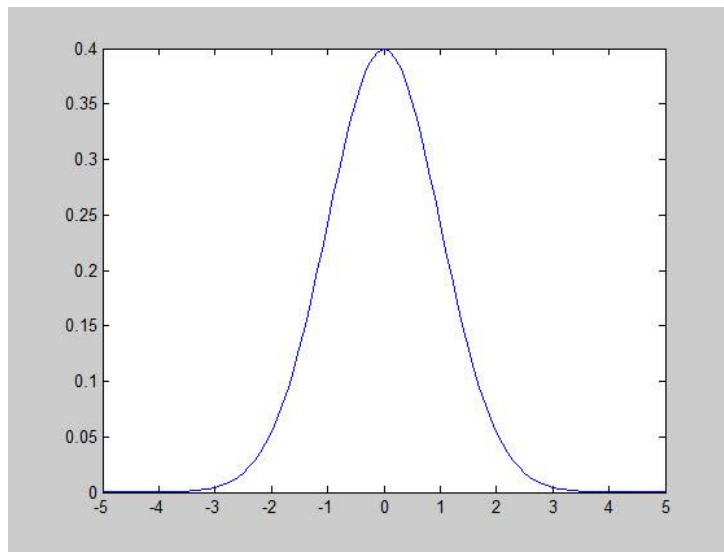


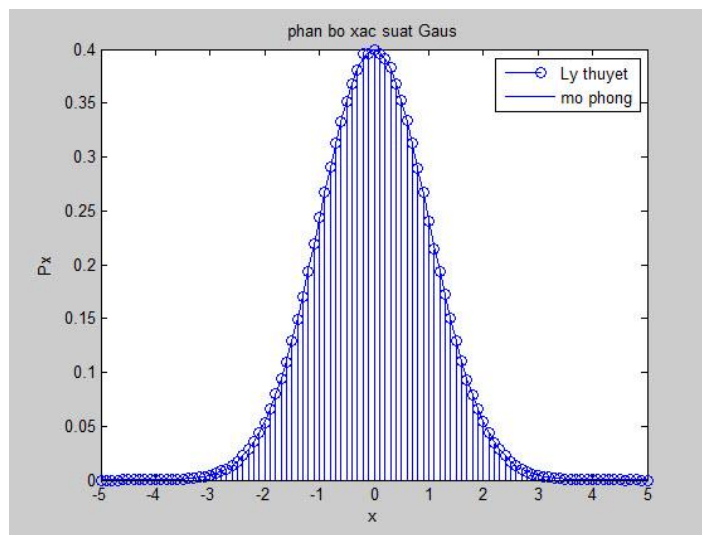
bài 1.1

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



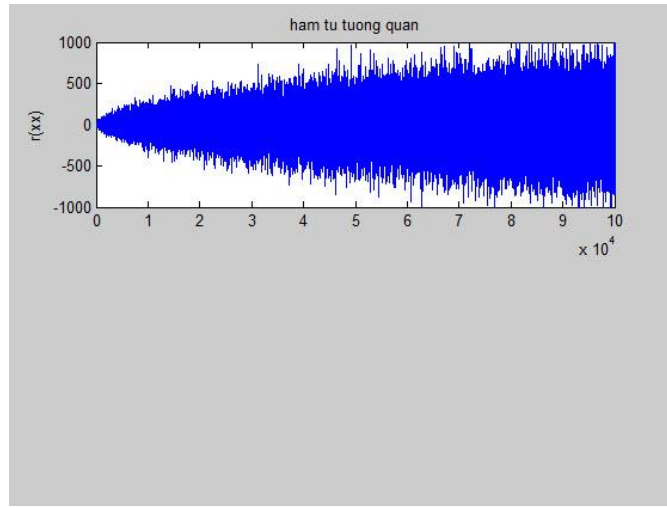
bài 1.2

```
len = 1000000;  
x = randn(1,len);  
step = 0.1;  
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 Gaus');  
xlabel('x');  
ylabel('Px');  
legend('Ly thuyet','mo phong');  
hold off;
```



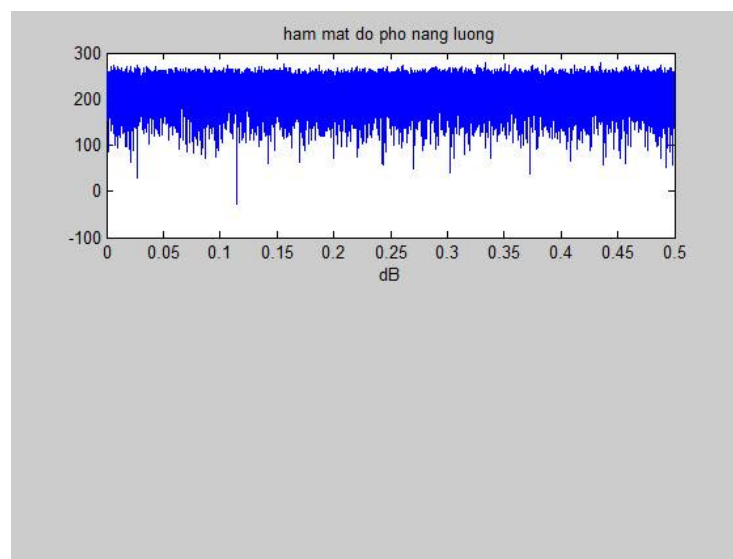
bài 2.1

```
len = 100000;  
n=1:len;  
x=randn(1,len);  
y=xcorr(x,x);  
subplot(2,1,1);  
plot(y);  
axis([0 10^5 -1000 1000]);  
save acorr_x;  
title('ham tu tuong quan');  
ylabel('r(xx)');
```



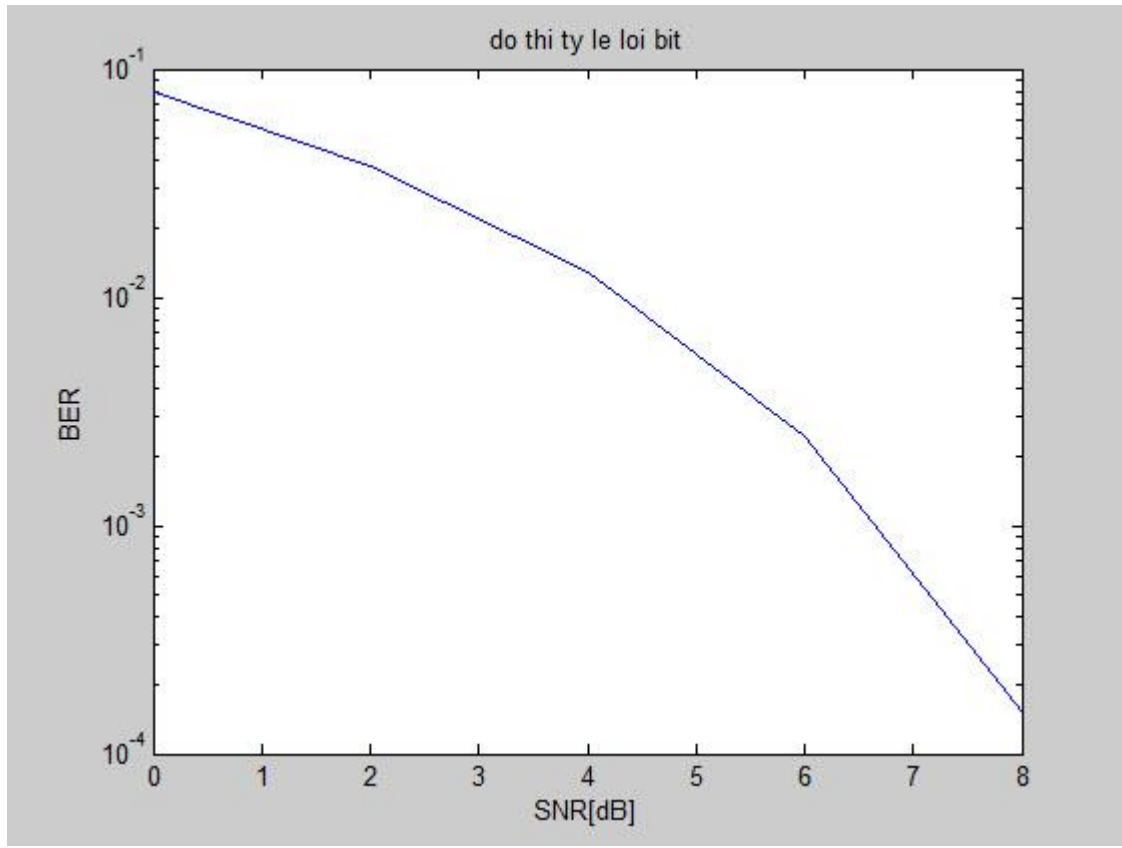
bài 2.2

```
len=100000;  
z=abs(fft(y));  
s=(0:len/2-1)/len;  
subplot(2,1,1);  
plot(s,20*log(z(1:len/2)));  
title('ham mat do pho nang luong');  
xlabel('dB');
```



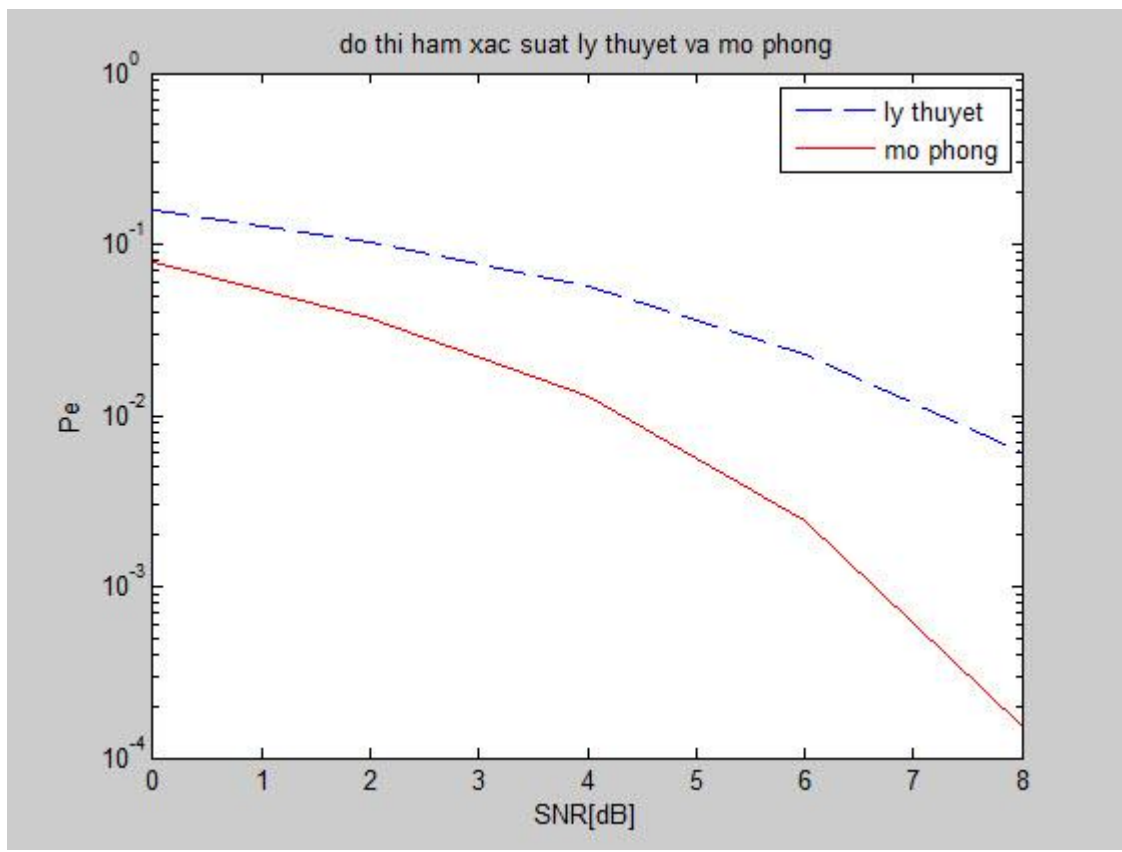
bài 3.1

```
len = 100000;
bsignal = rand(1,len)>0.5;%tạo 1 vecto len bit ngẫu nhiên
NRZ_signal = bsignal*2 -1;%điều chế BPSK
n=1/sqrt(2)*[randn(1,len)+j*randn(1,len)];%tạo 1 nhiễu phức Gauss
SNR_db = 0:2:8;
for i=1:length(SNR_db)
    y=NRZ_signal + 10^(-SNR_db(i)/20)*n;%cộng nhiễu trắng
    ur=real(y)>0;%ký hiệu thu được
    c(i)=size(find([bsignal-ur]),2);
end
BER=c/len;
Berlt=0.5*erfc(sqrt(10.^(SNR_db/10)));%xác suất lỗi bit lý thuyết
save bai31 BER;
semilogy(SNR_db,BER, 'b-');
xlabel('SNR[dB]');
ylabel('BER');
title('do thi ty le loi bit');
```



bài 3.2

```
SNR_db=0:2:8;
for i=1:length(SNR_db)
    SNR(i)=10^(SNR_db(i)/10);
    p(i)=1/2*[1-erf(1/sqrt(2)*sqrt(SNR(i)))];%công thức Pe lý thuyết
end
semilogy(SNR_db,p,'b--');
xlabel('SNR[dB]');
ylabel('Pe');
title('do thi ham xac suat ly thuyet va mo phong')
load bai31 BER
hold on;
semilogy(SNR_db,BER,'r-');
legend('ly thuyet','mo phong');
hold off;
```



bài 4.1

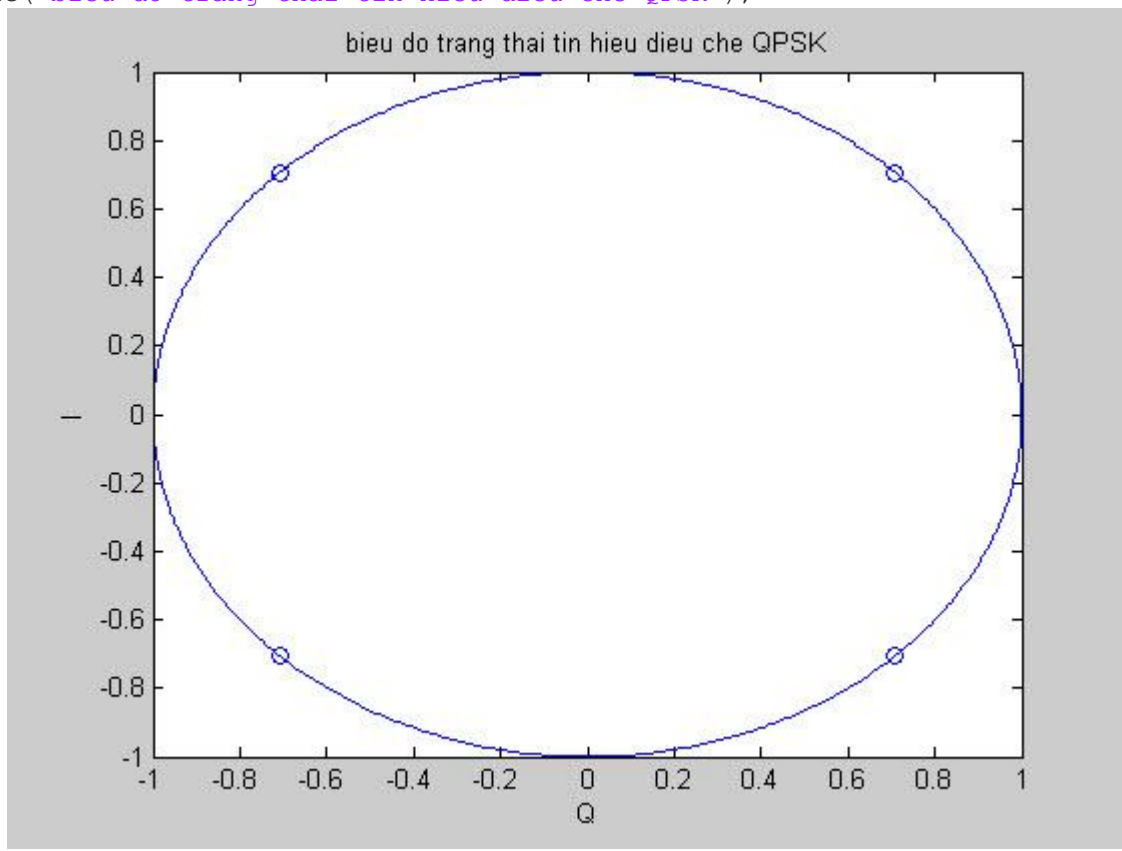
```
clear;
x=round(rand(1,10000));%tạo 1 vecto bit ngẫu nhiên
plot(x);
for i=1:2:length(x)
    if x(i)==0 & x(i+1)==0
        s((i+1)/2)=exp(j*pi/4);
    elseif x(i)==0 & x(i+1)==1
        s((i+1)/2)=exp(j*3*pi/4);
    elseif x(i)==1 & x(i+1)==1
        s((i+1)/2)=exp(j*5*pi/4);
    elseif x(i)==1 & x(i+1)==0
        s((i+1)/2)=exp(j*7*pi/4);
    end
end

save qpsk_signal s x;

plot(s,'o');
hold on;

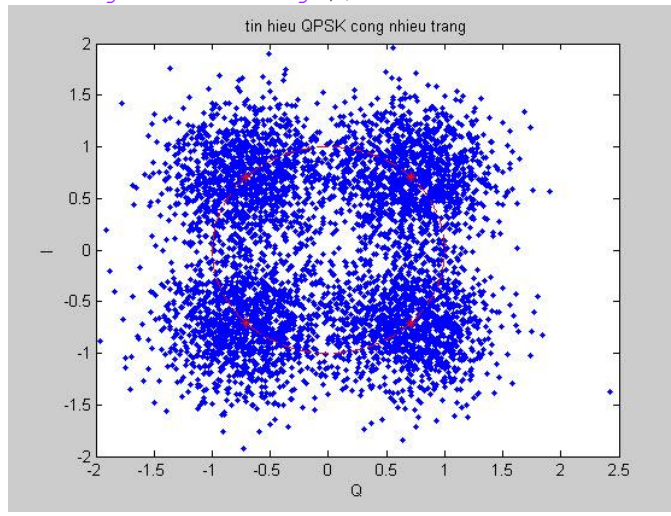
t=0:0.01:2*pi;%khai báo biến t
plot(exp(j*t),'-');

xlabel('Q');
ylabel('I');
title('biểu đồ trạng thái tín hiệu điều chế QPSK');
```

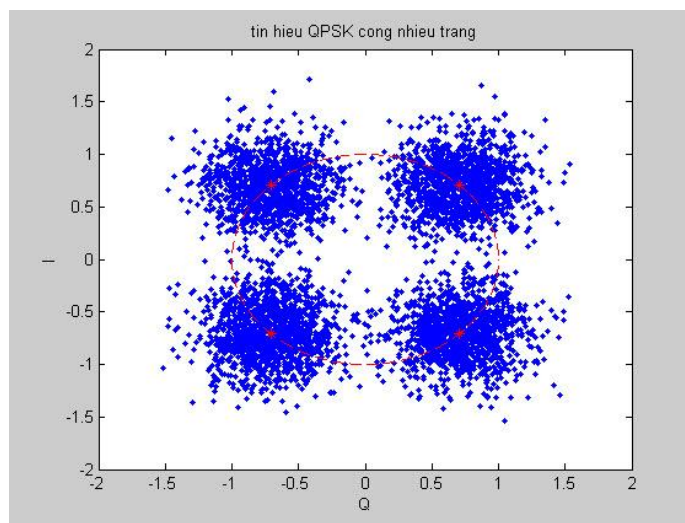


bài 4.2

```
clear;
load qpsk_signal;
es=var(s);%năng lượng của 1 symbol
eb=es/2;
snr_db=6;
n_0=eb/10^(snr_db/10);
n=sqrt(n_0/2)*(randn(size(s))+j*randn(size(s)));%nhiều trắng phức cùng chiều
dài
r=s+n;
plot(r, '.');
hold on;
plot(s, 'r*');
hold on;
t=0:0.01:2*pi;
plot(exp(j*t), 'r--');
xlabel('Q');
ylabel('I');
title('tin hieu QPSK cong nhieu trang');
```



tỷ lệ SNR = 3dB



tỷ lệ SNR=6dB

bài 5:

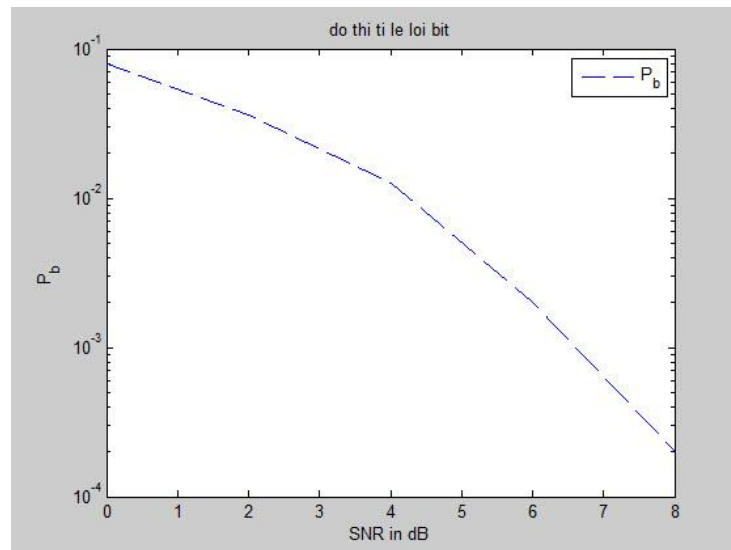
file ex5.m

```
function y=ex5(SNR_db,s,x)
es=var(s);
eb=es/2;
N_0=eb/10.^(SNR_db/10);
N0=sqrt(N_0/2)*(randn(size(s))+j*randn(size(s)));%tạo nhiễu trắng phức
ns=s+N0;
theta_m=[pi/4,3*pi/4,5*pi/4,7*pi/4];%vòng lặp so sánh độ lệch của ký hiệu thu
được với các giá trị ký hiệu chuẩn
S_m=exp(j*theta_m);
for i=1:length(s)
    d=abs(S_m-ns(i));
    md=min(abs(S_m-ns(i)));
    if md==d(1);
        r(2*i-1)=0;
        r(2*i)=0;
    elseif md==d(2);
        r(2*i-1)=0;
        r(2*i)=1;
    elseif md==d(3);
        r(2*i-1)=1;
        r(2*i)=1;
    elseif md==d(4);
        r(2*i-1)=1;
        r(2*i)=0;
    end
end
c=0;%mặc định biến đếm lỗi bit =0
for i=1:length(x)
    if r(i)~=x(i);
        c=c+1;
    end
end
y=c;
```

bài 5:

```
clear all;
load ex5pl_res s x;
snr_db=0:2:8;
for i=1:length(snr_db)
    c(i)=ex5(snr_db(i),s,x);
end

BEP=c/length(x);
semilogy(snr_db,BEP,'--');
title('đồ thị tỉ lệ lỗi bit');
xlabel('SNR in dB');
ylabel('P_b');
legend('P_b');
save bai5 c BEP;
```



đồ thị bài 5

bài 6:

```
clear;
snr_db=0:8;
snr_db_simulation=0:2:8;

for i=1:length(snr_db)
    snr(i)=10^(snr_db(i)/10);
    gamma_b(i)=snr(i);
    p_b(i)=erfc(sqrt(2*gamma_b(i))/sqrt(2))/2;%hàm lỗi bù
end
semilogy(snr_db,p_b,'ro--')
hold on
load bai5 c BEP;
semilogy(snr_db_simulation,BEP,'x--')
title('xác suất lỗi bit lý thuyết và mô phỏng');
xlabel('SNR in dB');
ylabel('P_b');
legend('lý thuyết','mô phỏng');
hold off
```

