BÁO CÁO THÍ NGHIỆM THÔNG TIN SỐ

Sinh viên thực hiện : **Bùi Văn Tài**

Lớp : **ĐTVT 05-K56**

Mã sinh viên : **20112102**

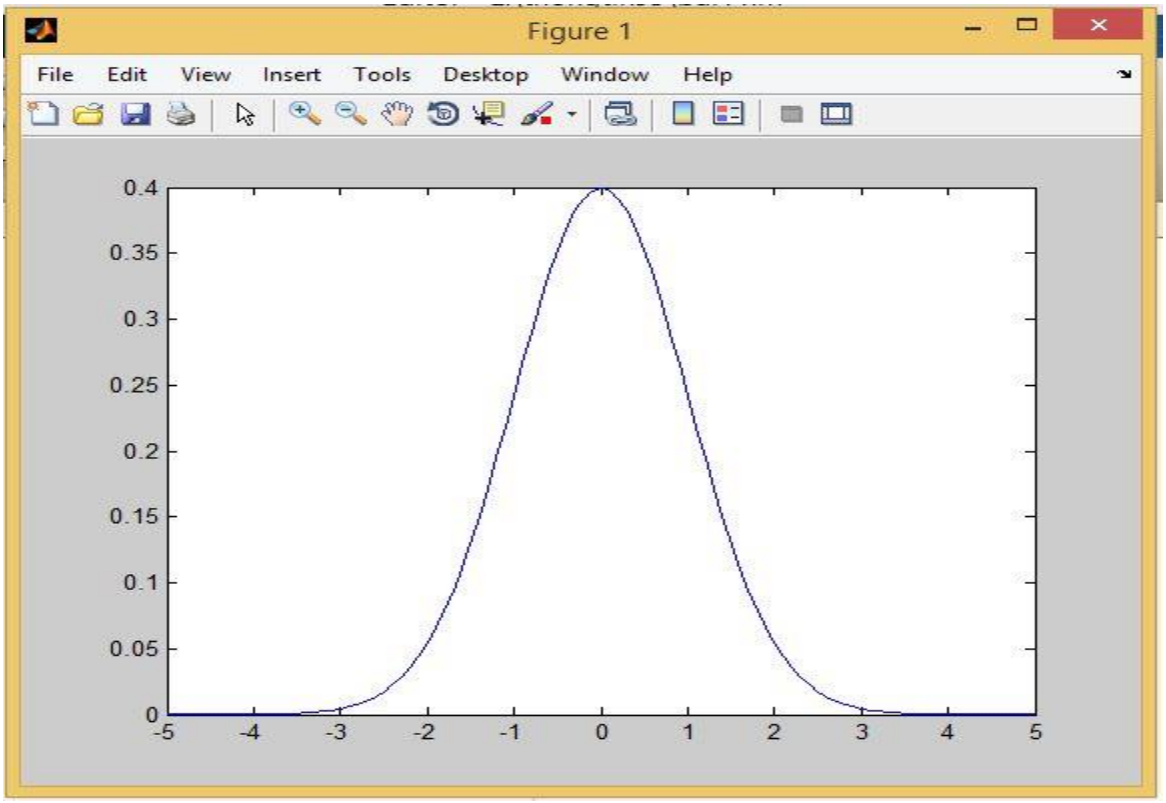
Bài 1 : Quá trình ngẫu nhiên của tín hiệu

Bài 1.1

Code :

```
x = -5:0.1:5;
px = (1/sqrt(2*pi))*exp(-x.^2/2);
plot(x,px);
```

Figure

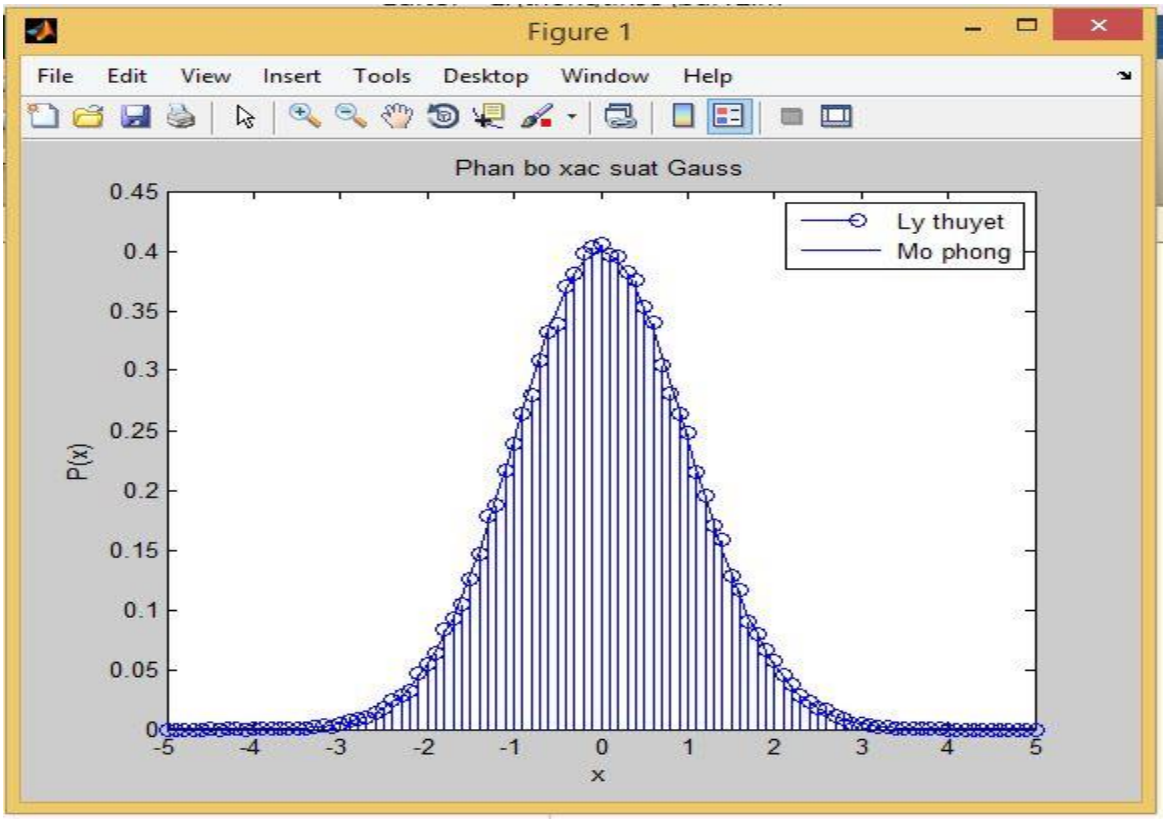


Bài 1.2

Code :

```
len = 100000;
x = randn(1,len);
step = .1;
k = -5:step:5;
px = hist(x,k)/len/step;
stem(k,px);
Px_lythuyet = exp(-k.^2/2)/sqrt(2*pi);
hold on;
plot(k,Px_lythuyet);
title(' Phan bo xac suat Gauss ');
xlabel('x');
ylabel('P(x)');
legend(' Ly thuyet',' Mo phong ');
hold off;
```

Figure



Bài 2 : Lượng tử hóa tuyến tính

Bài 2.2

Code :

Hàm lquan :

```
function [indx qy] = lquan(x,xmin,xmax,nbit)

nlevel = 2^nbit;
q = (xmax-xmin)/nlevel;
[indx qy] = quantiz(x,xmin+q:q:xmax-q,xmin+q/2:q:xmax-q/2);
```

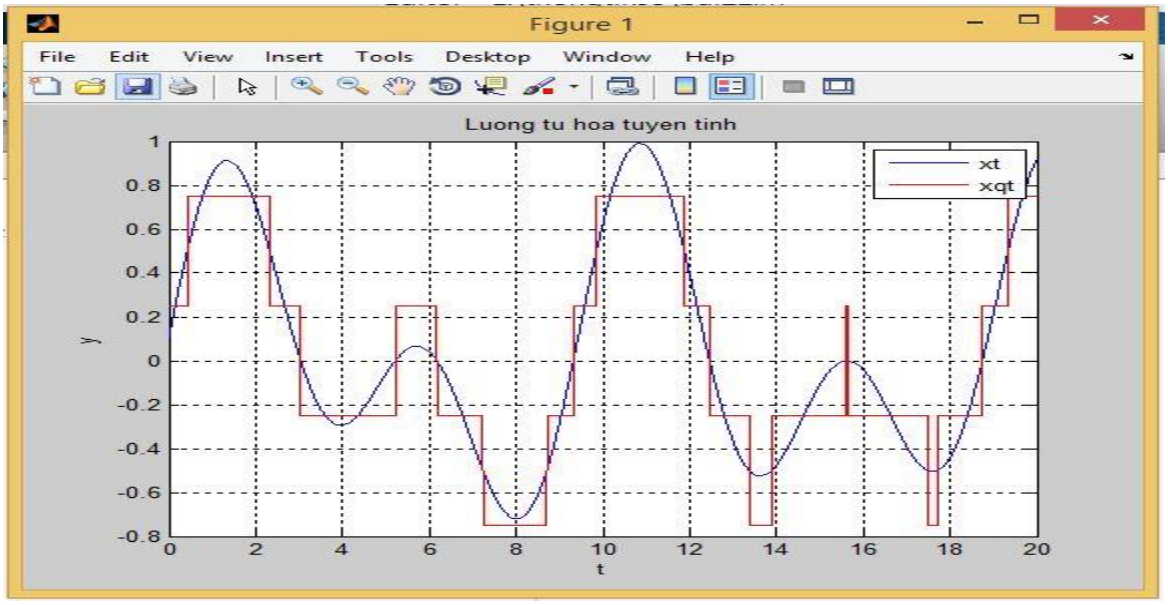
Hàm chính :

```
t = 0:.01:20;

xt = sin(randn()+t).*cos(rand()*t);

[inx xqt] = lquan(xt,-1,1,randint(1,1,3)+2);
plot(t,xt,'b',t,xqt,'r');
grid on;
title(' Luong tu hoa tuyen tinh ');
xlabel ('t');
ylabel ('y');
legend('xt','xqt');
```

Figure :



Bài 3 : Tạp âm lượng tử trong kĩ thuật lượng tử hóa tuyến tính

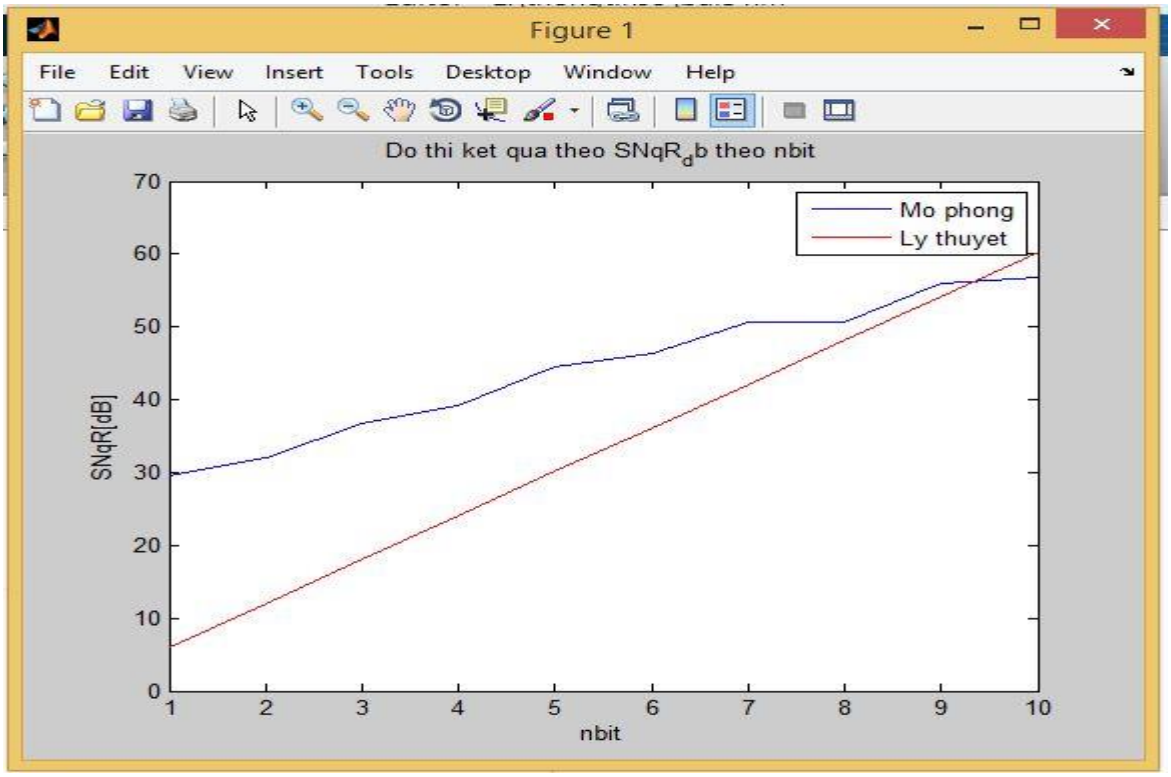
Bài 3.1

Code :

```
rand(1,len)
N = 1000;
x = 2*rand(1,N)-1;                  %x phan bo deu tu -1 den 1
nbit = 1:10;                        %denso bit luong tu tu 1 den 10
SNqR = zeros(size(nbit));           % khoi tao mang SNqR chua ket qua
SNqR_lt = 6.02*nbit;                % khoi tao mang SNqR tinh theo ly thuyet
Ps = sum(x.^2)/N;                   % cong suat tin hieu x theo (3-3)
for i=1:size(nbit,2)                % size(n,2)tra ve so cot cua n
[inx xq] = lquan(x,-1,1,nbit(i));   % luong tu hoa x voi so bit nbit i luu vao xq
eq(i) = x(i)-xq(i);                 % tính sai so  eq
Pq = (eq(i))/N;                     % tinh cong suat tap am luong tu Pq theo 3-4
SNqR(i) = 10*log10(Ps/Pq);          % Tính SNqR(i)
end;
plot(nbit,SNqR,'b',nbit,SNqR_lt,'r');  % Ve do thi ket qua SNqR_db theo nbit
title(' Do thi ket qua theo SNqR_db theo nbit ');
xlabel ('nbit');
ylabel ('SNqR[dB]');
legend('Mo phong','Ly thuyet');
```

Figure :

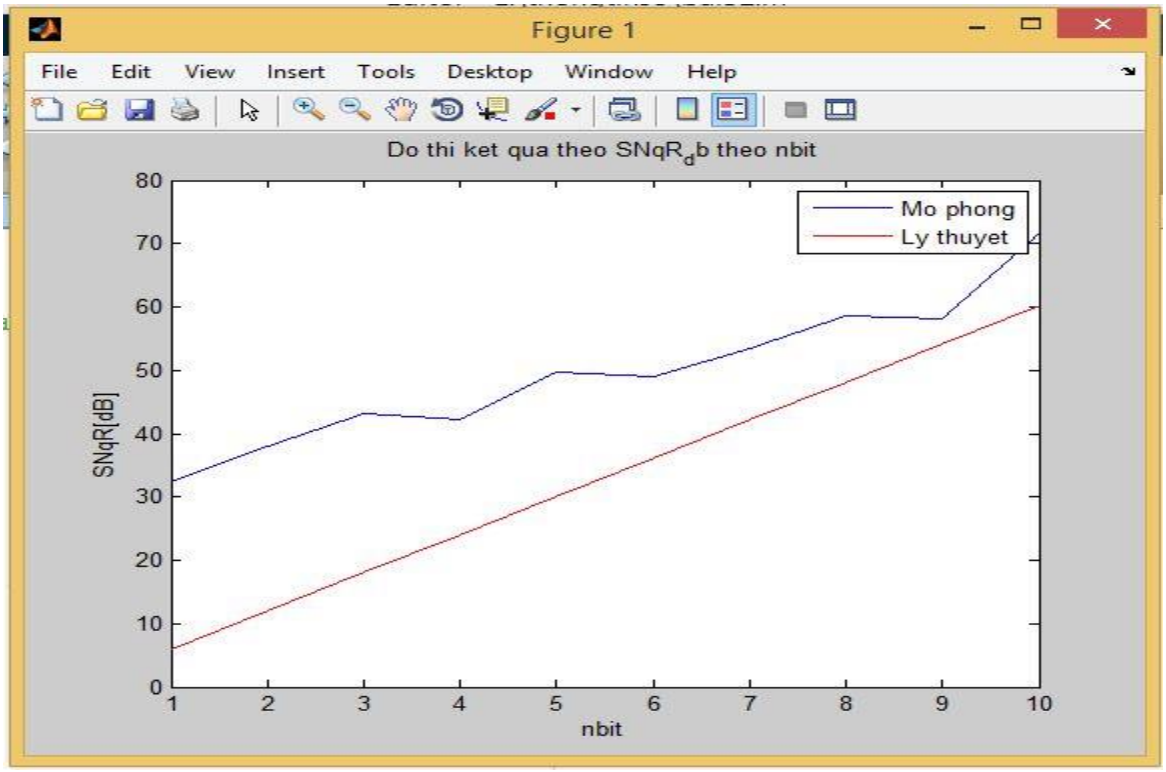
Bài 3.2



Code :

```
rand(1,len)
N = 1000;
x = sin(linspace(1,5,N));            %x phan bo deu tu -1 den 1
nbit = 1:10;                         %denso bit luong tu tu 1 den 10
SNqR = zeros(size(nbit));            % khoi tao mang SNqR chua ket qua
SNqR_lt = 6.02*nbit;                 % khoi tao mang SNqR tinh theo ly thuyet
Ps = sum(x.^2)/N;                    % cong suat tin hieu x theo (3-3)
for i=1:size(nbit,2)                 % size(n,2)tra ve so cot cua n
[inx xq] = lquan(x,-1,1,nbit(i));    % luong tu hoa x voi so bit nbit i luu vao xq
eq(i) = x(i)-xq(i);                  % tính sai so  eq
Pq = (eq(i))/N;                      % tinh cong suat tap am luong tu Pq theo 3-4
SNqR(i) = 10*log10(Ps/Pq);           % Tính SNqR(i)
end;
plot(nbit,SNqR,'b',nbit,SNqR_lt,'r');  % Ve do thi ket qua SNqR_db theo nbit
title(' Do thi ket qua theo SNqR_db theo nbit ');
xlabel ('nbit');
ylabel ('SNqR[dB]');
legend('Mo phong','Ly thuyet');
```

Figure :



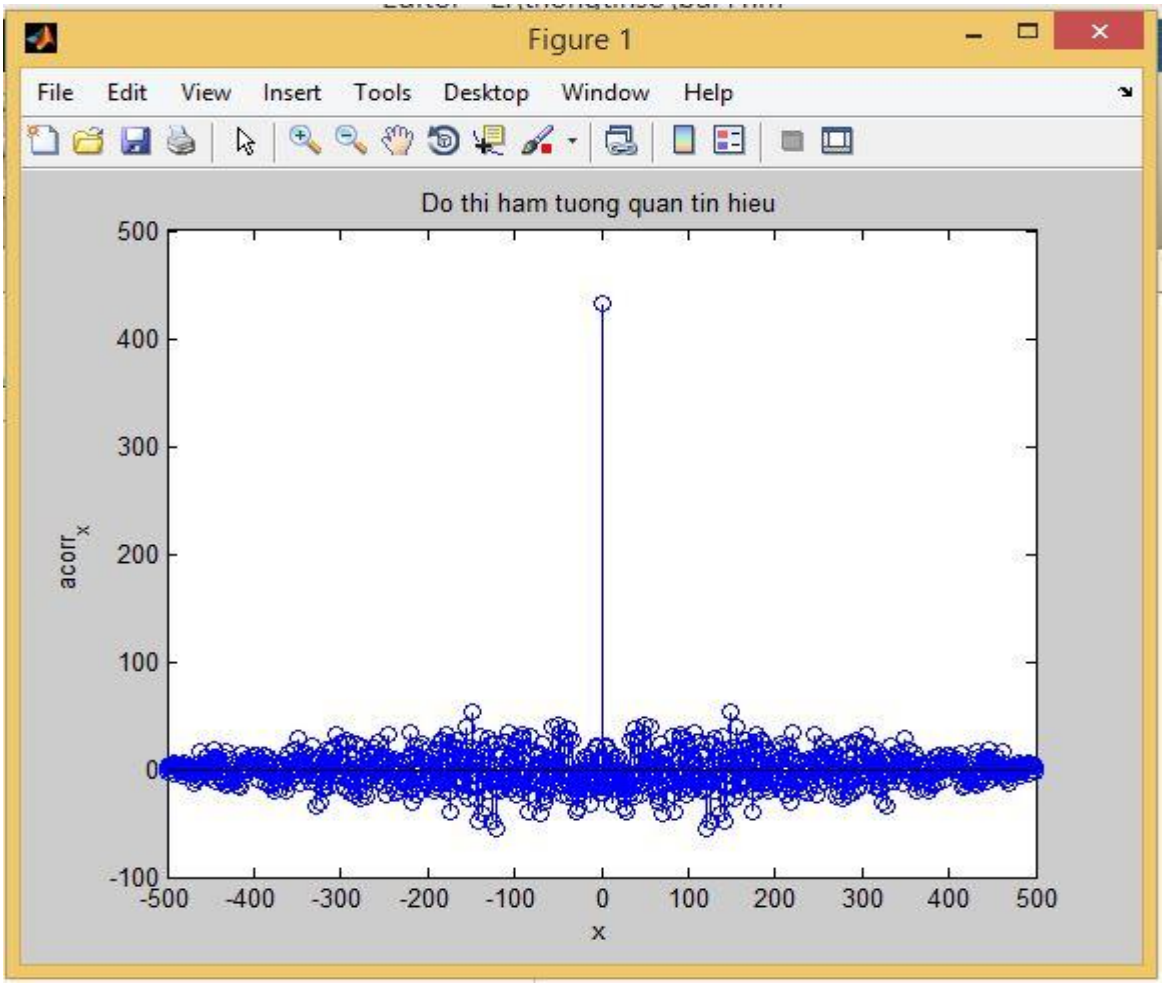
Bài 4 : Mật độ phổ năng lượng và hàm tự tương quan của tín hiệu

Bài 4.1

Code :

```
L=500;
x= randn(1,L);                % Tao 1 vecto ngau nhien co 500 phan tu
[x acorr_x] = xcorr(x);       % tinh ham tu tuong quan cua vecto tin hieu x
stem(acorr_x,x);              % Ve do thi ham tuong quan
title ('Do thi ham tuong quan tin hieu');
xlabel('x');
ylabel('acorr_x');
```

Figure :

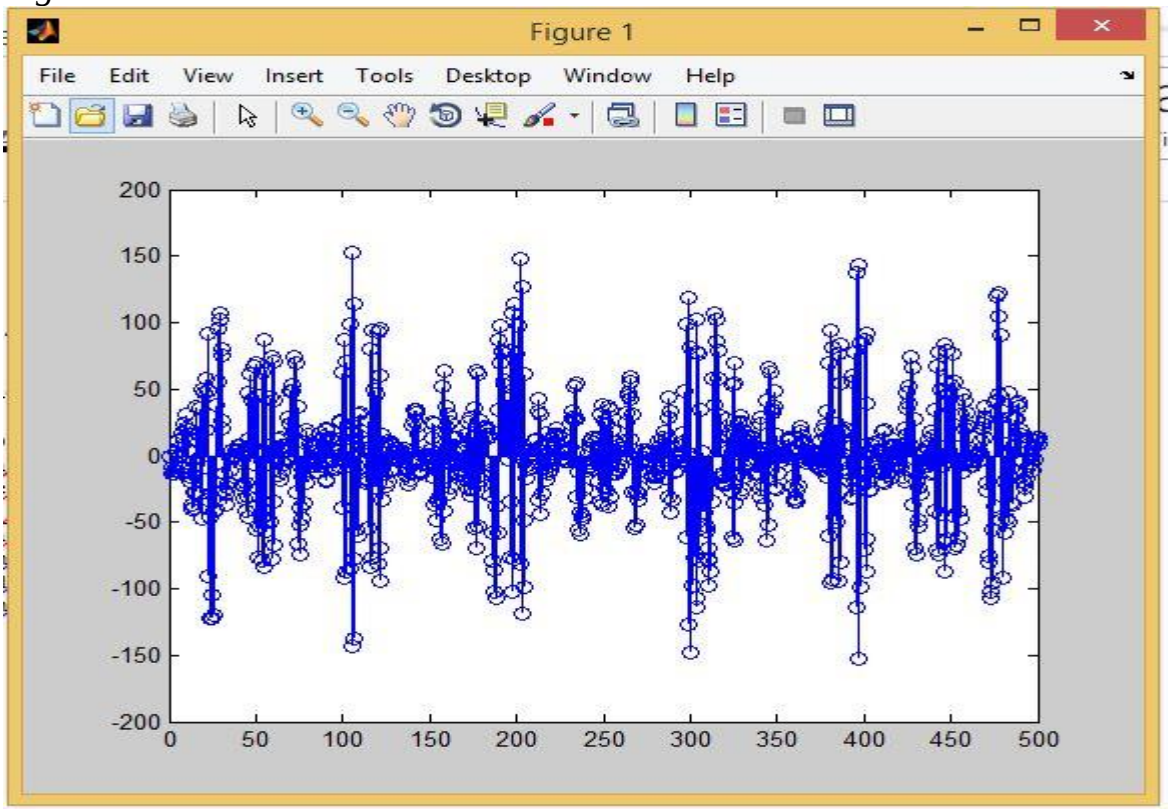


Bài 4.2

Code :

```
L=50;
x=randn(1,L);       % Tao 1 vecto ngau nhien co 50 phan tu
y=xcorr(x);          % tính y=xcorr(x)
esd_x=(fft(x,500)).^2;    % Ham tra ve bien doi Fourier roi rac 500 diem
ft_acorr_x=fft(y,500);
stem(esd_x);
hold on;
stem(ft_acorr_x);
```

Figure :



Bài 5 : Mã đường dây NRZ

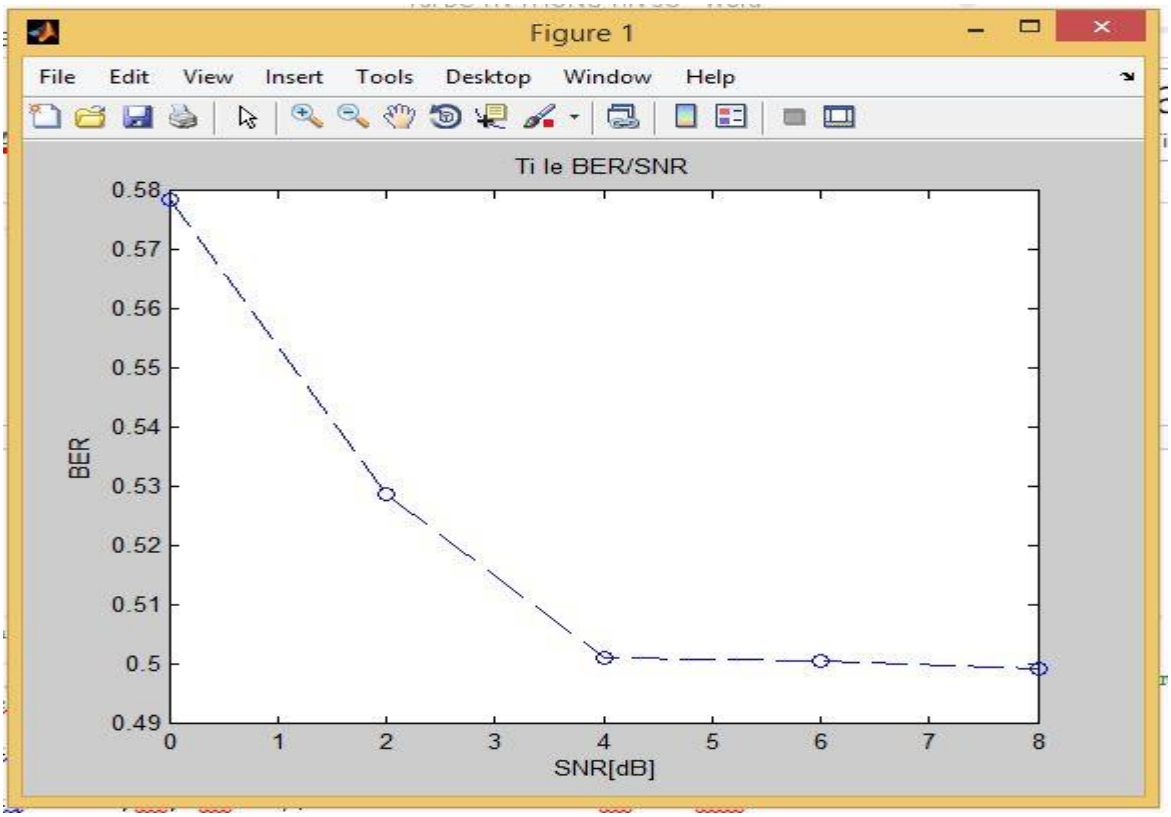
Bài 5.1

Code :

```
len=100000;  % do dai dong bit mo phong
SNR_db=0:2:8; % tao vecto SNR_db=0 2 4 6 8
SNR=10.^(SNR_db/10); % doi SNR tu decibel sang lan
bsignal =randint(1,len); % tao dong bit ngau nhien co do dai len
NRZ_signal = bsignal*2-1; % bien doi dong bit 0 1 sang -1 1
N0 =1./SNR; % phuong sai cua tap am = cong suat tap am
% cho tin hieu di qua kenh nhieu trang va dai dieu che
for i=1:length(SNR_db)
   noise = sqrt(N0(i)*randn(1,len)); % tao tap am noise
   r_signal=NRZ_signal+noise; %tin hieu thu duoc= tin hieu NRZ ben phat+tap
am noise
   NRZ_decoded= sign(r_signal);  % giai ma tin hieu NRZ thu duoc
```

```
    Pe(i)=symerr(NRZ_signal,NRZ_decoded)/len; % dem so bit loi thong %qua
ham symerr() roi chia cho do dai dong bit, ra ti so bit loi
end
plot(SNR_db,Pe,'bo--');              % Ve do thi
title(' Ti le BER/SNR ');
xlabel (' SNR[dB]');
ylabel (' BER ');
```

Figure



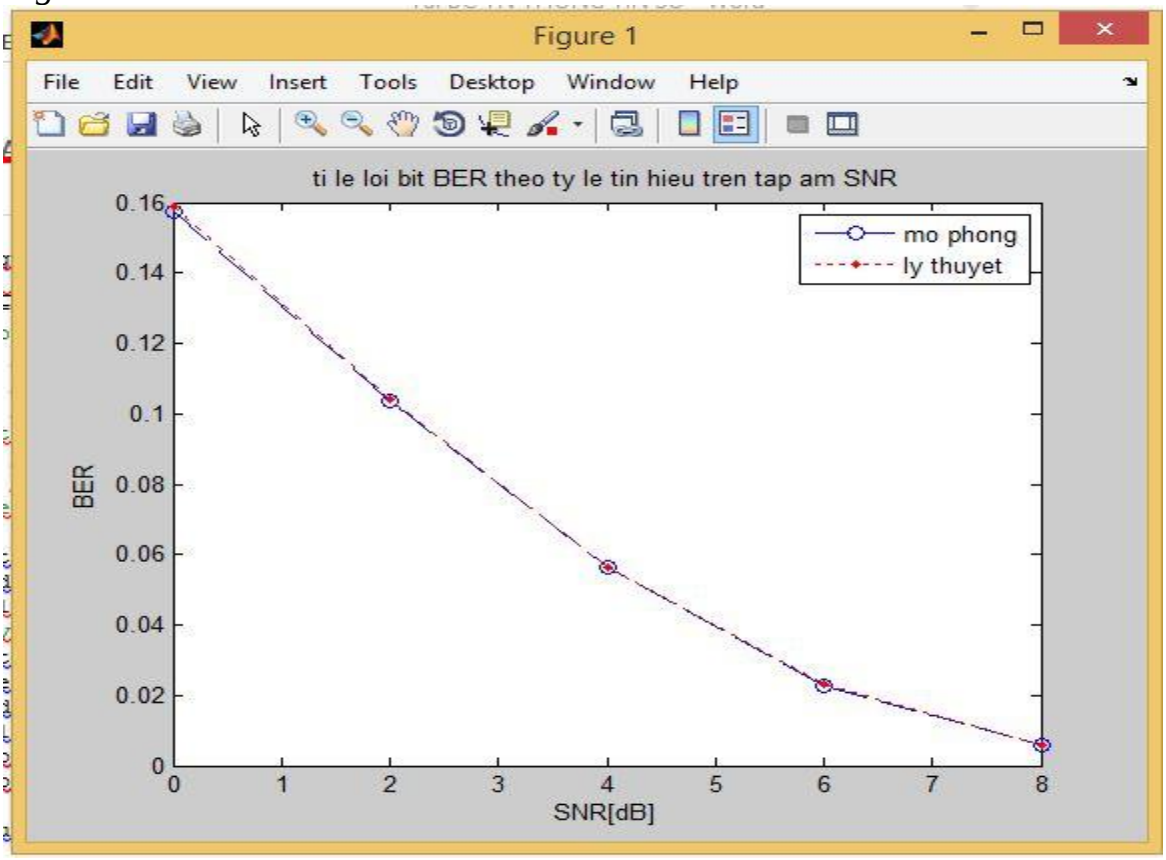
Bài 5.2

Code :

```
len = 100000;    %Do dai dong bit mo phong
SNR_db = 0:2:8;  %Tao vecto SNR_db = 0 2 4 6 8
SNR = 10.^(SNR_db/10);   %Doi SNR tu decibel sang lan
bsignal = (rand(1,len) >= 0.5);   %Tao dong bit ngau nhien do dai len
NRZ_signal = bsignal*2 - 1;   %Bien doi dong bit 0 1 sang -1 1
N0 = 1./SNR;     %Phuong sai cua tap am = cong suat tap am
```

```
%Cho tin hieu di qua kenh nhieu trang va giai dieu che
for i=1:length(SNR_db)
    noise = sqrt(N0(i))*randn(1,len);   %Tao tap am noise
    r_signal = NRZ_signal + noise;      %Tin hieu thu duoc = Tin hieu NRZ ben phat
+ tap am noise
    NRZ_decoded = sign(r_signal);       %Giai ma tin hieu NRZ thu %duoc
    Pe(i) = symerr(NRZ_signal,NRZ_decoded)/len; %Dem so bit loi thong qua ham
symerr() roi chia cho do dai dong bit ra ty so bit loi
end
plot(SNR_db,Pe,'bo--');
hold on;
Pe_lythuyet = (1/2)*(1-erf(sqrt(SNR/2))); % Tinh ti so bit loi theo %ly thuyet
plot(SNR_db,Pe_lythuyet,'r.:');
legend('mo phong','ly thuyet');
hold off;
title('ti le loi bit BER theo ty le tin hieu tren tap am SNR');
xlabel('SNR[dB]');
ylabel('BER');
```

Figure :



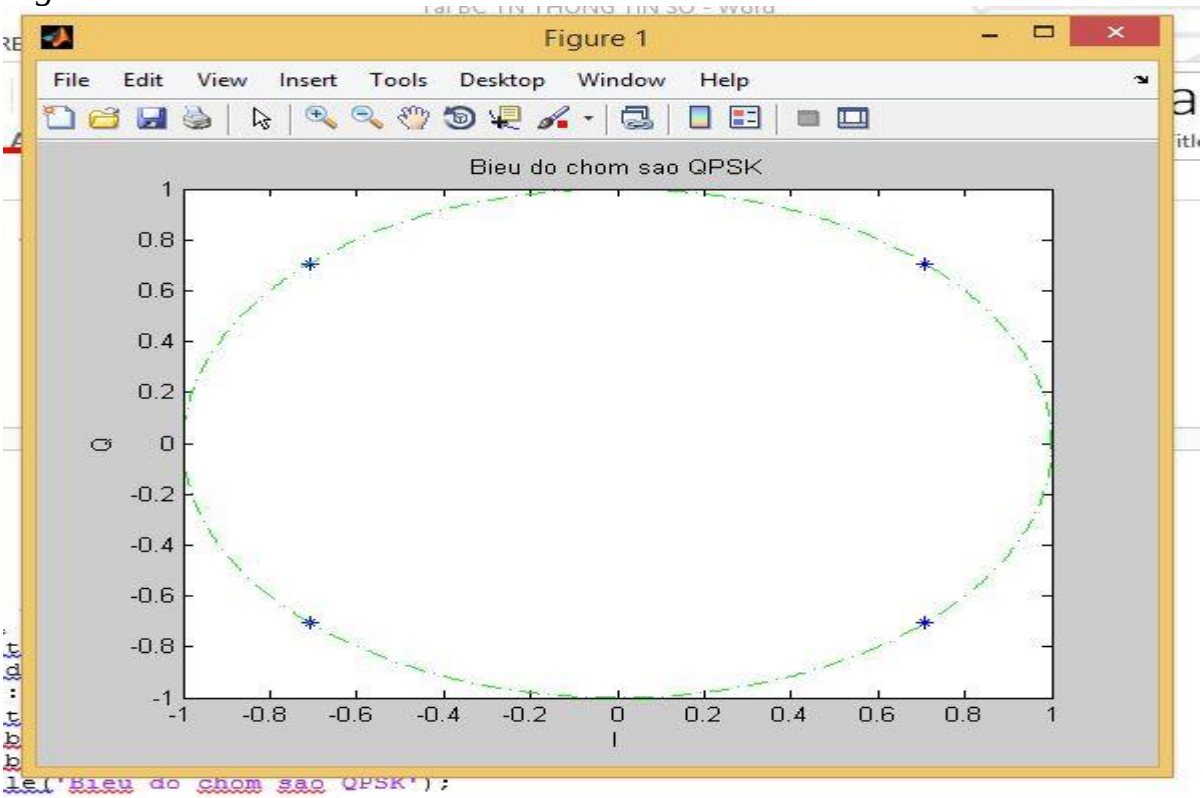
Bài 6 : Điều chế QPSK

Bài 6.1

Code :

```
len=50000; % do dai dong bit mo phong
bsignal= randint(1,len);  %tao dong bit ngau nhien do dai len
qpsk_signal=[];
for i=1:2:length(bsignal)
   if bsignal(i)==0 & bsignal(i+1)==0   % anh xa tin hieu 00 thanh -1+j
      qpsk_signal((i+1)/2)=exp(j*3*pi/4);
   elseif bsignal(i)==0 & bsignal(i+1)==1  % anh xa tin hieu 01 thanh -1-j
      qpsk_signal((i+1)/2)=exp(j*5*pi/4);
   elseif bsignal(i)==1 & bsignal(i+1)==1   % anh xa tin hieu 11 thanh 1-j
      qpsk_signal((i+1)/2)=exp(j*7*pi/4);
   elseif bsignal(i)==1 & bsignal(i+1)==0   % anh xa tin hieu 10 thanh 1+j
      qpsk_signal((i+1)/2)=exp(j*pi/4);
   end
end
plot(qpsk_signal,'*');
hold on;
t=0:0.01:2*pi;
plot(exp(j*t),'g-.');
xlabel('I');
ylabel('Q');
title('Bieu do chom sao QPSK');
```

Figure :



Bài 6.2

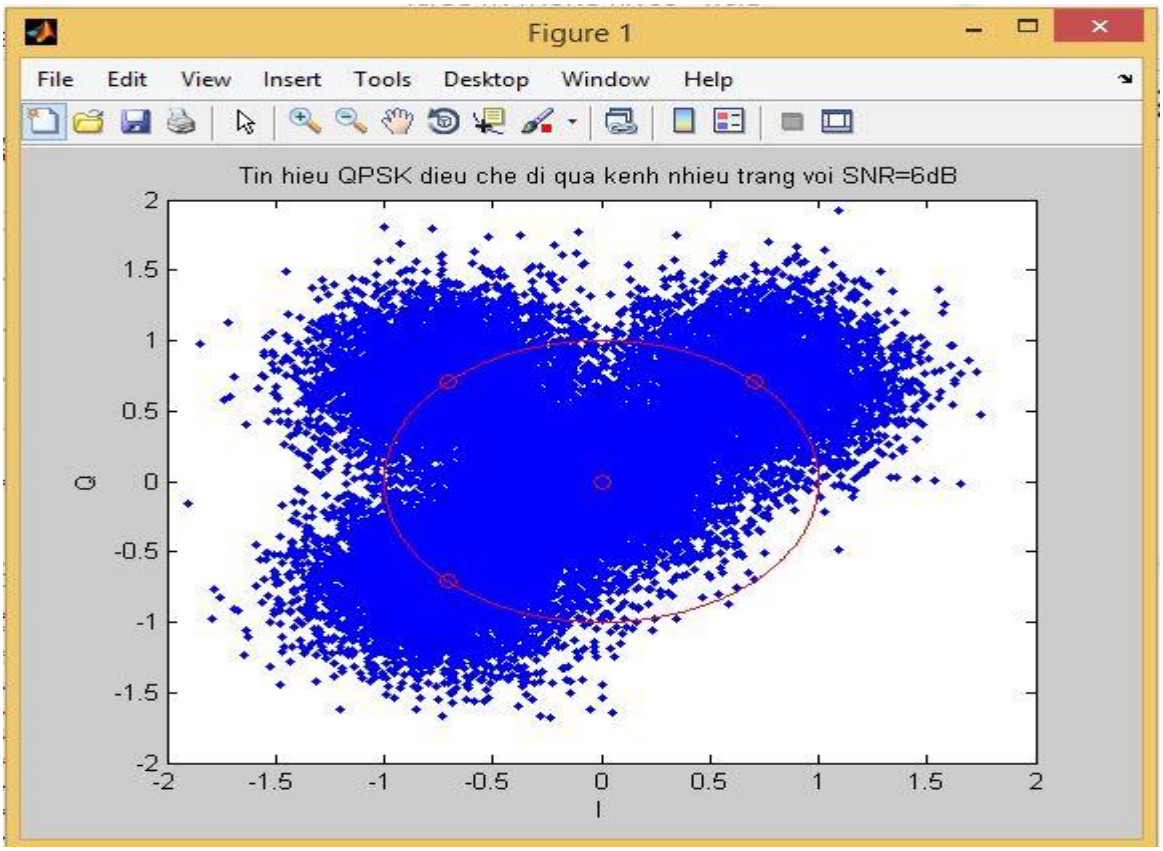
Code :

```
len =50000;     % do dai dong bit mo phong
bsignal= randint(1,len);   % tao dong bit ngau nhien co do dai len
qpsk_signal=[];
for i=1:2:length(bsignal)
   if bsignal(i)==0 & bsignal(i+1)==0      % anh xa tin hieu 00 thanh -1+j
      qpsk_signal((i+1)/2)=exp(j*3*pi/4);
   elseif bsignal(i)==0 & bsignal(i+1)==1   % anh xa tin hieu 01 thanh-1-j
      qpsk_signal((i+1)/2)=exp(j*5*pi/4);
   elseif bsignal(i==1) & bsignal(i+1)==1    % anh xa tin hieu 11 thanh 1+j
      qpsk_signal((i+1)/2)=exp(j*7*pi/4);
   elseif bsignal(i)==1 & bsignal(i+1)==0    % anh xa tin hieu 10 thanh 1+j
      qpsk_signal((i+1)/2)=exp(j*pi/4);
   end

hold off;
```

```
end
Es= std(qpsk_signal).^2;  % tinh cong suat ki hieu = phuong sai cua tin hieu QPSK
SNR_db=6; % ti le tin hieu tren nhieu 6 dB
SNR=10^(SNR_db/10);   %quy doi tu dB sang lan
N0 =Es/SNR;   % cong suat tap am
noise = sqrt(N0/2)*(randn(size(qpsk_signal))+j*randn(size(qpsk_signal)));  %
%tao kenh nhieu trang voi SNR tuong ung
output_signal = qpsk_signal+noise;  % dau ra cua tin hieu QPSK sau khi di qua
kenh nhieu trang
plot(output_signal,'.');  % ve tin hieu ra tren do thi
hold on;
plot(qpsk_signal,'ro');    % ve tin hieu dieu che len do thi
t=0:0.01:2*pi;
plot(exp(j*t),'r-');
title('Tin hieu QPSK dieu che di qua kenh nhieu trang voi SNR=6dB');
xlabel('I');
ylabel('Q');
```

Figure :



Bài 7 : Mô phỏng điều chế QDSK qua kênh nhiễu GAUSS

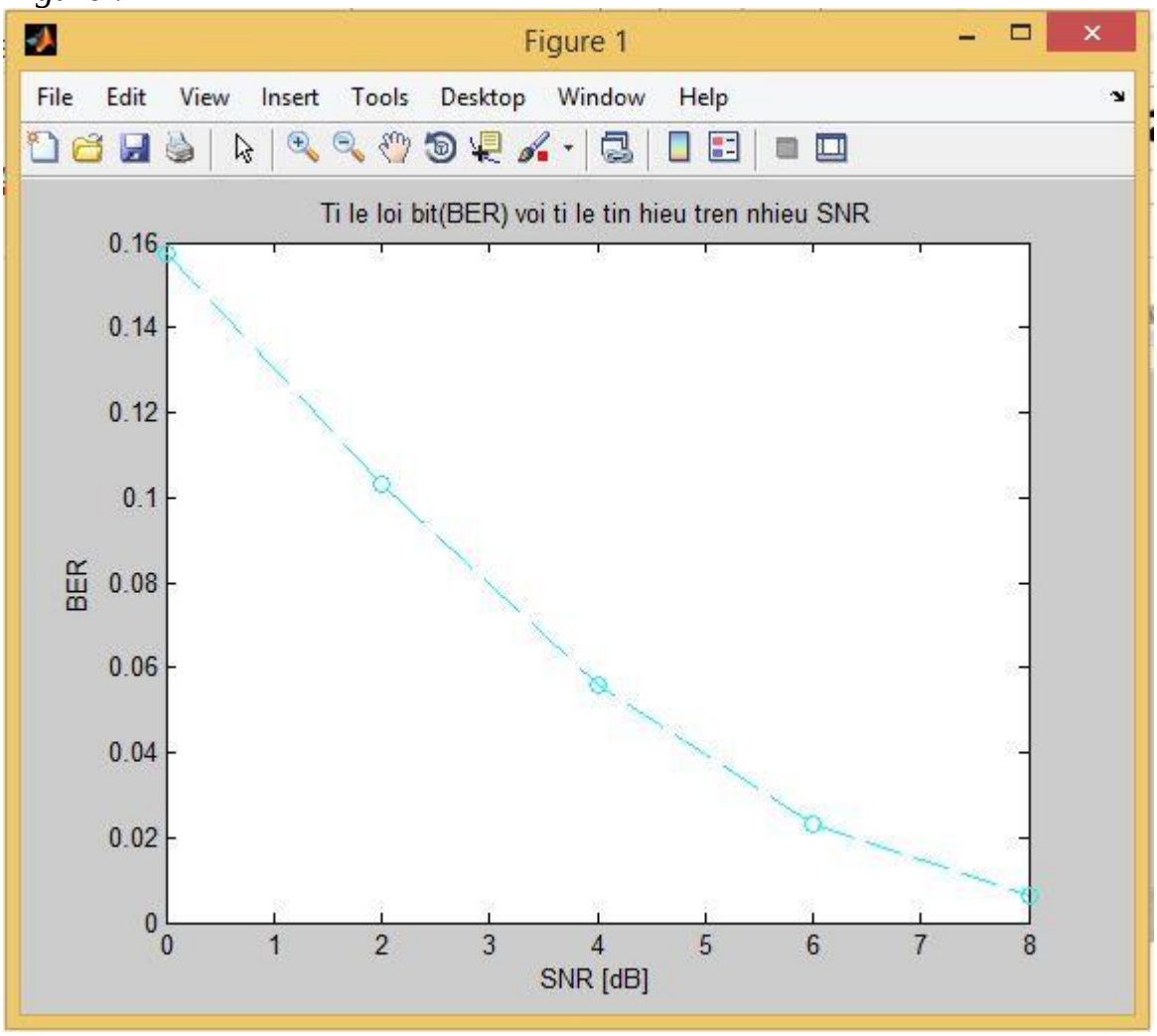
Bài 7.1

Code :

```
len = 100000;        %do dai dong bit mo phong
bsignal = randint(1,len);   %tao dong bit ngau nhien do dai len
qpsk_signal = [];
for i = 1:2:length(bsignal)
if bsignal(i)==0&bsignal(i+1)==0     %anh xa tin hieu 00 thanh -1+j
       qpsk_signal((i+1)/2) = exp(j*3*pi/4);
elseif bsignal(i)==0&bsignal(i+1)==1 %anh xa tin hieu 01 thanh -1-j
       qpsk_signal((i+1)/2) = exp(j*5*pi/4);
elseif bsignal(i)==1&bsignal(i+1)==1 %anh xa tin hieu 11 thanh 1-j
       qpsk_signal((i+1)/2) = exp(j*7*pi/4);
elseif bsignal(i)==1&bsignal(i+1)==0 %anh xa tin hieu 10 thanh 1+j
       qpsk_signal((i+1)/2) = exp(j*pi/4);
end
end
Es = std(qpsk_signal).^2;  %tinh cong suat ky hieu = phuong sai cua tin hieu
QPSK
SNR_db = 0:2:8;    %ty le tin hieu tren nhieu
SNR = 10.^(SNR_db/10);   %quy doi tu dB sang lan
N0 = Es./SNR;   %cong suat tap am
for i = 1:length(SNR_db)
noise=sqrt(N0(i)/2)*(randn(size(qpsk_signal)) + j*randn(size(qpsk_signal)));
%tao kenh nhieu trang voi SNR tuong ung
output_signal = qpsk_signal + noise;
%dau ra cua tin hieu QPSK sau khi di qua kenh nhieu trang
   demodulated_signal = []; %tao vecto rong tin hieu giai dieu che
   a = [1 0 0 0 0 1 1 1];   %tao vecto cac bit them vao tin hieu %giai dieu che
%qua trinh giai dieu che theo phuong phap xac suat cuc dai
for p = 1:len/2
       d(1)=(real(output_signal(p))-real(exp(j*pi*1/4)))^2 +
(imag(output_signal(p))-imag(exp(j*pi*1/4)))^2;
       m=1;
for k = 2:4;
```

```
        d(k) = (real(output_signal(p))-real(exp(j*pi*(2*k-1)/4)))^2 +
(imag(output_signal(p))-imag(exp(j*pi*(2*k-1)/4)))^2;
if d(k) <= d(m)
            m=k;
end
end
      demodulated_signal = [demodulated_signal a(2*m-1) a(2*m)];
end
   Pe(i) = sum(xor(bsignal,demodulated_signal))/len;  %Ty le loi bit %BER
end
plot(SNR_db,Pe,'co--');   %ve do thi mo phong
title('Ti le loi bit(BER) voi ti le tin hieu tren nhieu SNR');
xlabel('SNR [dB]');
ylabel('BER');
```

Figure :



Bài 8 : Xác suất lỗi bit trong điều chế QPSK

Code :

```
len = 100000;              %do dai dong bit mo phong
bsignal = randint(1,len);   %tao dong bit ngau nhien do dai len
qpsk_signal = [];
for i = 1:2:length(bsignal)
if bsignal(i)==0&bsignal(i+1)==0     %anh xa tin hieu 00 thanh -1+j
       qpsk_signal((i+1)/2) = exp(j*3*pi/4);
elseif bsignal(i)==0&bsignal(i+1)==1 %anh xa tin hieu 01 thanh -1-j
       qpsk_signal((i+1)/2) = exp(j*5*pi/4);
elseif bsignal(i)==1&bsignal(i+1)==1 %anh xa tin hieu 11 thanh 1-j
       qpsk_signal((i+1)/2) = exp(j*7*pi/4);
elseif bsignal(i)==1&bsignal(i+1)==0 %anh xa tin hieu 10 thanh 1+j
       qpsk_signal((i+1)/2) = exp(j*pi/4);
end
end
Es = std(qpsk_signal).^2;  %tinh cong suat ky hieu = phuong sai cua tin hieu
QPSK
SNR_db = 0:2:8;    %ty le tin hieu tren nhieu
SNR = 10.^(SNR_db/10);   %quy doi tu dB sang lan
N0 = Es./SNR;   %cong suat tap am
%thuc hieu truyen tin hieu tren kenh nhieu voi SNR tu 0 den 8 dB va giai
%dieu che, sau do tinh ty le loi bit BER
for i = 1:length(SNR_db)
noise=sqrt(N0(i)/2)*(randn(size(qpsk_signal)) + j*randn(size(qpsk_signal)));
%tao kenh nhieu trang voi SNR tuong ung
output_signal = qpsk_signal + noise;
%dau ra cua tin hieu QPSK sau khi di qua kenh nhieu trang
   demodulated_signal = []; %Tao vecto rong tin hieu giai dieu che
   a = [1 0 0 0 0 1 1 1];   %Tao vecto cac bit them vao tin hieu giai dieu che
%qua trinh giai dieu che theo phuong phap xac suat cuc dai
for p = 1:len/2
       d(1)=(real(output_signal(p))-real(exp(j*pi*1/4)))^2 +
(imag(output_signal(p))-imag(exp(j*pi*1/4)))^2;
       m=1;
for k = 2:4;
          d(k) = (real(output_signal(p))-real(exp(j*pi*(2*k-1)/4)))^2 +
(imag(output_signal(p))-imag(exp(j*pi*(2*k-1)/4)))^2;
```

```
if d(k) <= d(m)
            m=k;
end
end
      demodulated_signal = [demodulated_signal a(2*m-1) a(2*m)];
end
   Pe(i) = sum(xor(bsignal,demodulated_signal))/len;  %Ty le loi bit BER
end
plot(SNR_db,Pe,'ko--');   %ve do thi mo phong
title('Ti le loi bit(BER) voi ti le tin hieu tren nhieu SNR');
xlabel('SNR [dB]');
ylabel('BER');
hold on;
Pb = (erfc(sqrt(SNR./2)))./2;
plot(SNR_db,Pb,'rx:');
legend('Mo phong','Ly thuyet');
hold off;
```

Figure :