**Họ và tên sinh viên : Nguyễn Như Đồng**

**MSSV : 20146191**

**Lớp : CN-Điện tử 1**

**Nhóm : N14**

**PHẦN 2: CÁC BÀI THÍ NGHIỆM**

**BÀI SỐ 1: QUÁ TRÌNH NGẪU NHIÊN CỦA TÍN HIỆU**

**Bài 1.1**

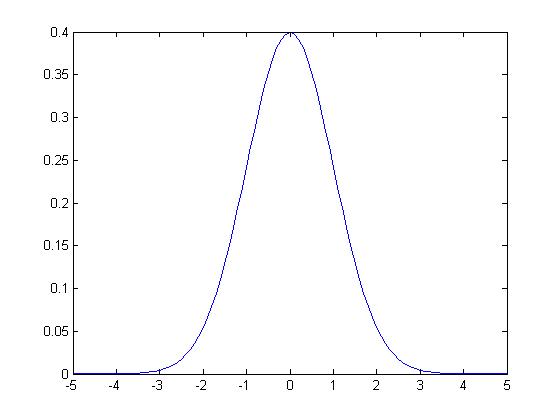
Chạy chương trình:

x = -5:0.1:5;

Px = exp(-x.^2/2)/sqrt(2\*pi);

plot(x,Px);

Ta có kết quả:



**Bài 1.2**

Chạy chương trình:

len = 100000;

x = randn(1,len);

step = .1;

k = -5:step:5;

px = hist(x,k)/len/step;

stem(k,px,'k.-');

Px\_lythuyet = exp(-k.^2/2)/sqrt(2\*pi);

hold on;

plot(k,Px\_lythuyet,'b');

title(' Phan bo xac suat Gauss ');

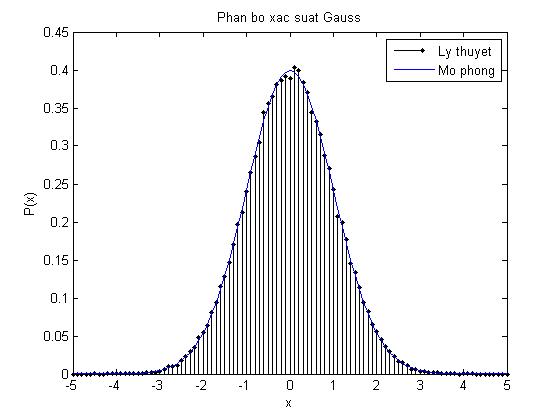
xlabel('x');

ylabel('P(x)');

legend('Ly thuyet','Mo phong');

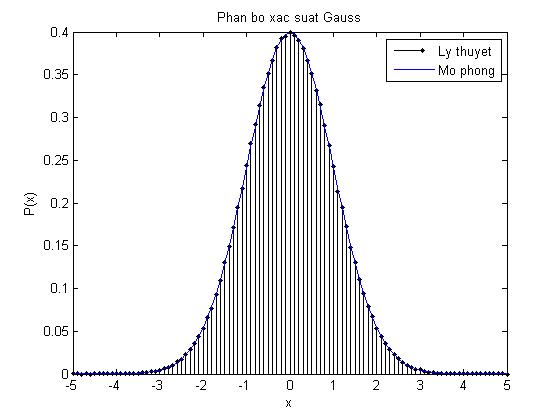
hold off;

Ta có kết quả:



Với : len = 1.000.000

Ta có kết quả:



**BÀI SỐ 2: LƯỢNG TỬ HÓA TUYẾN TÍNH**

**Lý thuyết:**

Lưu hàm sau với tên lquan.m

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);

Tại cửa sổ Command Window:

Gõ:

>> a = [-0.2 1.9 3.2 -2.5]

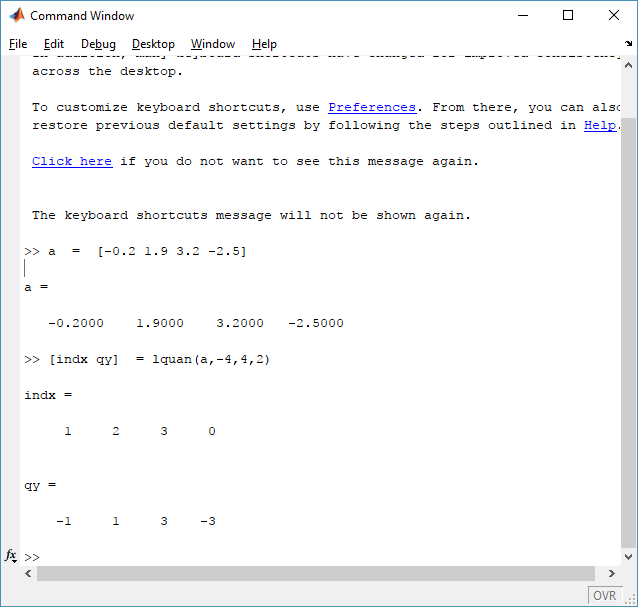
Và ấn Enter

Gõ:

>> [indx qy] = lquan(a,-4,4,2)

Và ấn Enter

Kết quả :



**Bài 2.1**

Tại cửa sổ Command Window:

Gõ:

>> xs=rand(1,5)\*2-1

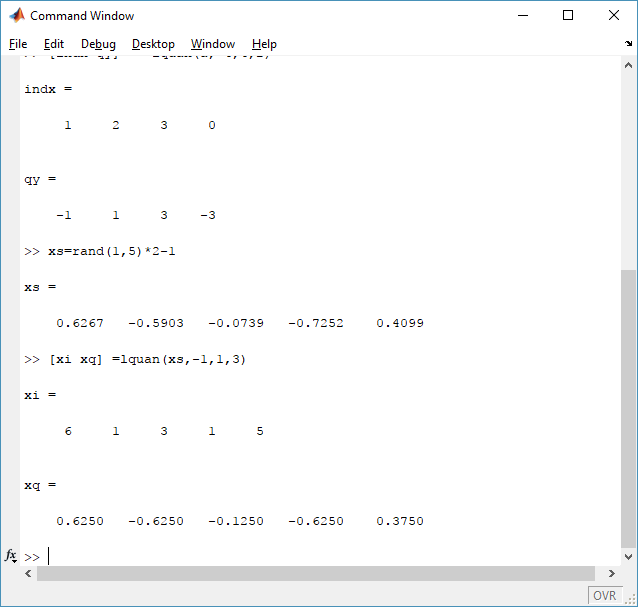
Và ấn Enter

Gõ:

>> [xi xq] =lquan(xs,-1,1,3)

Và ấn Enter

Kết quả :



**Bài 2.2**

Chạy chương trình:

t = 0:0.01:20;

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

[inx xqt] = lquan(xt,-1,1,randi(3)+1);

plot(t,xt,'b',t,xqt,'r');

grid on;

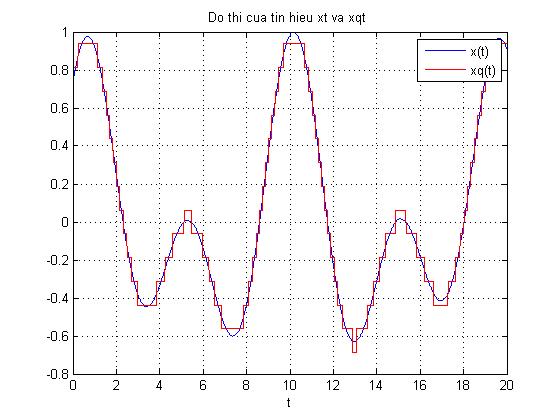
title('Do thi cua tin hieu xt va xqt ');

xlabel('t');

legend('x(t)','xq(t)');

hold off;

Ta có kết quả:



**Q14**: Số bit dùng để lượng tử hóa cho một mẫu tín hiệu là n = 4 (vì có 14 mức lượng tử hóa, 2^3 < 14 < 2^4)

**Q15**: Bước lượng tử q = 0.125

**Q16**: Liệt kê biên độ của tất cả các mức lượng tử:

Từ cao xuống thấp ta có:

0.9375; 0.8125; 0.6875; 0.5625; 0.4375; 0.3125; 0.1875; 0.0625; -0.0625; -0.1875; -0.3125; -0.4375; -0.5625; -0.6875

**BÀI SỐ 3: TẠP ÂM LƯỢNG TỬ TRONG KỸ THUẬT**

**LƯỢNG TỬ HÓA TUYẾN TÍNH**

**Bài 3:**

Chạy chương trình:

N = 1000;

x\_uni = 2\*rand(1,N)-1; % x\_uni phan bo deu tu -1 den 1

x\_sin = sin(linspace(1,5,N)); % tin hieu sin

nbit = 1:10; % so bit luong tu tu 1 den 10

SNqR\_uni = zeros(size(nbit)); % Khoi tao mang SNqR\_uni va SNqR\_sin chua

SNqR\_sin = zeros(size(nbit)); % SNqR cua tin hieu phan bo deu va tin hieu sin

SNqR\_lt = 6.02\*nbit; % Mang chua SNqR tinh theo ly thuyet

Ps\_uni = sum(x\_uni.^2)/N; % Cong suat tin hieu x theo (3-3)

Ps\_sin = sum(x\_sin.^2)/N;

for i = 1:size(nbit,2) % size(n,2) tra ve so cot cua n

[indx\_uni xq\_uni] = lquan(x\_uni,-1,1,nbit(i)); % Luong tu hoa tin hieu x\_uni

[indx\_sin xq\_sin] = lquan(x\_sin,-1,1,nbit(i)); % Luong tu hoa tin hieu x\_sin

eq\_uni = x\_uni - xq\_uni; % tinh sai so luong tu hoa x\_uni

eq\_sin = x\_sin - xq\_sin; % tinh sai so luong tu hoa x\_sin

Pq\_uni = sum(eq\_uni.^2)/N; % tinh cong suat tap am luong tu x\_uni

Pq\_sin = sum(eq\_sin.^2)/N; % tinh cong suat tap am luong tu x\_sin

SNqR\_uni(i) = 10\*log10(Ps\_uni/Pq\_uni); % tinh SNR\_db cua x\_uni

SNqR\_sin(i) = 10\*log10(Ps\_sin/Pq\_sin); % tinh SNR\_db cua x\_sin

end

plot(nbit,SNqR\_uni,'c'); % ve do thi SNR tin hieu phan bo deu mo phong

hold on;

plot(nbit,SNqR\_sin,'b'); % ve do thi SNR tin hieu sin mo phong

plot(nbit,SNqR\_lt,'r--'); % Ve do thi SNR tin hieu phan bo deu ly thuyet

title('Do thi SN\_qR theo nbit');

xlabel('nbit');

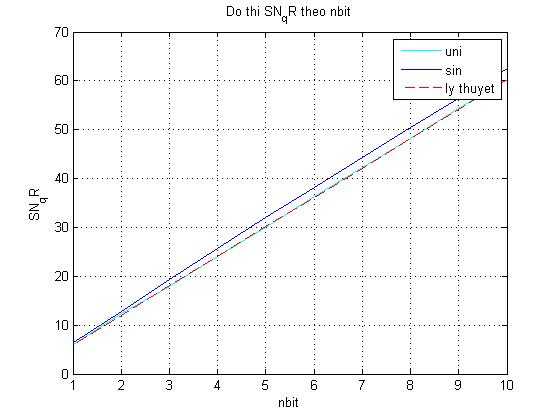
ylabel('SN\_qR');

legend('uni','sin','ly thuyet');

grid on;

hold off;

Ta có kết quả:



Trong Workspace

SNqR lý thuyết: SNqR\_lt

[6.02000000000000; 12.0400000000000; 18.0600000000000;24.0800000000000; 30.1000000000000; 36.1200000000000; 42.1400000000000; 48.1600000000000; 54.1800000000000; 60.2000000000000]

SNqR mô phỏng phân bố đều: SNqR\_uni

[6.03728139554425; 12.3422822918711; 17.9599025158900; 24.1060157366843; 30.0256506759976; 36.2676061973195; 42.2526221609806 48.2100213773512; 54.2967703637297; 60.3731665423741]

SNqR mô phỏng sin: SNqR\_sin

[6.41712349788038; 12.8834079154256; 19.3981074413044; 25.6701977872994; 32.0389037082399; 38.1619934571255; 44.3265570727482; 50.4033153693498; 56.4544016641732; 62.5722480149807]

**BÀI SỐ 4: MẬT ĐỘ PHỔ NĂNG LƯỢNG VÀ**

**HÀM TỰ TƯƠNG QUAN CỦA TÍN HIỆU**

**Bài 4.1**

Chạy chương trình:

L = 500; % Chieu dai tin hieu

x = randn(1,L); % Tao tin hieu ngau nhien

acorr\_x = xcorr(x); % Tinh ham tu tuong quan

n = (-L+1):(L-1); % Cac mau gia tri

plot(n,acorr\_x); % Ve do thi ham tu tuong quan tin hieu ngau nhien

title('Ham tu tuong quan');

xlabel('n');

ylabel('r\_x\_x');

hold on;

x = linspace(-1,1,L); % Tao tin hieu co bien do tang dan

acorr\_x = xcorr(x); % Tinh ham tu tuong quan

plot(n,acorr\_x,'k'); % Ve do thi ham tu tuong quan tin hieu co bien do tang dan

x = sin(linspace(-10,10,L)); % Tao tin hieu hinh sin

acorr\_x = xcorr(x); % Tinh ham tu tuong quan

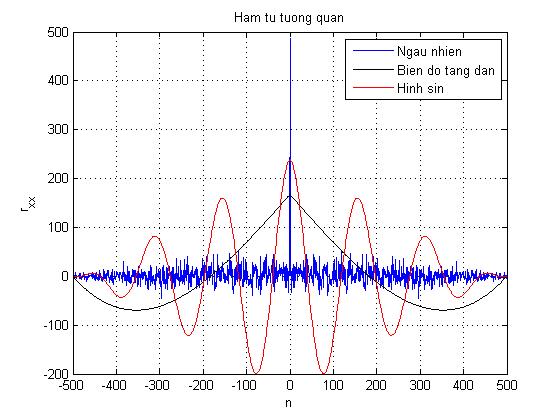
plot(n,acorr\_x,'r'); % Ve do thi ham tu tuong quan tin hieu hinh sin

legend('Ngau nhien','Bien do tang dan','Hinh sin')

hold off;

grid on;

Ta có kết quả:



**Bài 4.2**

Chạy chương trình:

L = 50; % Do dai tin hieu

N = 200; % So luong cac tan so roi rac trong khoang 0 den 2\*pi

x = rand(1,L); % Tao tin hieu ngau nhien

w = linspace(0,2\*pi,N); % Tao N tan so tang dan tu 0 den 2\*pi

fx = freqz(x,1,w); % Bien doi Fourier cua x tai cac tan so roi rac

esd\_x = fx.\*conj(fx); % Tinh ham mat do pho nang luong

acorr\_x = xcorr(x); % Tinh ham tu tuong quan cua tin hieu x

ft\_acorr\_x = freqz(acorr\_x,1,w).\*exp(j\*w\*(L-1)); % Bien doi Fourier cua ham tu tuong quan cua tin hieu x

% Ve do thi

subplot(2,1,1);

semilogy(w/pi,esd\_x);

title('Mat do pho nang luong');

xlabel('n');

ylabel('S(e^j^\omega)')

hold off;grid on;

subplot(2,1,2);

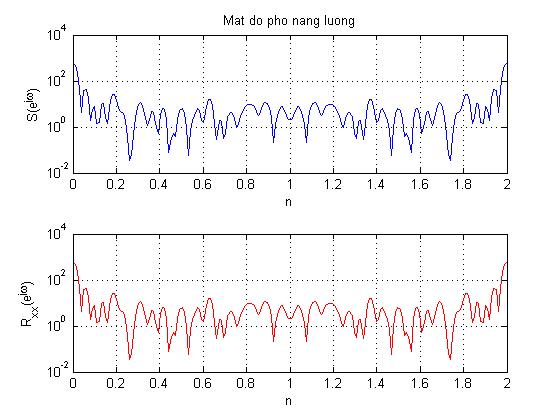
semilogy(w/pi,real(ft\_acorr\_x),'r');

xlabel('n');

ylabel('R\_x\_x(e^j^\omega)');

hold off;grid on;

Ta có kết quả:



**BÀI SỐ 5: MÃ ĐƯỜNG DÂY NRZ**

**Bài 5.1**

Chạy chương trình:

len = 100000; % Do dai dong bit mo phong

SNR\_db = 0:2:8; % Tao vector SNR\_db = 0 2 4 6 8

SNR = 10.^(SNR\_db/10); % Doi SNR tu Decibel sang lan

bsignal = randi([0 1],1,len); % Dong bit ngau nhiên do dai len

NRZ\_signal = bsignal\*2-1; % Bien doi dòng bit 0 1 sang -1 1

N0 = 1./SNR; % Cong suat tap am

for i=1:length(SNR\_db)

noise = sqrt(N0(i))\*randn(1,len); % Tao tap am noise voi ti so SNR(i)

r\_signal = NRZ\_signal + noise; % Tin hieu thu duoc = NRZ + noise

NRZ\_decoded = sign(r\_signal); % Giai ma tin hieu NRZ thu duoc

[n,BER(i)] = symerr(NRZ\_decoded,NRZ\_signal); % Tinh xac suat loi

end

plot(SNR\_db,BER,'bo--'); % Ve do thi

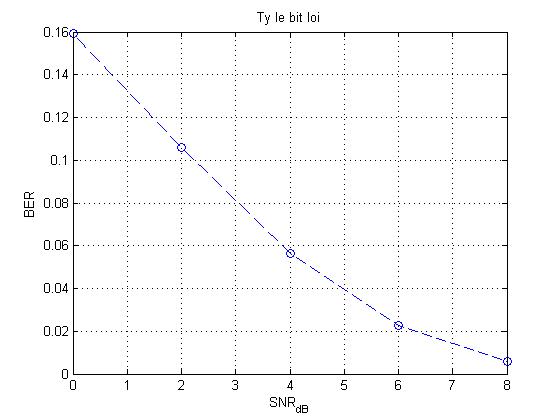
title('Ty le bit loi');

xlabel('SNR\_d\_B');

ylabel('BER');

grid on;

Ta có kết quả:



**Bài 5.2**

Chạy chương trình:

len = 100000; % Do dai dong bit mo phong

SNR\_db = 0:2:8; % Tao vector SNR\_db = 0 2 4 6 8

SNR = 10.^(SNR\_db/10); % Doi SNR tu Decibel sang lan

bsignal = randi([0 1],1,len); % Dong bit ngau nhiên do dai len

NRZ\_signal = bsignal\*2-1; % Bien doi dòng bit 0 1 sang -1 1

N0 = 1./SNR; % Cong suat tap am

for i=1:length(SNR\_db)

noise = sqrt(N0(i))\*randn(1,len); % Tao tap am noise voi ti so SNR(i)

r\_signal = NRZ\_signal + noise; % Tin hieu thu duoc = NRZ + noise

NRZ\_decoded = sign(r\_signal); % Giai ma tin hieu NRZ thu duoc

[n,BER(i)] = symerr(NRZ\_decoded,NRZ\_signal); % Tinh xac suat loi

end

plot(SNR\_db,BER,'r\*'); % Ve do thi BER

Pe = 1/2\*(1-erf(sqrt(SNR)/sqrt(2))); % Xac suat loi theo ly thuyet

hold on;

plot(SNR\_db,Pe); % Ve do thi Pe

title('Do thi ty le bit loi theo ly thuyet va mo phong');

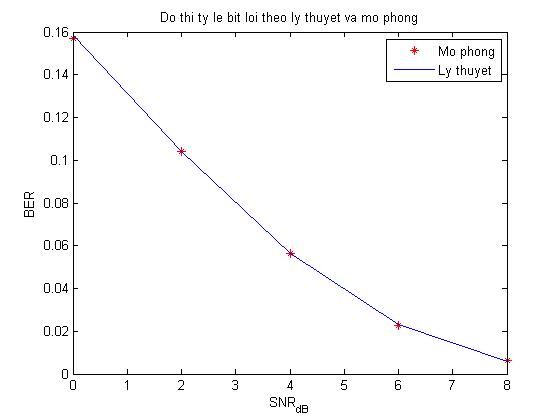
xlabel('SNR\_d\_B');

ylabel('BER');

legend('Mo phong','Ly thuyet');

hold off;

Ta có kết quả:



Trong Workspace

Pe Lý thuyết:

[0.158655253931457;0.104028637085389;0.0564953017493617; 0.0230071388778660; 0.00600438640016354]

BER mô phỏng:

[0.156960000000000;0.104060000000000;0.0564000000000000; 0.0227300000000000; 0.00612000000000000]

**BÀI SỐ 6: KỸ THUẬT ĐIỀU CHẾ SỐ QPSK**

**Bài 6.1**

Chạy chương trình:

len = 50000; % Do dai dong bit mo phong

SNR\_db = 6; % SNR co don vi Decibel

SNR = 10^(SNR\_db/10); % Doi SNR tu Decibel sang lan

bsignal = randi([0 1],1,len); % Tao dong bit ngau nhien do dai len

% Thuc hien dieu che QPSK

for i=1:2:len

if bsignal(i)==0 & bsignal(i+1)==0 % 00

qpsk\_signal((i+1)/2) = exp(j\*3\*pi/4);

elseif bsignal(i)==0 & bsignal(i+1)==1 % 01

qpsk\_signal((i+1)/2) = exp(j\*5\*pi/4);

elseif bsignal(i)==1 & bsignal(i+1)==1 % 11

qpsk\_signal((i+1)/2) = exp(j\*7\*pi/4);

elseif bsignal(i)==1 & bsignal(i+1)==0 % 10

qpsk\_signal((i+1)/2) = exp(j\*pi/4);

end

end

Es = std(qpsk\_signal)^2; % Nang luong ky hieu

N0 = Es/SNR; % Cong suat tap am

% Tao nhieu Gauss

noise = sqrt(N0/2)\*(randn(1,length(qpsk\_signal))+j\*randn(1,length(qpsk\_signal)));

qpsk\_awgn = qpsk\_signal + noise; % Cho tin hieu dieu che di qua kenh AWGN

plot(qpsk\_awgn,'.'); % Ve bieu do chom sao tin hieu co nhieu

title('Do thi chom sao khong nhieu va co nhieu voi SNR=6dB');

xlabel('I');

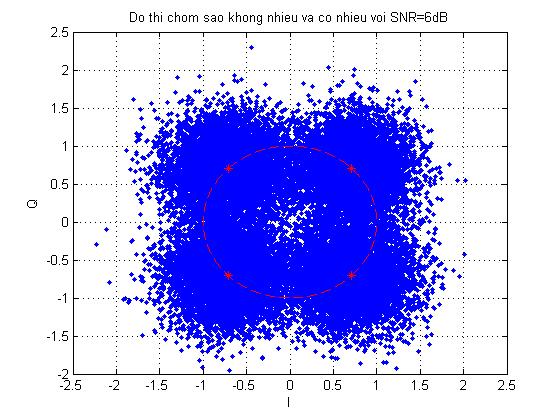
ylabel('Q');

hold on;

plot(qpsk\_signal,'r\*'); % Ve bieu do chom sao tin hieu khong nhieu

plot(exp(j\*[0:0.01:2\*pi]),'r--');

Ta có kết quả:



**Bài 6.2**

Chạy chương trình:

len = 50000; % Do dai dong bit mo phong

SNR\_db = 3; % SNR co don vi Decibel

SNR = 10^(SNR\_db/10); % Doi SNR tu Decibel sang lan

bsignal = randi([0 1],1,len); % Tao dong bit ngau nhien do dai len

% Thuc hien dieu che QPSK

for i=1:2:len

if bsignal(i)==0 & bsignal(i+1)==0 % 00

qpsk\_signal((i+1)/2) = exp(j\*3\*pi/4);

elseif bsignal(i)==0 & bsignal(i+1)==1 % 01

qpsk\_signal((i+1)/2) = exp(j\*5\*pi/4);

elseif bsignal(i)==1 & bsignal(i+1)==1 % 11

qpsk\_signal((i+1)/2) = exp(j\*7\*pi/4);

elseif bsignal(i)==1 & bsignal(i+1)==0 % 10

qpsk\_signal((i+1)/2) = exp(j\*pi/4);

end

end

Es = std(qpsk\_signal)^2; % Nang luong ky hieu

N0 = Es/SNR; % Cong suat tap am

% Tao nhieu Gauss

noise = sqrt(N0/2)\*(randn(1,length(qpsk\_signal))+j\*randn(1,length(qpsk\_signal)));

qpsk\_awgn = qpsk\_signal + noise; % Cho tin hieu dieu che di qua kenh AWGN

plot(qpsk\_awgn,'.'); % Ve bieu do chom sao tin hieu co nhieu

title('Do thi chom sao khong nhieu va co nhieu voi SNR=3dB');

xlabel('I');

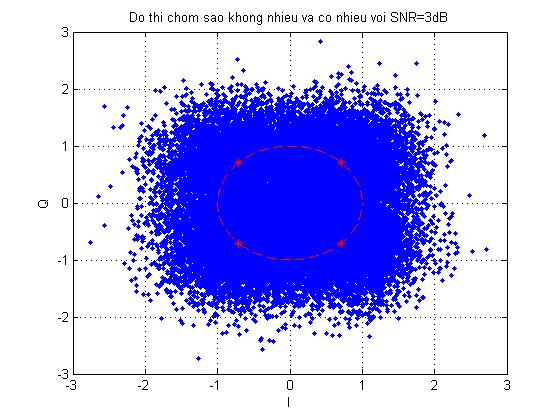
ylabel('Q');

hold on;

plot(qpsk\_signal,'r\*'); % Ve bieu do chom sao tin hieu khong nhieu

plot(exp(j\*[0:0.01:2\*pi]),'r--');

Ta có kết quả:



**Bài 6.3**

Chạy chương trình:

len = 50000; % Do dai dong bit mo phong

SNR\_db = 0; % SNR co don vi Decibel

SNR = 10^(SNR\_db/10); % Doi SNR tu Decibel sang lan

bsignal = randi([0 1],1,len); % Tao dong bit ngau nhien do dai len

% Thuc hien dieu che QPSK

for i=1:2:len

if bsignal(i)==0 & bsignal(i+1)==0 % 00

qpsk\_signal((i+1)/2) = exp(j\*3\*pi/4);

elseif bsignal(i)==0 & bsignal(i+1)==1 % 01

qpsk\_signal((i+1)/2) = exp(j\*5\*pi/4);

elseif bsignal(i)==1 & bsignal(i+1)==1 % 11

qpsk\_signal((i+1)/2) = exp(j\*7\*pi/4);

elseif bsignal(i)==1 & bsignal(i+1)==0 % 10

qpsk\_signal((i+1)/2) = exp(j\*pi/4);

end

end

Es = std(qpsk\_signal)^2; % Nang luong ky hieu

N0 = Es/SNR; % Cong suat tap am

% Tao nhieu Gauss

noise = sqrt(N0/2)\*(randn(1,length(qpsk\_signal))+j\*randn(1,length(qpsk\_signal)));

qpsk\_awgn = qpsk\_signal + noise; % Cho tin hieu dieu che di qua kenh AWGN

plot(qpsk\_awgn,'.'); % Ve bieu do chom sao tin hieu co nhieu

title('Do thi chom sao khong nhieu va co nhieu voi SNR=0dB');

xlabel('I');

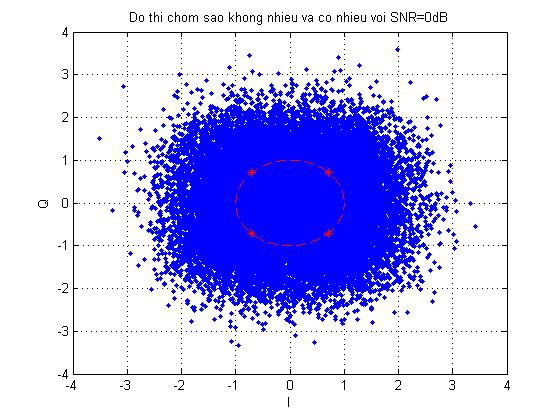
ylabel('Q');

hold on;

plot(qpsk\_signal,'r\*'); % Ve bieu do chom sao tin hieu khong nhieu

plot(exp(j\*[0:0.01:2\*pi]),'r--');

Ta có kết quả:



**BÀI SỐ 7: XÁC SUẤT LỖI BIT TRONG ĐIỀU CHẾ QPSK**

Chạy chương trình

len = 50000; % Do dai dong bit mo phong

SNR\_db = 0:2:8; % Tao vector SNR\_db = 0 2 4 6 8

SNR = 10.^(SNR\_db/10); % Doi SNR tu Decibel sang lan

bsignal = randi([0 1],1,len); % Tao dong bit ngau nhien do dai len

% Thuc hien dieu che QPSK

for i=1:2:len

if bsignal(i)==0 & bsignal(i+1)==0 % 00

qpsk\_signal((i+1)/2) = exp(j\*3\*pi/4);

elseif bsignal(i)==0 & bsignal(i+1)==1 % 01

qpsk\_signal((i+1)/2) = exp(j\*5\*pi/4);

elseif bsignal(i)==1 & bsignal(i+1)==1 % 11

qpsk\_signal((i+1)/2) = exp(j\*7\*pi/4);

elseif bsignal(i)==1 & bsignal(i+1)==0 % 10

qpsk\_signal((i+1)/2) = exp(j\*pi/4);

end

end

% Tim BER mo phong

for i=1:length(SNR\_db)

r\_signal = awgn(qpsk\_signal,SNR\_db(i)); % Dieu che QPSK di qua nhieu AWGN

for j=1:2:len % Giai dieu che tin hieu QPSK co nhieu

if real(r\_signal((j+1)/2))>=0

if imag(r\_signal((j+1)/2))>=0 % Goc phan tu I

r\_bsignal(j) = 1;

r\_bsignal(j+1) = 0;

else % Goc phan tu IV

r\_bsignal(j) = 1;

r\_bsignal(j+1) = 1;

end

else

if imag(r\_signal((j+1)/2))>=0 % Goc phan tu II

r\_bsignal(j) = 0;

r\_bsignal(j+1) = 0;

else % Goc phan tu III

r\_bsignal(j) = 0;

r\_bsignal(j+1) = 1;

end

end

end

[n,BER(i)] = biterr(r\_bsignal,bsignal);

end

Pb = 1/2\*erfc(1/sqrt(2).\*sqrt(SNR)); % Xac suat loi bit

plot(SNR\_db,Pb,'r'); % Ve do thi Pb ly thuyet

title('Do thi ty le bit loi ly thuyet va mo phong');

xlabel('SNR\_d\_B');

ylabel('BER');

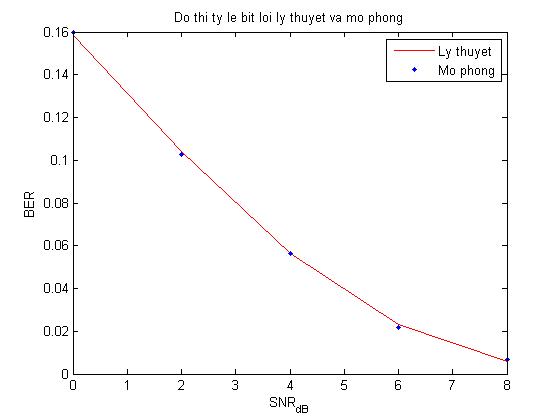
hold on;

plot(SNR\_db,BER,'b.');

legend('Ly thuyet','Mo phong');

hold off;

Ta có kết quả:



Trong Workspace

Pb lý thuyết:

[0.158655253931457;0.104028637085389;0.0564953017493617; 0.0230071388778660; 0.00600438640016357]

BER mô phỏng:

[0.159800000000000;0.102640000000000;0.0565600000000000; 0.0219400000000000; 0.00680000000000000]

**BÀI SỐ 8: MÔ PHỎNG ĐIỀU CHẾ M-QAM QUA KÊNH NHIỄU GAUSS**

Chạy chương trình:

n\_sym = 50000; % so ki tu dieu che

M = [16 64 128];

SNR\_db = 0:25;

BER = zeros(length(M),length(SNR\_db));

for k=1:size(M,2) % size(M,2) la so cot cua M

s\_stream = randi([0 M(k)-1],1,n\_sym); % tao dong bieu tuong do dai n\_sym

s\_mod = qammod(s\_stream,M(k),0,'GRAY'); % dieu che tin hieu y

for r=1:size(SNR\_db,2) % vong lap tinh BER

s\_mod\_awgn = awgn(s\_mod,SNR\_db(r),'measured'); % tin hieu y qua kenh nhieu tro thanh y\_awgn

s\_demod = qamdemod(s\_mod\_awgn,M(k),0,'GRAY'); % giai dieu che M-QAM

[number,ratio] = biterr(s\_stream,s\_demod); % tinh ti le loi bit

BER(k,r) = ratio % luu ti le loi bit vua tinh vao BER

end

end

semilogy(SNR\_db,BER(1,:),'bo-',SNR\_db,BER(2,:),'rs-',SNR\_db,BER(3,:),'m\*-');

grid on;

title('Do thi ty le loi bit dieu che M-QAM');

xlabel('SNR\_ db');

ylabel('BER');

legend('16-QAM','64-QAM','128-QAM');

Ta có kết quả:

