Time(s) ---->');
ylabel('Amplitude(V) ---->')
```

Output:

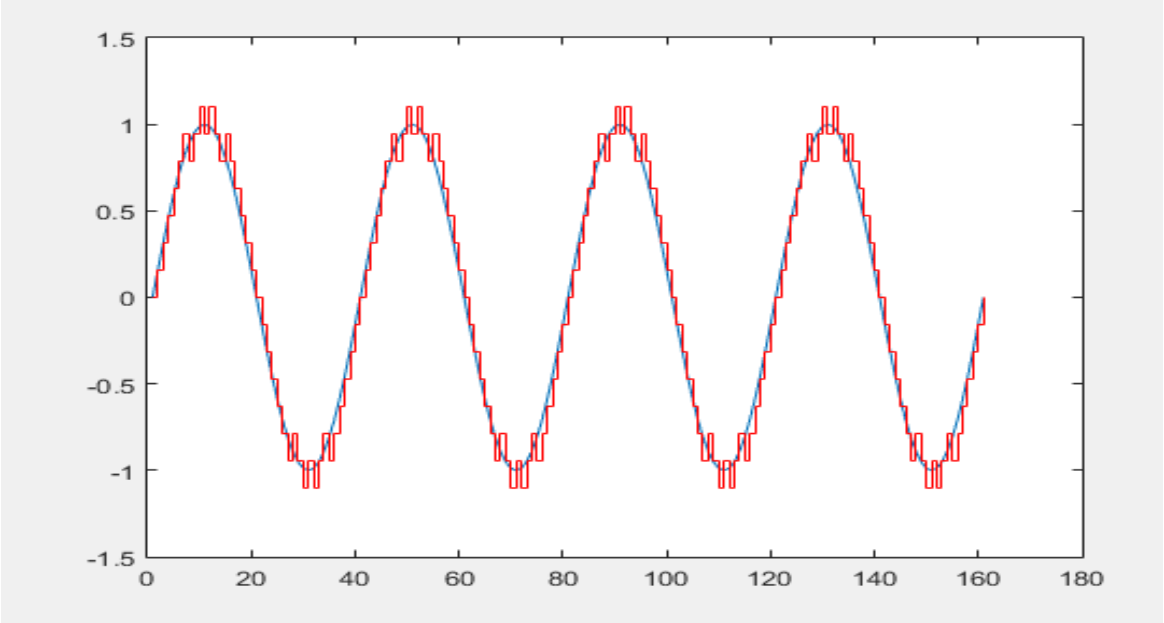


Problem 06: Write a MATLAB program to simulate Delta Modulation.

Code:

```
am=1;
fm=1;
fs=40*fm;%over sampling
t=0:1/fs:4;
x=am*sin(2*pi*fm*t);
plot(x)
hold on
d=(2*pi*am*fm)/fs;%step size
for n= 1: length(x)
if n==1
e(n)=x(n);
eq(n)=d*sign(e(n));
xq(n)=eq(n);
else
e(n)=x(n)-xq(n-1);
eq(n)=d*sign(e(n));
xq(n)=xq(n-1)+eq(n);
end
end
stairs(xq,'red');
%hleg=legend('original signal','stair case approximated signal')
hold off
```

Output:



Problem 07: Write a MATLAB program to perform ASK, FSK & PSK using MATLAB and to obtain its output waveforms.

ASK Modulation:

Code:

```
clc
close all
clear all
fc=input('Enter the freq of Sine Wave carrier:');
fp=input('Enter the freq of Periodic Binary pulse (Message):');
amp=input('Enter the amplitude (For Carrier & Binary Pulse
Message):');
t=0:0.001:1; % For setting the sampling interval
c=amp.*sin(2*pi*fc*t);
subplot(3,1,1) %For Plotting The Carrier wave
plot(t,c)
xlabel('Time')
ylabel('Amplitude')
title('Carrier Wave')
m=amp/2.*square(2*pi*fp*t)+(amp/2);
subplot(3,1,2) %For Plotting The Square Binary Pulse (Message)
plot(t,m)
xlabel('Time')
ylabel('Amplitude')
title('Binary Message Pulses')
w=c.*m; % The Shift Keyed Wave
subplot(3,1,3) %For Plotting The Amplitude Shift Keyed Wave
plot(t,w)
xlabel('Time')
ylabel('Amplitude')
title('Amplitide Shift Keyed Signal')
```

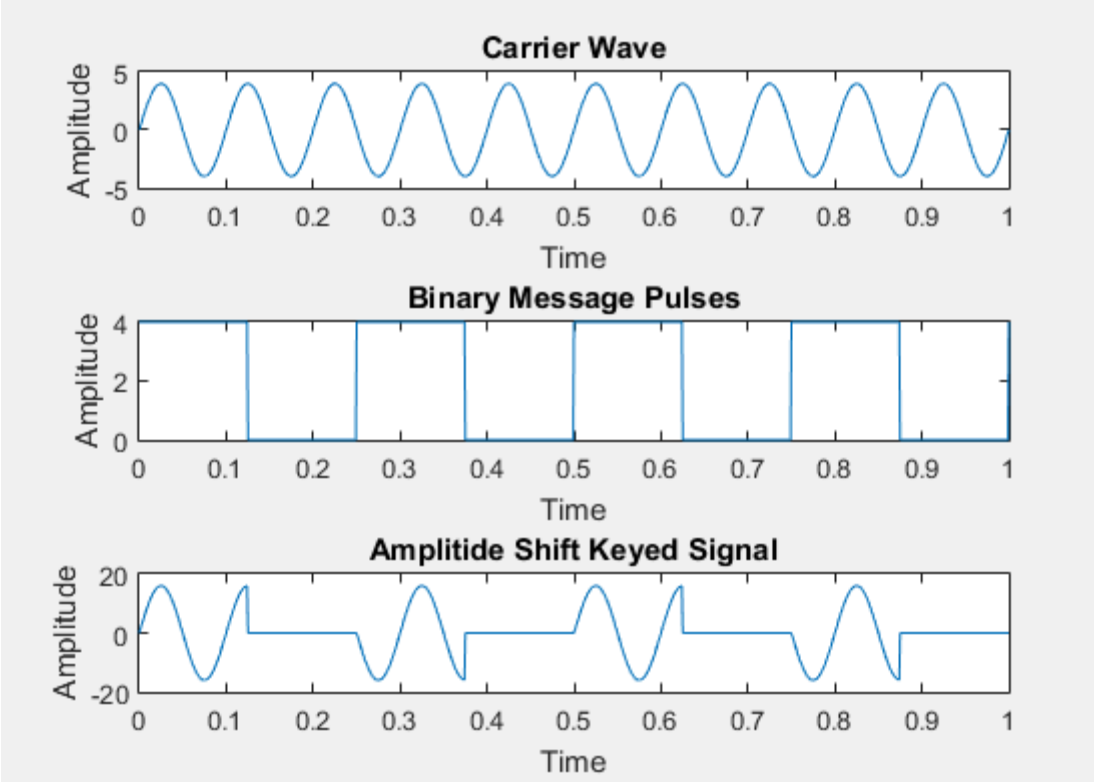
Input :

Enter the freq of Sine Wave carrier:10

Enter the freq of Periodic Binary pulse (Message):4

Enter the amplitude (For Carrier & Binary Pulse Message):4

Output:



FSK Modulation:

Code:

```
clc
close all
clear all
fc1=input('Enter the freq of 1st Sine Wave carrier:');
fc2=input('Enter the freq of 2nd Sine Wave carrier:');
fp=input('Enter the freq of Periodic Binary pulse (Message):');
amp=input('Enter the amplitude (For Both Carrier & Binary Pulse
Message):');
amp=amp/2;
t=0:0.001:1; % For setting the sampling interval
c1=amp.*sin(2*pi*fc1*t);% For Generating 1st Carrier Sine wave
c2=amp.*sin(2*pi*fc2*t);% For Generating 2nd Carrier Sine wave
subplot(4,1,1); %For Plotting The Carrier wave
plot(t,c1)
xlabel('Time')
ylabel('Amplitude')
title('Carrier 1 Wave')
subplot(4,1,2) %For Plotting The Carrier wave
```

```
plot(t,c2)
xlabel('Time')
ylabel('Amplitude')
title('Carrier 2 Wave')
m=amp.*square(2*pi*fp*t)+amp;%For Generating Square wave message
subplot(4,1,3) %For Plotting The Square Binary Pulse (Message)
plot(t,m,'red')
xlabel('Time')
ylabel('Amplitude')
title('Binary Message Pulses')
for i=0:1000 %here we are generating the modulated wave
    if m(i+1)==0
        mm(i+1)=c2(i+1);
        else
        mm(i+1)=c1(i+1);
    end
end
subplot(4,1,4) %For Plotting The Modulated wave
plot(t,mm)
xlabel('Time')
ylabel('Amplitude')
title('Modulated Wave')
```

Input:

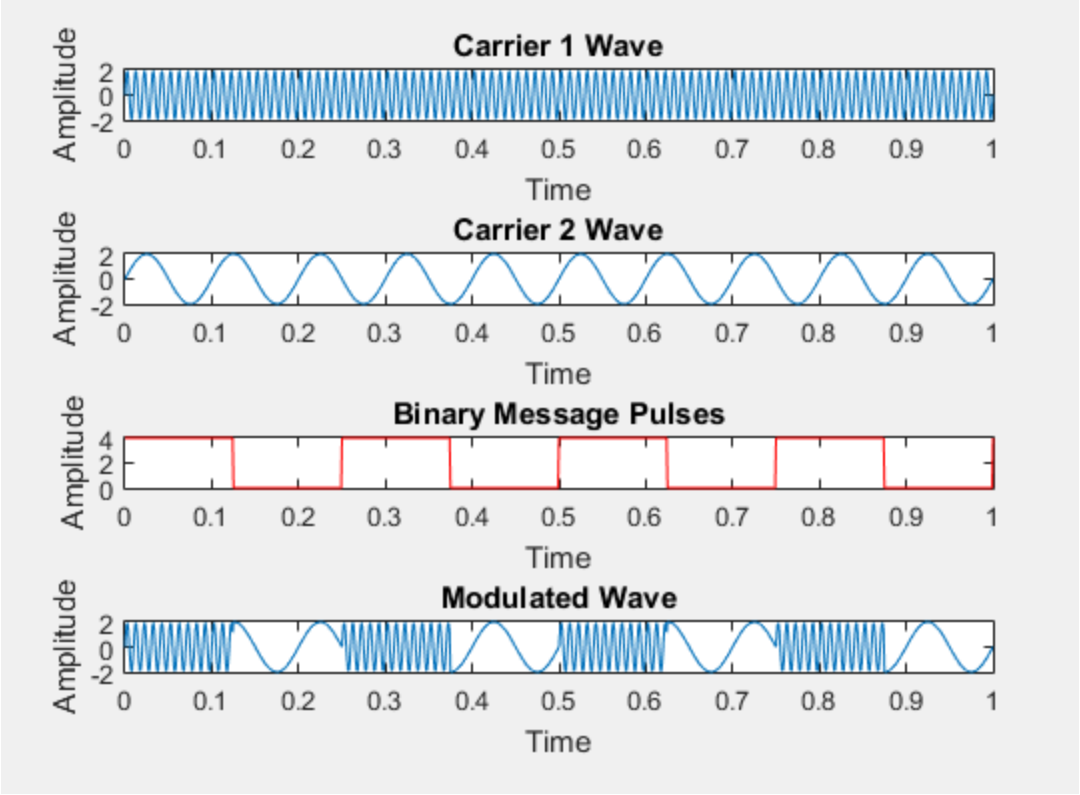
Enter the freq of 1st Sine Wave carrier:100

Enter the freq of 2nd Sine Wave carrier:10

Enter the freq of Periodic Binary pulse (Message):4

Enter the amplitude (For Both Carrier & Binary Pulse Message):4

Output:



PSK Modulation:

Code:

```
clc
close all
clear all
t=0:.001:1; % For setting the sampling interval
fc=input('Enter frequency of Carrier Sine wave: ');
fm=input('Enter Message frequency : ');
amp=input('Enter Carrier & Message Amplitude(Assuming Both
Equal):');
c=amp.*sin(2*pi*fc*t);% Generating Carrier Sine
subplot(3,1,1)
plot(t,c)
xlabel('Time')
ylabel('Amplitude')
title('Carrier')
m=square(2*pi*fm*t);% For Plotting Message signal
subplot(3,1,2)
plot(t,m)
xlabel('time')
```

```
ylabel('ampmplitude')
title('Message Signal')
x=c.*m;
subplot(3,1,3)
plot(t,x)
xlabel('t')
ylabel('y')
title('PSK')
```

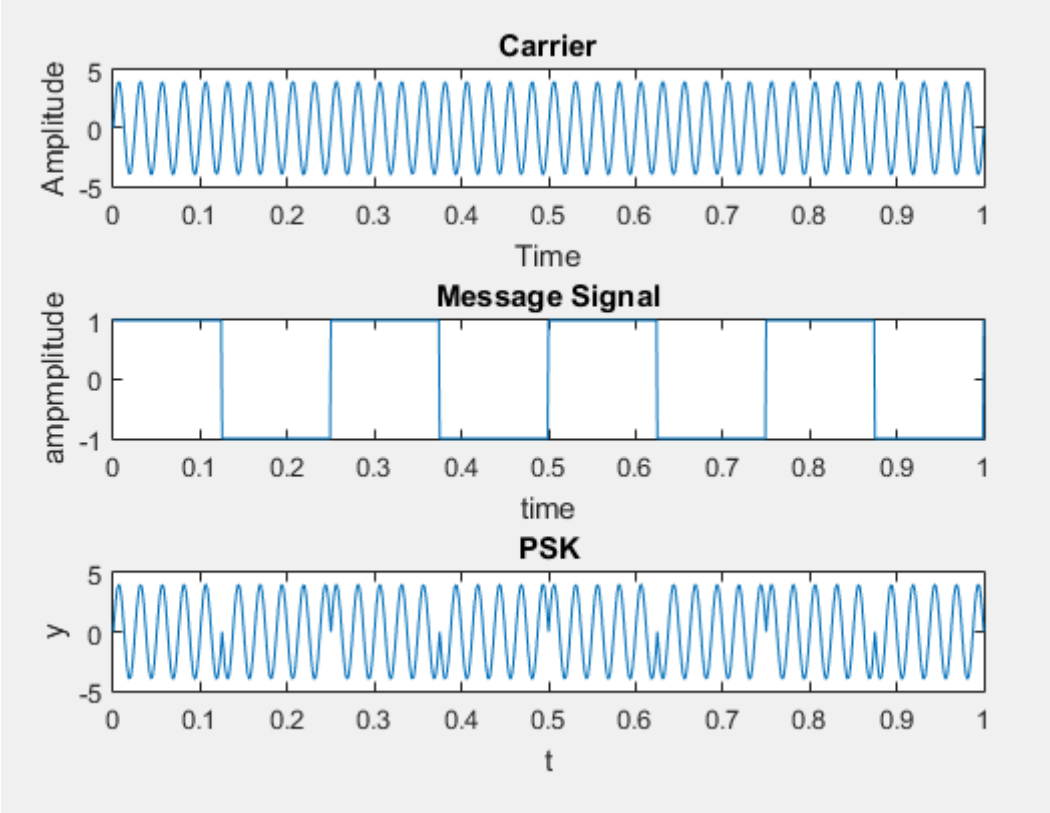
Input:

Enter frequency of Carrier Sine wave: 40

Enter Message frequency : 4

Enter Carrier & Message Amplitude(Assuming Both Equal):4

Output:



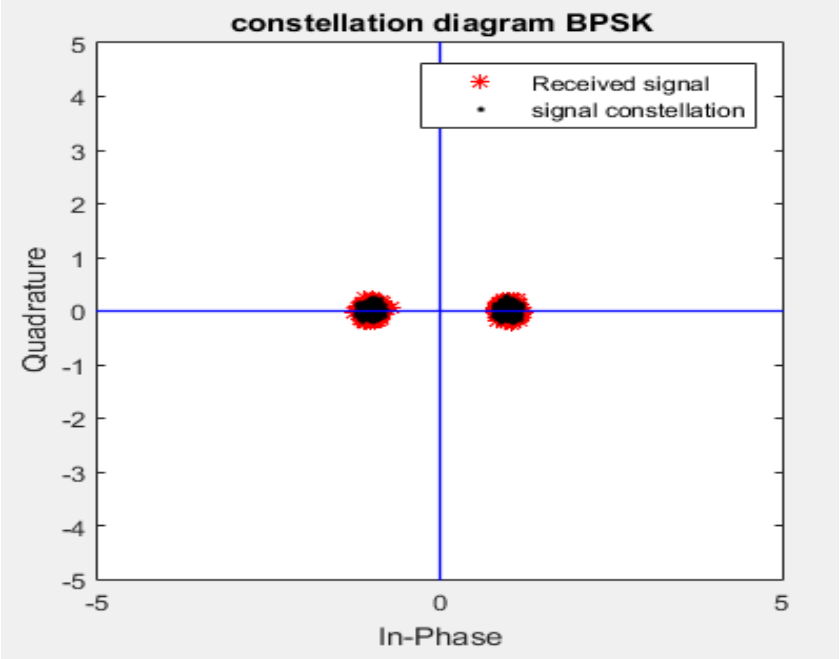
Problem 08: Write a MATLAB program to plot the constellation diagram of digital modulation system BPSK, QPSK & QAM using MATLAB.

For BPSK:

Code:

```
clc;
clear all;
close all;
M=2;
k=log2(M);
n=3*1e5;
nsamp=8;
X=randint(n,1);
xsym = bi2de(reshape(X,k,length(X)/k).','left-msb');
Y_psk= modulate(modem.pskmod(M),xsym);
Ytx_psk = Y_psk;
EbNo=30;
SNR=EbNo+10*log10(k)-10*log10(nsamp);
Ynoisy_psk = awgn(Ytx_psk,SNR,'measured');
Yrx_psk = Ynoisy_psk;
h1=scatterplot(Yrx_psk(1:nsamp*5e3),nsamp,0,'r*');
hold on;
scatterplot(Yrx_psk(1:5e3),1,0,'k.',h1);
xlim([-5 +5])
ylim([-5 +5])
line(xlim,[0 0],'color','b','linewidth',1);
line([0 0],ylim,'color','b','linewidth',1);
title('constellation diagram BPSK');
legend('Received signal' ,'signal constellation');
axis([-5 5 -5 5]);
hold off;
```

Output:

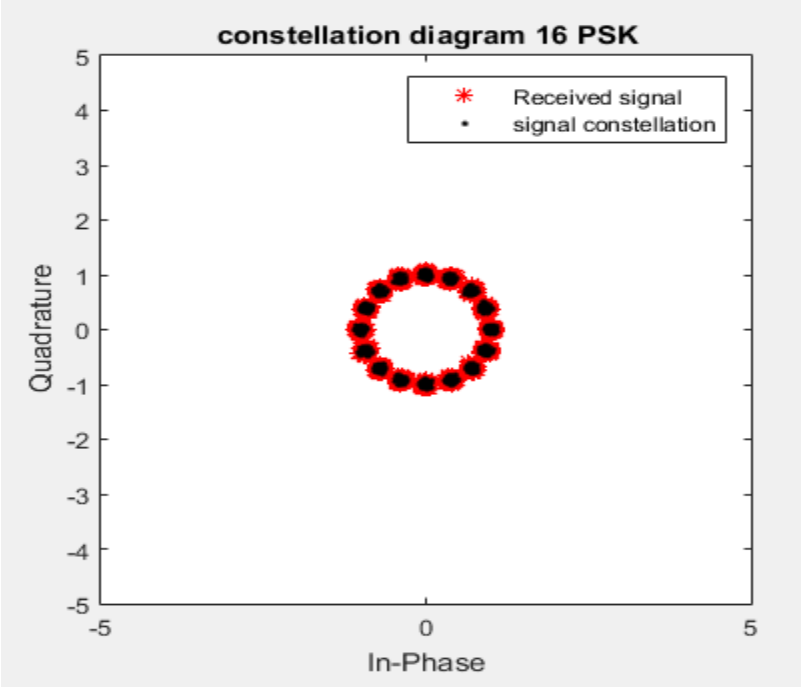


For QPSK

Code:

```
clc;
clear all;
close all;
M=16;
k=log2(M);
n=3*1e5;
nsamp=8;
X=randint(n,1);
xsym = bi2de(reshape(X,k,length(X)/k).','left-msb');
Y_qpsk= modulate(modem.pskmod(M),xsym);
Ytx_qpsk = Y_qpsk;
EbNo=30;
SNR=EbNo+10*log10(k)-10*log10(nsamp);
Ynoisy_qpsk = awgn(Ytx_qpsk,SNR,'measured');
Yrx_qpsk = Ynoisy_qpsk;
h2=scatterplot(Yrx_qpsk(1:nsamp*5e3),nsamp,0,'r*');
hold on;
scatterplot(Yrx_qpsk(1:5e3),1,0,'k.',h2);
title('constellation diagram 16 PSK');
legend('Received signal' ,'signal constellation');
axis([-5 5 -5 5]);
hold off;
```

Output:



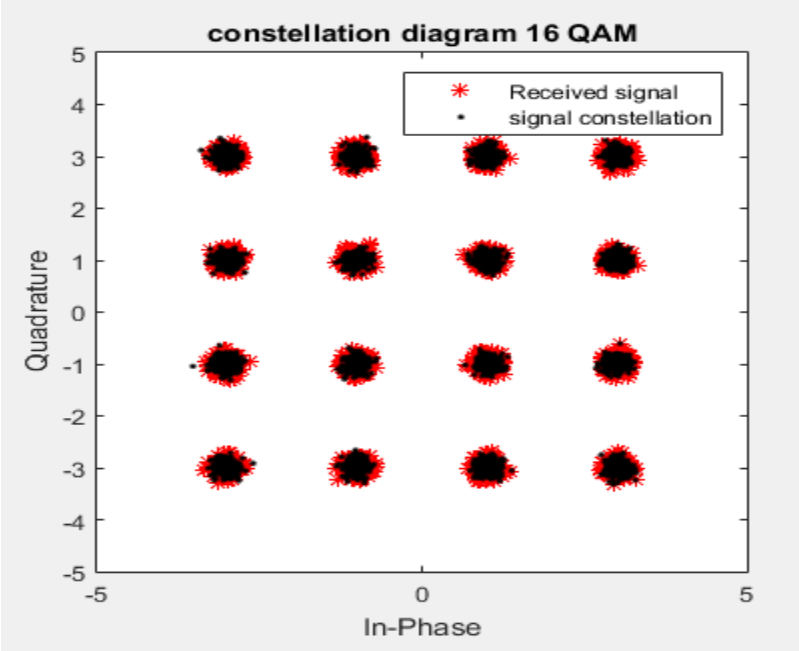
For QAM:

Code:

```
clc;
clear all;
close all;
M=16;
k=log2(M);
n=3*1e5;
nsamp=8;
X=randint(n,1);
xsym = bi2de(reshape(X,k,length(X)/k).','left-msb');
Y_qam= modulate(modem.qammod(M),xsym);
Ytx_qam = Y_qam;
EbNo=30;
SNR=EbNo+10*log10(k)-10*log10(nsamp);
Ynoisy_qam = awgn(Ytx_qam,SNR,'measured');

Yrx_qam = Ynoisy_qam;
h1=scatterplot(Yrx_qam(1:nsamp*5e3),nsamp,0,'r*');
hold on;
scatterplot(Yrx_qam(1:5e3),1,0,'k.',h1);
title('constellation diagram 16 QAM');
legend('Received signal' ,'signal constellation');
axis([-5 5 -5 5]);
hold off;
```

Output:



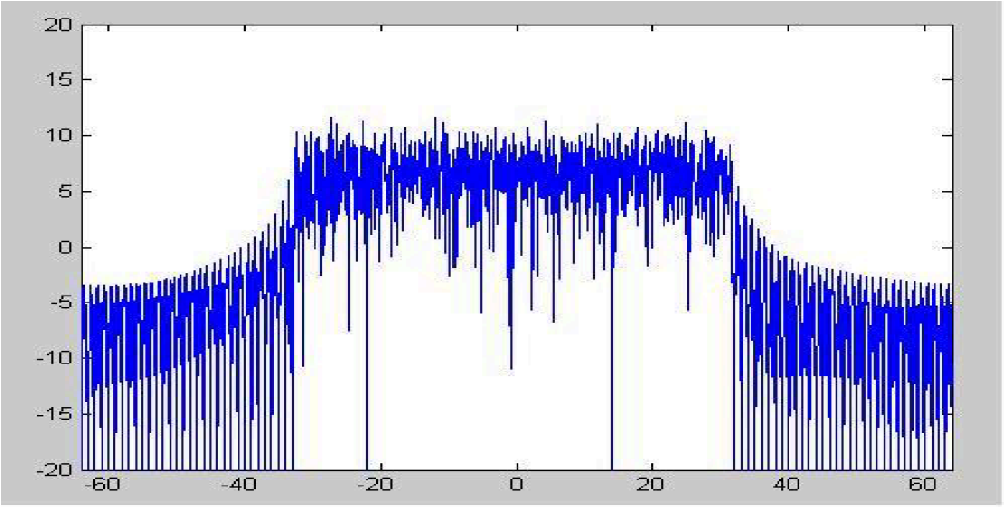
Problem 09: Write a MATLAB program for OFDM spectrum.

Code :

```
clear
nFFTSize = 64;
subcarrierIndex = [-26:-1 1:26];
nBit = 2500;
ip = rand(1,nBit) > 0.5; % generating 1's and 0's
nBitPerSymbol = 52;
nSymbol = ceil(nBit/nBitPerSymbol);
ipMod = 2*ip - 1;
ipMod = [ipMod zeros(1,nBitPerSymbol*nSymbol-nBit)];
ipMod = reshape(ipMod,nSymbol,nBitPerSymbol);
st = []; % empty vector
for ii = 1:nSymbol
inputiFFT = zeros(1,nFFTSize);
% assigning bits a1 to a52 to subcarriers [-26 to -1, 1 to 26]
inputiFFT(subcarrierIndex+nFFTSize/2+1) = ipMod(ii,:);
%  shift subcarriers at indices [-26 to -1] to fft input indices [38 to 63]
inputiFFT = fftshift(inputiFFT);
outputiFFT = ifft(inputiFFT,nFFTSize);
% adding cyclic prefix of 16 samples
outputiFFT_with_CP = [outputiFFT(49:64) outputiFFT];
st = [st outputiFFT_with_CP];
end
close all
fsMHz = 20;
```

```
[Pxx,W] = pwelch(st,[],[],4096,20);
plot([-2048:2047]*fsMHz/4096,10*log10(fftshift(Pxx)));
xlabel('frequency, MHz')
ylabel('power spectral density')
title('Transmit spectrum OFDM (based on 802.11a)');
```

Output:



Problem 10: Simulation of OFDM Transceiver Using OFDM.

Code:

```
close all
clear all
clc
t_data=randint(9600,1)';
x=1;
si=1; %for BER rows
%%
for d=1:100;
data=t_data(x:x+95);
x=x+96;
k=3;
n=6;
s1=size(data,2);  % Size of input matrix
j=s1/k;
constlen=7;
codegen = [171 133];    % Polynomial
trellis = poly2trellis(constlen, codegen);
codedata = convenc(data, trellis);
s2=size(codedata,2);
j=s2/4;
matrix=reshape(codedata,j,4);
```

```
intlvddata = matintrlv(matrix',2,2)'; % Interleave.
intlvddata=intlvddata';
dec=bi2de(intlvddata','left-msb');
M=16;
y = qammod(dec,M);
lendata=length(y);
pilt=3+3j;
nofpits=4;
k=1;
for i=(1:13:52)

    pilt_data1(i)=pilt;
    for j=(i+1:i+12);
        pilt_data1(j)=y(k);
        k=k+1;
    end
end
pilt_data1=pilt_data1';   % size of pilt_data =52
pilt_data(1:52)=pilt_data1(1:52);    % upsizing to 64
pilt_data(13:64)=pilt_data1(1:52);   % upsizing to 64
for i=1:52

    pilt_data(i+6)=pilt_data1(i);

end
ifft_sig=ifft(pilt_data',64);
cext_data=zeros(80,1);
cext_data(1:16)=ifft_sig(49:64);
for i=1:64

    cext_data(i+16)=ifft_sig(i);

end
o=1;
for snr=0:2:50
ofdm_sig=awgn(cext_data,snr,'measured');
for i=1:64

    rxed_sig(i)=ofdm_sig(i+16);

end
ff_sig=fft(rxed_sig,64);
for i=1:52

    synched_sig1(i)=ff_sig(i+6);

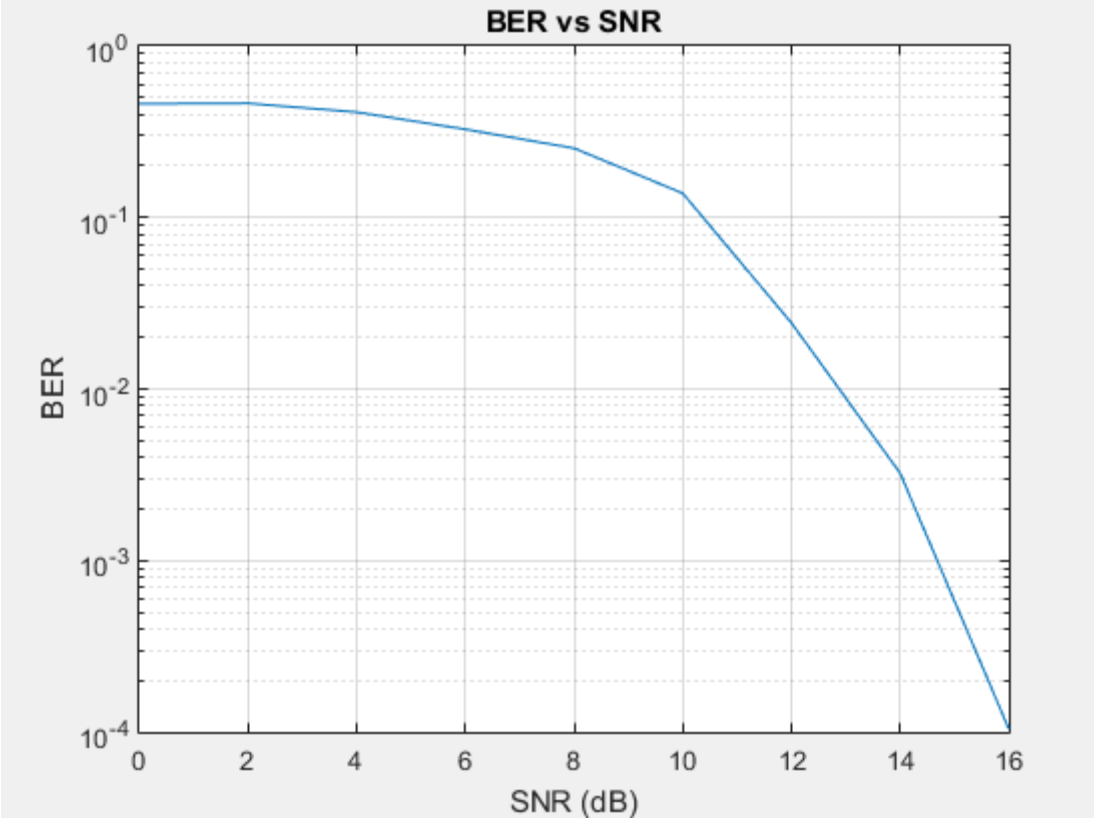
end
k=1;
for i=(1:13:52)

    for j=(i+1:i+12);
        synched_sig(k)=synched_sig1(j);
```

```
        k=k+1;
    end
end
dem_data= qamdemod(synched_sig,16);
bin=de2bi(dem_data','left-msb');
bin=bin';
deintlvddata = matdeintrlv(bin,2,2); % De-Interleave
deintlvddata=deintlvddata';
deintlvddata=deintlvddata(:)';
n=6;
k=3;
decodedata =vitdec(deintlvddata,trellis,5,'trunc','hard');  %
decoding datausing veterbi decoder
rxed_data=decodedata;
rxed_data=rxed_data(:)';
errors=0;
c=xor(data,rxed_data);
errors=nnz(c);
BER(si,o)=errors/length(data);
o=o+1;
 end % SNR loop ends here
 si=si+1;
end
for col=1:25;         %%%change if SNR loop Changed
    ber(1,col)=0;
for row=1:100;


        ber(1,col)=ber(1,col)+BER(row,col);
    end
end
ber=ber./100;
%%
figure
i=0:2:48;
semilogy(i,ber);
title('BER vs SNR');
ylabel('BER');
xlabel('SNR (dB)');
grid on
```

Output:



Problem 11: Write a MATLAB program to simulate Least Mean Square (LMS) algorithm to adaptively adjust the coefficients of an FIR filter.

Code:

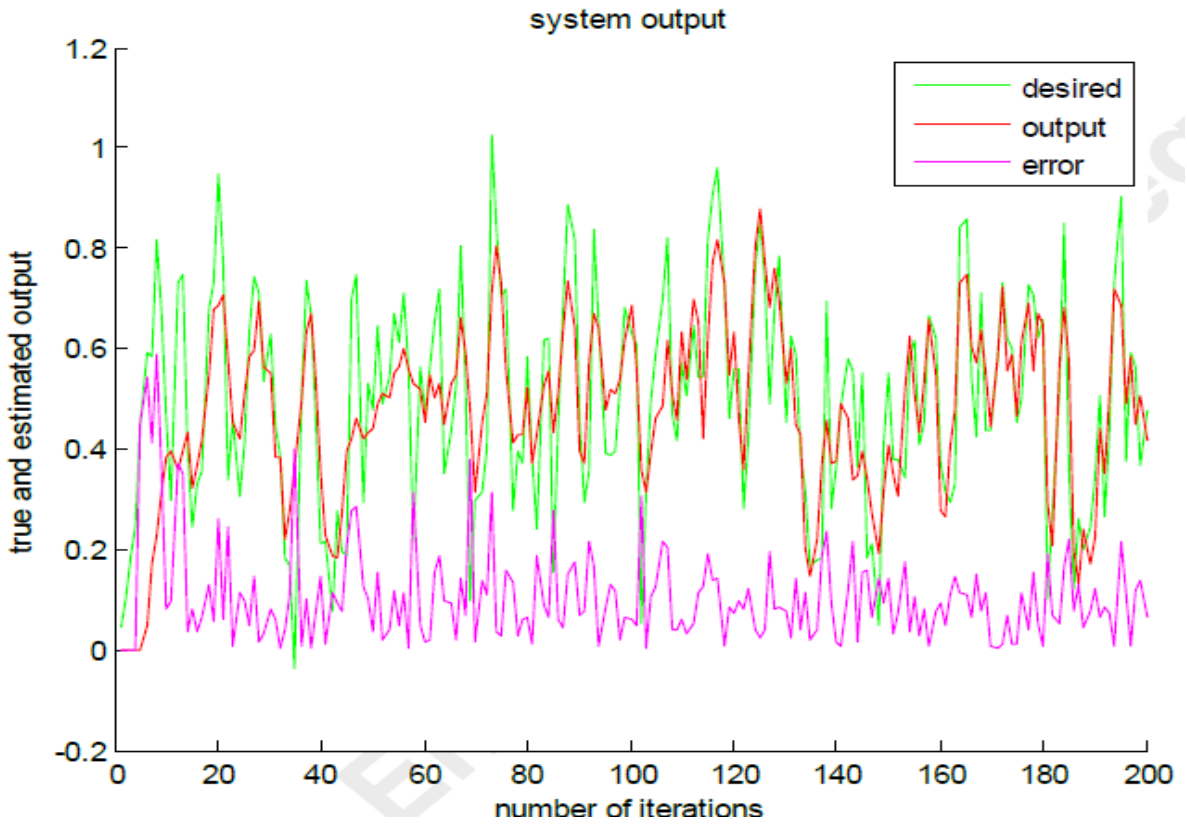
```
clc;clear all;close all;
N=input('enter the system order,N=');
M=input('enter the number of iterations,M=');
if((N>=2)&&(M>=2))
x=rand(M,1);
b=fir1(N-1,0.5);
n=0.1*randn(M,1);
d=filter(b,1,x)+n;
h=zeros(N,1);
Px=(1/length(x))*sum(x.^2);
mu=1/(5*N*Px);
for n=N:M
u=x(n:-1:n-N+1);
y(n)=h'*u;
e(n)=d(n)-y(n);
h=h+mu*u*e(n);
end
hold on;plot(d,'g');
```

```
plot(y(),'r');
semilogy((abs(e())),'m');
title('system output');
xlabel('number of iterations');
ylabel('true and estimated output');
legend('desired','output','error');
hold off;
figure,plot(b','k+');
hold on,plot(h,'r*');
legend('actual weights','estimated weights');
hold off;
title('comparison of actual weights and estimated weights');
else('system order and number of iterations should be greater than
1');
end
```

Input:

enter the system order,N=10

enter the number of iterations,M=100

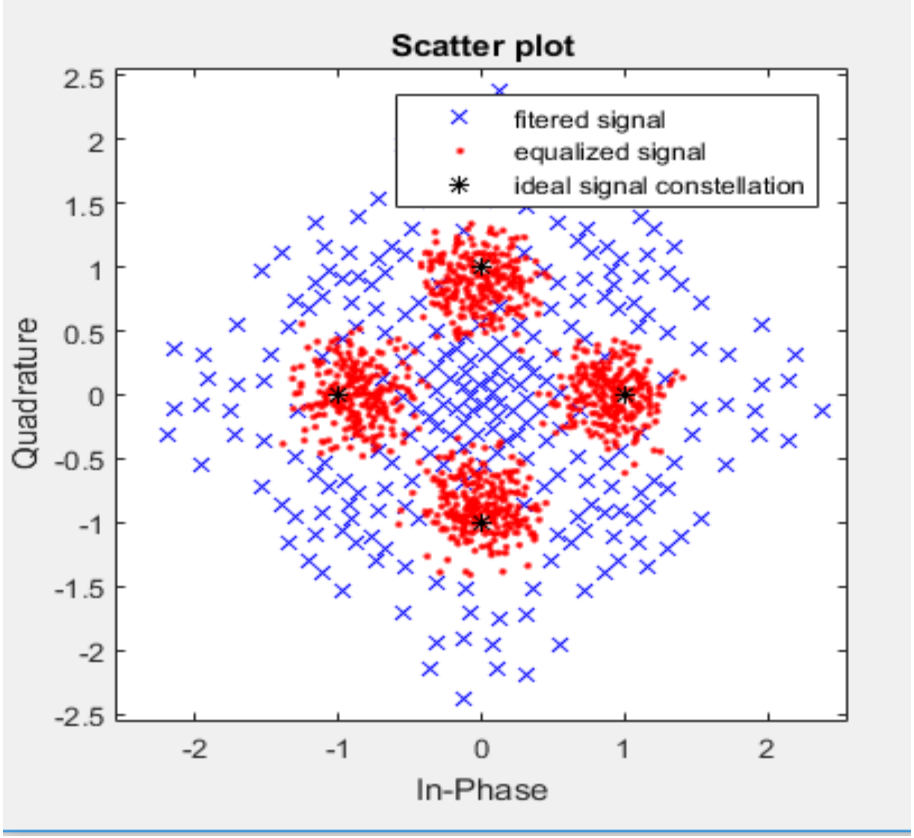


Problem 12: Write a MATLAB program to simulate the Zero Forcing Equalizer using MATLAB.

Code:

```
clc
clear all
close all
M=4;
msg=randint(1500,1,M);
modmsg=pskmod(msg,M);
sigconst=pskmod([0:M-1],M);
trainlen=500;
chan=[.986;.845;.237;.123+.31i];
filtmsg=filter(chan,1,modmsg);
eqobj =lineareq(8,lms(0.01),sigconst,1);
[symbolest,yd]=equalize(eqobj,filtmsg,modmsg(1:trainlen));
h=scatterplot(filtmsg,1,trainlen,'bx');hold on;
scatterplot(symbolest,1,trainlen,'r.',h);
scatterplot(sigconst,1,0,'k*',h);
legend('fitered signal','equalized signal','ideal signal
constellation');
hold off;
demodmsg_noeq=pskdemod(filtmsg,M);
demodmsg =pskdemod(yd,M);
[nnoeq,rnoeq]=symerr(demodmsg_noeq(trainlen+1:end),msg(trainlen+1:end
));
[neq,req] = symerr(demodmsg(trainlen+1:end),msg(trainlen+1:end));
disp('symbol error rate with equalizer:');
disp(req);
disp('symbol error rate without equalizer:');
disp(rnoeq)
```

Output:



Problem 13: Write a MATLAB code for Frequency Division Multiplexing.

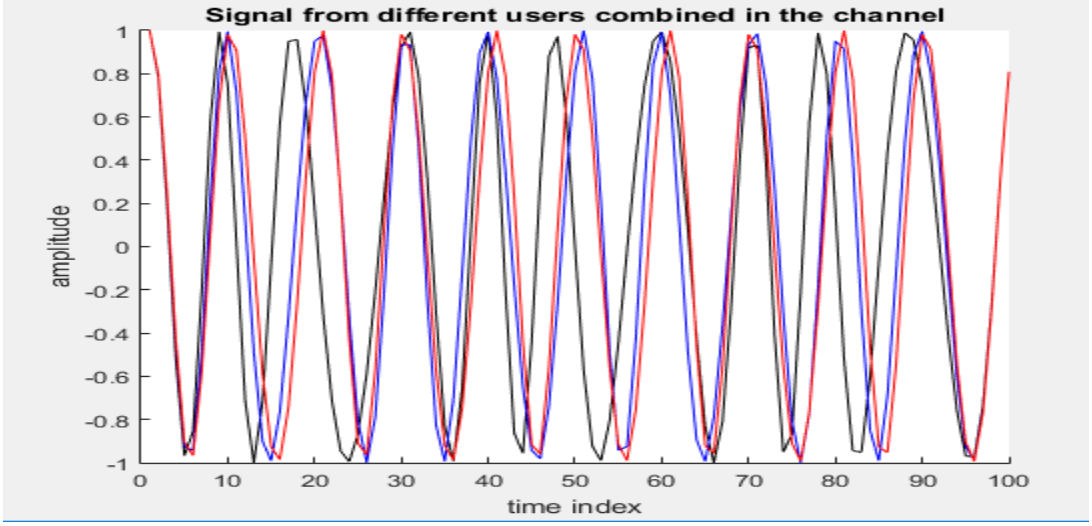
Code:

```
clc;
clear all;
close all;
samples=100;

nos=3;% number of users
mfreq=[30 40 50]; % modulating signal frequency in Hz
cfreq=[300 600 900freqdev=100; % choose frequency deviation
t=linspace(0,1000,samples);
parfor i=1:nos
m(i,:)=sin(2*pi*mfreq(1,i)*t);
end
parfor i=1:nos
y(i,:)=fmmod(m(i,:),cfreq(1,i),10*cfreq(1,i),freqdev);
end
parfor i=1:nos
z(i,:)=fmdemod(y(i,:),cfreq(1,i),10*cfreq(1,i),freqdev);
end
C = {'k','b','r','g','y',[.5 .6 .7],[.8 .2 .6],[.3 .2 .2]}; % Cell
```

```
for i=1:nos
figure (1)
hold on;
plot(y(i,:),'color',C{i});
xlabel('time index'); ylabel('amplitude'); title('Signal from
different users combined in the channel');
figure
subplot(3,1,1)
plot(m(i,:)) % modulating signal
xlabel('time index'); ylabel('amplitude'); title('modulating Signal
from user');
subplot(3,1,2)
plot(y(i,:),'color',C{i}); % modulated signal
xlabel('time index'); ylabel('amplitude'); title('modulated Signal
from user');
subplot(3,1,3)
plot(z(i,:),'color',C{i}); % demodulated signal
xlabel('time index'); ylabel('amplitude'); title('demodulated Signal
from user at the base station');
end
```

Output:



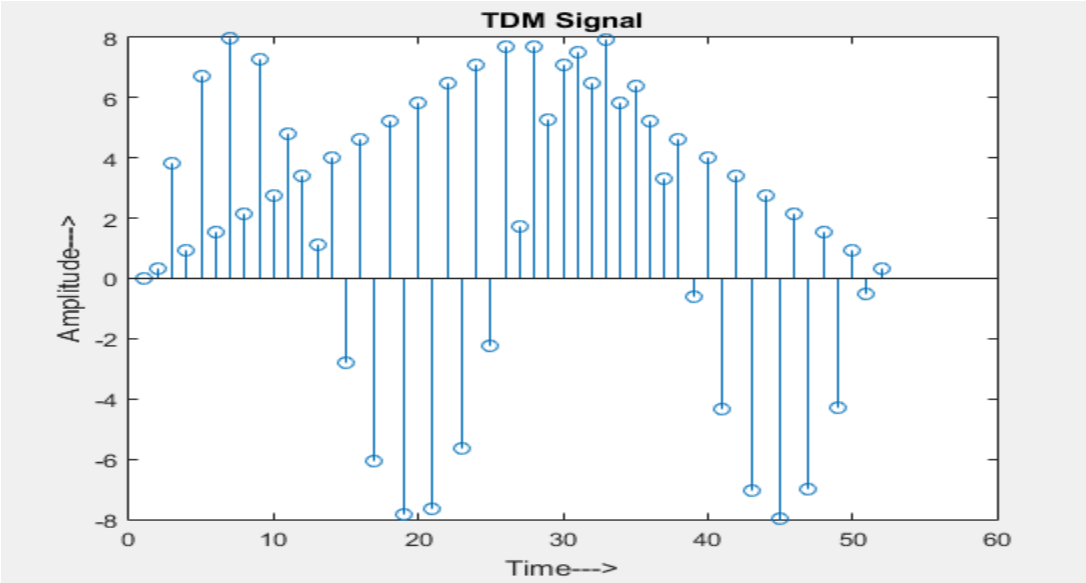
Problem 14; Write a MATLAB code for Time Division Multiplexing.

Code:

```
clc;
close all;
clear all;
x=0:.5:4*pi;                          % siganal taken upto 4pi
sig1=8*sin(x);                        % generate 1st sinusoidal
signal
l=length(sig1);
```

```
sig2=8*triang(l);                          % Generate 2nd traingular
Sigal
subplot(2,2,1);
plot(sig1);
title('Sinusoidal Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
subplot(2,2,2);
plot(sig2);
title('Triangular Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
subplot(2,2,3);
stem(sig1);
title('Sampled Sinusoidal Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
subplot(2,2,4);
stem(sig2);
title('Sampled Triangular Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
l1=length(sig1);
l2=length(sig2);
 for i=1:l1
  sig(1,i)=sig1(i);                          % Making Both row vector
to a matrix
  sig(2,i)=sig2(i);
 end
tdmsig=reshape(sig,1,2*l1);
figure
stem(tdmsig);
title('TDM Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
```

Output:



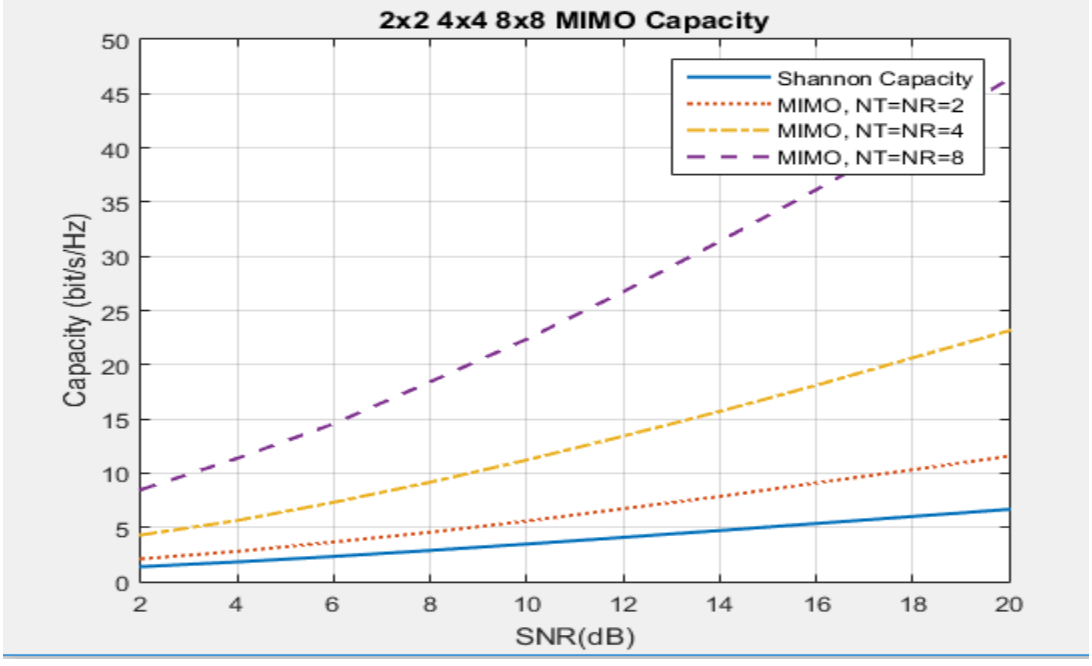
Problem 15: Write a MATLAB program to Simulation of 2 × 2, 4 × 4, 8 × 8 MIMO.

Code:

```
clc
clear
close all
%Shannon capacity
snr=0;
for i = 1:10
    snr = snr +2;
    c=(log(1+10^(snr/10)))/log(2);
    x(i)=snr;
    y(i)=c;
end
figure
plot(x,y,'-','LineWidth',1.5)
hold on
% capacity of MIMO Link with NR=2, NT=2
NR=2;
rand('state',456321)
snr=0;
for i=1:10;
    snr=snr+2;
    for j=1:10000;
        c(j)=(NR*log(1+(10^(snr/10))*abs(normrnd(0,1)))/log(2));
    end
    yy(i)=mean(c);
    xx(i)=snr;
end
plot(xx,yy,':','LineWidth',1.5)
% capacity of MIMO Link with NR=4, NT=4
NR=4;
```

```
rand('state',456321)
snr=0;
for i=1:10;
    snr=snr+2;
    for j=1:10000;
       c(j)=(NR*log(1+(10^(snr/10))*abs(normrnd(0,1)))/log(2));
    end
    yy(i)=mean(c);
    xx(i)=snr;
end
plot(xx,yy,'-.','LineWidth',1.5)
% capacity of MIMO Link with NR=8, NT=8
NR=8;
rand('state',456321)
snr=0;
for i=1:10;
    snr=snr+2;
    for j=1:10000;
        c(j)=(NR*log(1+(10^(snr/10))*abs(normrnd(0,1)))/log(2));
    end
    yy(i)=mean(c);
    xx(i)=snr;
end
plot(xx,yy,'--','LineWidth',1.5)
xlabel('SNR(dB)')
ylabel('Capacity (bit/s/Hz)')
grid on
legend('Shannon Capacity','MIMO, NT=NR=2','MIMO, NT=NR=4','MIMO, NT=NR=8',2)
title('2x2 4x4 8x8 MIMO Capacity')
```

Output:



Problem 16: Cyclic Code Generation Using MATLAB

Code:

```
clc;
n=7; % CODE LENGTH
k=4; % NUMBER OF MESSAGE BITS
disp('MESSAGE'); % RANDOM MESSAGE GENERATION
m=randint(2,k,[0,1]);
disp(m);
disp('POLYNOMIAL'); % GENERATOR POLYNOMIAL
pol=cyclpoly(n,k,'min');
disp(pol);
disp('CODE VECTOR'); % CODE VECTOR GENERATION
code=encode(m,n,k,'CYCLIC/FMT',pol);
disp(code);
disp('ERROR'); % RANDOM ERROR GENERATION
e=randerr(2,n,[1 0;0.8 0.2]);
disp(e);
disp('RECEIVED MATRIX'); % RECEIVED MATRIX
r=rem(plus(code,e),2);
disp(r);
[newmsg err cc]=decode(r,n,k,'CYCLIC'); % DECODING OF RECEIVED
MESSAGE
disp('DECODED RECEIVED VECTOR');
disp(cc);
disp('DECODED MESSAGE');
disp(newmsg);
```

Output:

MESSAGE

1 1 0 0

0 1 1 0

POLYNOMIAL

1 0 1 1

CODE VECTOR

0 1 0 1 1 0 0

0 0 1 0 1 1 0

ERROR

0 1 0 0 0 0 0

0 0 0 0 0 0 1

RECEIVED MATRIX

0 0 0 1 1 0 0

0 0 1 0 1 1 1

DECODED RECEIVED VECTOR

0 1 0 1 1 0 0

0 0 1 0 1 1 0

DECODED MESSAGE

1 1 0 0

0 1 1 0

Problem 17: Implementation Of Linear Block Codes Using MATLAB

Code:

```
clc;
clear
close all
 g=input('Enter The Generator Matrix: ');
disp('G=')
disp(g)
disp ('The Order of Linear block Code for given Generator Matrix
is:')
[n,k] = size(transpose(g));
disp('n=')
disp(n)
disp('k=')
disp(k)
for i = 1:2^k
    for j = k:-1:1
        if rem(i-1,2^(-j+k+1))>=2^(-j+k)
            u(i,j)=1;
        else
            u(i,j)=0;
        end
    end
end
```

```
u;
disp('The Possible Codewords are :')
c = rem(u*g,2);
disp('c=')
disp(c)
disp('The Minimum Hamming Distance dmin for given Block Code is= ')
d_min = min(sum((c(2:2^k,:))'));
disp('d_min')
disp(d_min)
r = input('Enter the Received Code Word:');
disp('r=')
disp(r)
p = g(:,n-k+2:n);
h = [transpose(p),eye(n-k)];
disp('Hammimg Code')
ht = transpose(h);
disp('ht=')
disp(ht)
disp('Syndrome of a Given Codeword is :')
s = rem(r*ht,2);
disp('s=')
disp(s)
for i = 1:1:size(ht)
    if(ht(i,1:3)==s)
        r(i) = 1-r(i);
        break;
    end
end
disp('The Error is in bit:')
disp('i=')
disp(i)
disp('The Corrected Codeword is :')
disp('r=')
disp(r)
```

Input:

Enter The Generator Matrix: [1 0 0 0 1 0 1;0 1 0 0 1 1 0;0 0 1 0 0 1 0]

Output:

The Possible Codewords are :

c=

0 0 0 0 0 0 0

0 0 1 0 0 1 0

0 1 0 0 1 1 0

0 1 1 0 1 0 0

1 0 0 0 1 0 1

1 0 1 0 1 1 1

1 1 0 0 0 1 1

1 1 1 0 0 0 1

The Minimum Hamming Distance dmin for given Block Code is=

d_min

2

Enter the Received Code Word:[1 1 0 0 1 0 1]

r=

1 1 0 0 1 0 1

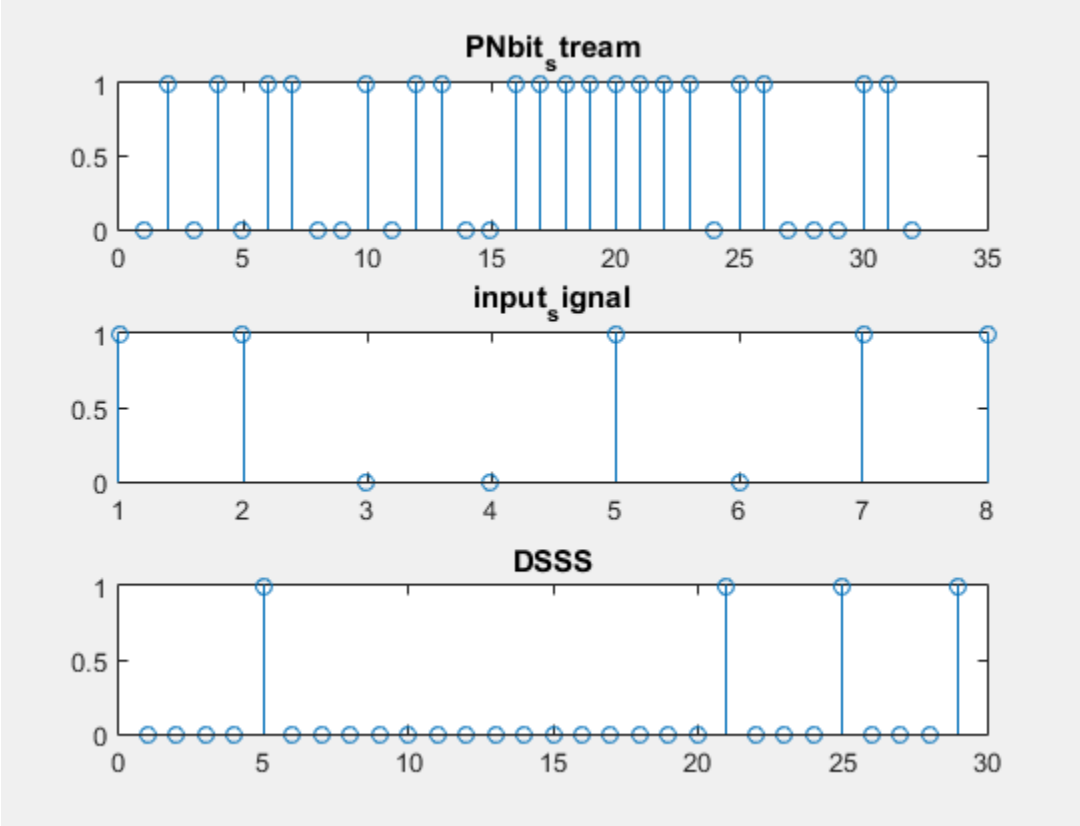
Problem 18: To obtain a Direct Sequence Spread Spectrum (DSSS) for a given input signal by simulating in MATLAB.

Code:

```
clc;
clear all;
close all;
PNbit_stream = round(rand(1,32));
input_signal=[1 1 0 0 1 0 1 1];
PNbit_stream;
for i=1:1:8
for j=1:4:32
for k=1:1:4
a(j)=xor(PNbit_stream(k+j-1),input_signal(i));
a(j);
end
end
end
subplot(3,1,1)
stem(PNbit_stream);
title('PNbit_stream');
subplot(3,1,2)
stem(input_signal);
title('input_signal');
subplot(3,1,3);
stem(a);
```

```
title('DSSS');
```

Output:



Problem 19: To perform the simulation of convolutional coding scheme and hamming code scheme using MATLAB.

Code:

```
clc;
clear all;
close all;
x=[1 1 0 0 1 0];
n=7;
k=5
t=poly2trellis(3,[n,k]);
code=convenc(x,t);
```

Output:

t =

numInputSymbols: 2

numOutputSymbols: 4

numStates: 4

nextStates: [4x2 double]

outputs: [4x2 double]

Hamming Code:

```
clc;
clear all;
close all;
n = 15; % Code length
k = 11;
data = randi([0 1],k,1);
encData = encode(data,n,k,'hamming/binary');
encData(4) = ~encData(4);
decData = decode(encData,n,k,'hamming/binary');
numerr = biterr(data,decData);
```

Output:

encData =0 0 1 1 0 1 0 0 1 0 1 0 0 0 0

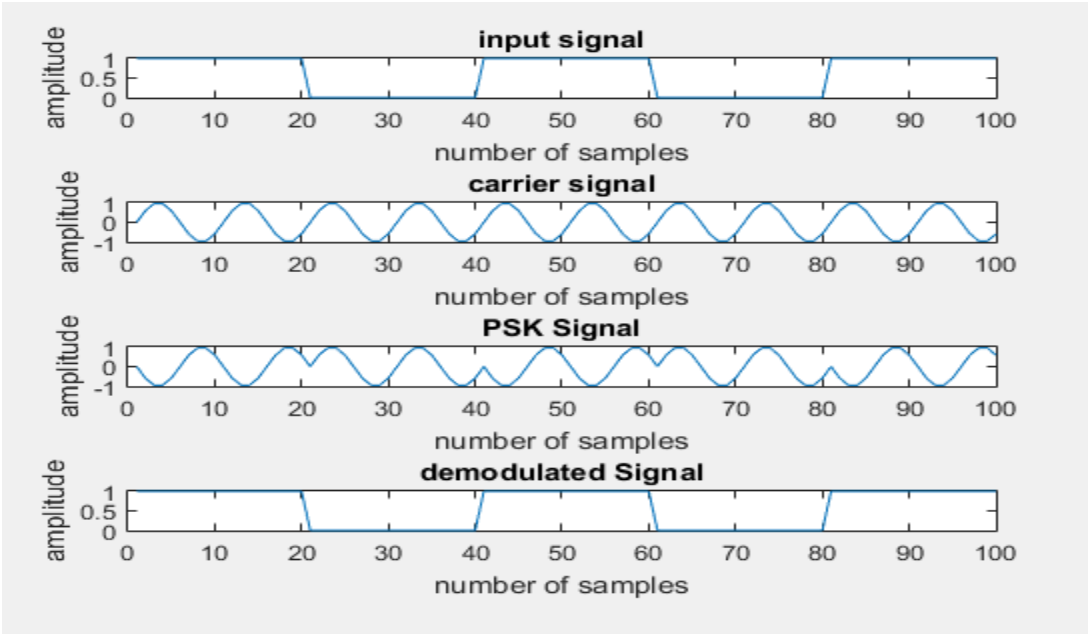
Problem 20: To perform the simulation of DPSK modulation and demodulation using MATLAB.

Code:

```
clear all;
close all;
n=100;
x=[ones(1,20) zeros(1,20) ones(1,20) zeros(1,20) ones(1,20)];
subplot(4,1,1);
plot(x);
title('input signal');
xlabel('number of samples');
ylabel('amplitude');
f=1*10^6;
fs=10*10^6;
for i=0:n-1
```

```
    d(i+1)=sin(2*pi*(f/fs)*i);
end
subplot(4,1,2);
plot(d);
title('carrier signal');
xlabel('number of samples');
ylabel('amplitude');
for i=0:n-1
if(x(i+1)==0)
x(i+1)=sin(2*pi*(f/fs)*i);
else
x(i+1)=sin(2*pi*(f/fs)*i+pi);
end
end
subplot(4,1,3);
plot(x);
title('PSK Signal');
xlabel('number of samples');
ylabel('amplitude');
% Demodulated signal
for i=0:n-1
if(x(i+1)==sin(2*pi*(f/fs)*i))
x(i+1)=0;
else
x(i+1)=1;
end
end
subplot(4,1,4);
plot(x);
title('demodulated Signal');
xlabel('number of samples');
ylabel('amplitude');
```

Output:


input signal
amplitude
1
0.5
0
0 10 20 30 40 50 60 70 80 90 100
number of samples
carrier signal
amplitude
1
0
-1
0 10 20 30 40 50 60 70 80 90 100
number of samples
PSK Signal
amplitude
1
0
-1
0 10 20 30 40 50 60 70 80 90 100
number of samples
demodulated Signal
amplitude
1
0.5
0
0 10 20 30 40 50 60 70 80 90 100
number of samples