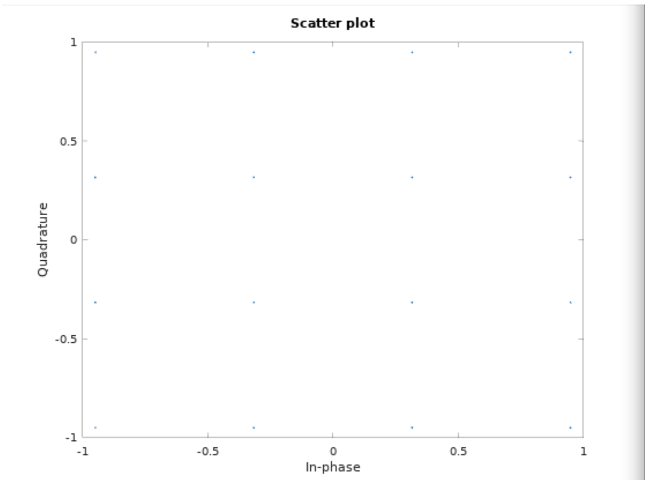
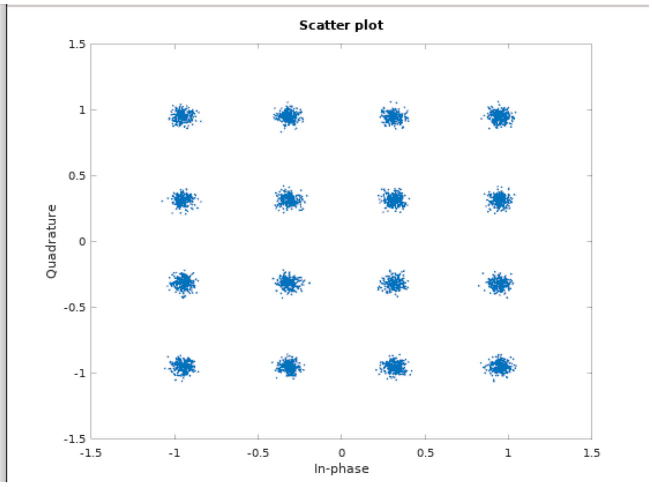
```
%                        <<Experiment-6 (16-Square QAM)>>
%                            << Objective-1 >>
% Aim: Simulation study of Performance of 16-Square QAM.
% Objective-1:Write a program to plot signal constellation diagram of received
%             16-Square QAM signal in the presence of AWGN.
% Objective-2:Write a program to plot Practical and Theoretical BER vs SNR graph
%             of received 16-Square QAM in the presence of AWGN for ML receiver.


% Name: Rathod Chittaranjan
% Roll No:32457

clc;
clear all;
close all;
pkg load communications
N = 16000;  % Number of bits to be transmitted using 16-Square QAM
          % Too large value may slow down the program
x = randi([0,1],1,N); % Random input bits generation
M = 16;     % Number of Symbols in 16-Square QAM
d = sqrt(2/5); % Average symbol energy is normalised to unity

% Symbol Generation
yy = [];
for i=1:4:length(x)
  if x(i)==0 && x(i+1)==0 & x(i+2)==0 & x(i+3)==0
    y = -3*d/2+j*(-3*d/2);
  elseif x(i)==0 && x(i+1)==0 & x(i+2)==0 & x(i+3)==1
    y = -3*d/2+j*(-d/2);
  elseif x(i)==0 && x(i+1)==0 & x(i+2)==1 & x(i+3)==1
    y = -3*d/2+j*(d/2);
  elseif x(i)==0 && x(i+1)==0 & x(i+2)==1 & x(i+3)==0
    y = -3*d/2+j*(3*d/2);

  elseif x(i)==0 && x(i+1)==1 & x(i+2)==0 & x(i+3)==0
    y = -d/2+j*(-3*d/2);
  elseif x(i)==0 && x(i+1)==1 & x(i+2)==0 & x(i+3)==1
    y = -d/2+j*(-d/2);
  elseif x(i)==0 && x(i+1)==1 & x(i+2)==1 & x(i+3)==1
    y = -d/2+j*(d/2);
  elseif x(i)==0 && x(i+1)==1 & x(i+2)==1 & x(i+3)==0
    y = -d/2+j*(3*d/2);

  elseif x(i)==1 && x(i+1)==1 & x(i+2)==0 & x(i+3)==0
    y = d/2+j*(-3*d/2);
  elseif x(i)==1 && x(i+1)==1 & x(i+2)==0 & x(i+3)==1
    y = d/2+j*(-d/2);
  elseif x(i)==1 && x(i+1)==1 & x(i+2)==1 & x(i+3)==1
    y = d/2+j*(d/2);
  elseif x(i)==1 && x(i+1)==1 & x(i+2)==1 & x(i+3)==0
    y = d/2+j*(3*d/2);
```

```
  elseif x(i)==1 && x(i+1)==0 & x(i+2)==0 & x(i+3)==0
    y = 3*d/2+j*(-3*d/2);
  elseif x(i)==1 && x(i+1)==0 & x(i+2)==0 & x(i+3)==1
    y = 3*d/2+j*(-d/2);
  elseif x(i)==1 && x(i+1)==0 & x(i+2)==1 & x(i+3)==1
    y = 3*d/2+j*(d/2);
  elseif x(i)==1 && x(i+1)==0 & x(i+2)==1 & x(i+3)==0
    y = 3*d/2+j*(3*d/2);
  endif
% Transmitted Symbols
yy = [yy y];
endfor
scatterplot(yy);    % Constellation Diagram without Noise
EbN0db = 20; % Change this value & run program to see the noisy constellation
EbN0 = 10^(EbN0db/10);

% AWGN Channel
n = (1/sqrt(2))*[randn(1,length(yy)) + 1j*randn(1,length(yy))];
sigma = sqrt(1/((log2(M))*EbN0));

% Received Symbols
r = yy + sigma*n;
scatterplot(r);    % Constellation Diagram with Noise
```

OUTPUT


Scatter plot
Quadrature
In-phase
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

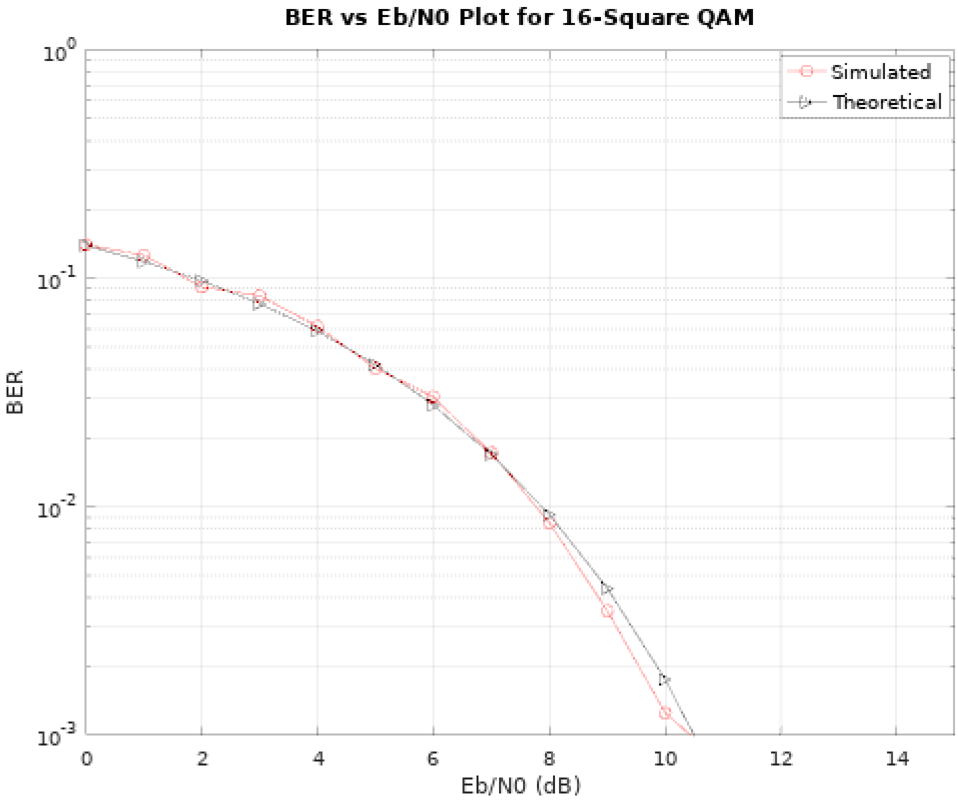

Scatter plot
Quadrature
In-phase
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

```
%                         <<Experiment-6 (16-Square QAM)>>
%                              << Objective-2 >>
% Aim: Simulation study of Performance of 16-Square QAM.
% Objective-1: Write a program to plot signal constellation diagram of received
%              16-Square QAM signal in the presence of AWGN.
% Objective-2: Write a program to plot Practical and Theoretical BER vs SNR graph
%              of received 16-Square QAM signal in the presence of AWGN for ML receiver.

% Name: Rathod Chittaranjan
% Roll No:32457


clc;
clear all;
close all;
pkg load communications
N = 4000;  % Number of bits to be transmitted using *16-Square QAM
         % Too large value may slow down the program
x = randi([0,1],1,N); % Random input bits generation
M = 16;     % Number of Symbols in 16-Square QAM
d = sqrt(2/5); % Average symbol energy is normalised to unity

% Symbol Generation
yy = [];
for i=1:4:length(x)
  if x(i)==0 && x(i+1)==0 & x(i+2)==0 & x(i+3)==0
    y = -3*d/2+j*(-3*d/2);
  elseif x(i)==0 && x(i+1)==0 & x(i+2)==0 & x(i+3)==1
    y = -3*d/2+j*(-d/2);
  elseif x(i)==0 && x(i+1)==0 & x(i+2)==1 & x(i+3)==1
    y = -3*d/2+j*(d/2);
  elseif x(i)==0 && x(i+1)==0 & x(i+2)==1 & x(i+3)==0
    y = -3*d/2+j*(3*d/2);

  elseif x(i)==0 && x(i+1)==1 & x(i+2)==0 & x(i+3)==0
    y = -d/2+j*(-3*d/2);
  elseif x(i)==0 && x(i+1)==1 & x(i+2)==0 & x(i+3)==1
    y = -d/2+j*(-d/2);
  elseif x(i)==0 && x(i+1)==1 & x(i+2)==1 & x(i+3)==1
    y = -d/2+j*(d/2);
  elseif x(i)==0 && x(i+1)==1 & x(i+2)==1 & x(i+3)==0
    y = -d/2+j*(3*d/2);

  elseif x(i)==1 && x(i+1)==1 & x(i+2)==0 & x(i+3)==0
    y = d/2+j*(-3*d/2);
  elseif x(i)==1 && x(i+1)==1 & x(i+2)==0 & x(i+3)==1
    y = d/2+j*(-d/2);
  elseif x(i)==1 && x(i+1)==1 & x(i+2)==1 & x(i+3)==1
    y = d/2+j*(d/2);
  elseif x(i)==1 && x(i+1)==1 & x(i+2)==1 & x(i+3)==0
```

```
    y = d/2+j*(3*d/2);

  elseif x(i)==1 && x(i+1)==0 & x(i+2)==0 & x(i+3)==0
    y = 3*d/2+j*(-3*d/2);
  elseif x(i)==1 && x(i+1)==0 & x(i+2)==0 & x(i+3)==1
    y = 3*d/2+j*(-d/2);
  elseif x(i)==1 && x(i+1)==0 & x(i+2)==1 & x(i+3)==1
    y = 3*d/2+j*(d/2);
  elseif x(i)==1 && x(i+1)==0 & x(i+2)==1 & x(i+3)==0
    y = 3*d/2+j*(3*d/2);
  endif
% Transmitted Symbols
yy = [yy y];
endfor

% Detection based on euclidean distance
ber_simulated = [];
ber_theoretical = [];
ref_symbols = [-3*d/2+j*(-3*d/2) -3*d/2+j*(-d/2) -3*d/2+j*d/2 -3*d/2+j*(3*d/2) -d/2+j*(-3*d/2) -
d/2+j*(-d/2) -d/2+j*d/2 -d/2+j*(3*d/2) d/2+j*(-3*d/2) d/2+j*(-d/2) d/2+j*d/2 d/2+j*(3*d/2)
3*d/2+j*(-3*d/2) 3*d/2+j*(-d/2) 3*d/2+j*d/2 3*d/2+j*(3*d/2)];
for EbN0db = 0:15
EbN0 = 10^(EbN0db/10);
n = (1/sqrt(2))*[randn(1,length(yy)) + 1j*randn(1,length(yy))];
sigma = sqrt(1/((log2(M))*EbN0)); % Symbol energy is normalized to Unity
r = yy + sigma*n;

% Calculation of Euclidian Distances of received symbols from reference symobols
min_dist_index = [];
for i=1:length(r)
  Dist = [];
  for k=1:length(ref_symbols)
  dist = sqrt((real(r(i))-real(ref_symbols(k)))^2 + (imag(r(i))-imag(ref_symbols(k)))^2);
  Dist = [Dist dist];
  endfor
  min_dist_index = [min_dist_index find(Dist==min(Dist))];
endfor

% Estimation of Bits
x_estimated = [];
for i=1:length(r)
  if ref_symbols(min_dist_index(i))== -3*d/2+j*(-3*d/2);
    x_estimated = [x_estimated 0 0 0 0];
  elseif ref_symbols(min_dist_index(i))== -3*d/2+j*(-d/2);
    x_estimated = [x_estimated 0 0 0 1];
  elseif ref_symbols(min_dist_index(i))== -3*d/2+j*(d/2);
    x_estimated = [x_estimated 0 0 1 1];
  elseif ref_symbols(min_dist_index(i))== -3*d/2+j*(3*d/2);
    x_estimated = [x_estimated 0 0 1 0];

  elseif ref_symbols(min_dist_index(i))== -d/2+j*(-3*d/2);
    x_estimated = [x_estimated 0 1 0 0];
```

```
  elseif ref_symbols(min_dist_index(i))== -d/2+j*(-d/2);
    x_estimated = [x_estimated 0 1 0 1];
  elseif ref_symbols(min_dist_index(i))== -d/2+j*(d/2);
    x_estimated = [x_estimated 0 1 1 1];
  elseif ref_symbols(min_dist_index(i))== -d/2+j*(3*d/2);
    x_estimated = [x_estimated 0 1 1 0];

  elseif ref_symbols(min_dist_index(i))== d/2+j*(-3*d/2);
    x_estimated = [x_estimated 1 1 0 0];
  elseif ref_symbols(min_dist_index(i))== d/2+j*(-d/2);
    x_estimated = [x_estimated 1 1 0 1];
  elseif ref_symbols(min_dist_index(i))== d/2+j*(d/2);
    x_estimated = [x_estimated 1 1 1 1];
  elseif ref_symbols(min_dist_index(i))== d/2+j*(3*d/2);
    x_estimated = [x_estimated 1 1 1 0];

  elseif ref_symbols(min_dist_index(i))== 3*d/2+j*(-3*d/2);
    x_estimated = [x_estimated 1 0 0 0];
  elseif ref_symbols(min_dist_index(i))== 3*d/2+j*(-d/2);
    x_estimated = [x_estimated 1 0 0 1];
  elseif ref_symbols(min_dist_index(i))== 3*d/2+j*(d/2);
    x_estimated = [x_estimated 1 0 1 1];
  elseif ref_symbols(min_dist_index(i))== 3*d/2+j*(3*d/2);
    x_estimated = [x_estimated 1 0 1 0];
  endif
endfor

% BER Computation
ber_simulated =[ber_simulated sum(x~=x_estimated)/N];
ber_theoretical = [ber_theoretical (3/(2*log2(M)))*erfc(sqrt(2*EbN0/5))];
endfor
EbN0db = 0:15;

% BER Plotting
semilogy(EbN0db, ber_simulated, 'ro-', EbN0db, ber_theoretical, 'k>-');
title('BER vs Eb/N0 Plot for 16-Square QAM');
xlabel('Eb/N0 (dB)');
ylabel('BER');
grid on;
legend('Simulated', 'Theoretical');
axis([0 15 10^-3 10^0]);
```

OUTPUT



CONCLUSION:

In this practical we understand the signal constellation diagram of received 16-Square QAM signal in the presence of AWGN.

Figure a and Figure b shows the constellation diagram of the transmitter and receiver of the system. Similarly, the constellation diagram at the transmitting end is identical to the ideal

constellation map. Due to the presence of noise on the transmission channel, the constellation

at the receiving end has a certain deviation from the ideal constellation point. The size of the

deviation can be used to visually judge the performance of the system. • We also write the program for plot Practical and Theoretical BER vs SNR graph of received 16

Square QAM in the presence of AWGN for ML receiver.