**BÁO CÁO THÍ NGHIỆM THÔNG TIN SỐ**

Sinh viên thực hiện: Nguyễn Thế Hoạch

Lớp: ĐTVT 10 k55

Mã sinh viên : 20101590

**Bài số 1**

* 1. **Đồ thị hàm phân bố xác suất Gauss**

Code:

step = 0.1;

x= -5:step:5;

px=(1/sqrt(2\*pi))\*exp(-x.^2/2);

plot(x,px); % vẽ đồ thị

xlabel('x');%tiêu đề trục x

ylabel('P(x)');%tiêu đề trục y

title('do thi phan bo xac suat Gauss'); %tiêu đề đồ thị

Figure

* 1. **Tạo một quá**

Code

len=100000;

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 Gauss');

xlabel('x');

ylabel('y');

legend('ly thuyet', ' mo phong');

hold off;

Figure



**Bài số 2**

**2.1. Xây dựng hàm tự tương quan của quá trình ngẫu nhiên bài 1.2**

Code

x=randn(1,100); %tao mot vecto ngau nhien co 100 phan tu

[t y]=xcorr(x);

stem(y,t); %ve do thi ham tu tuong quan

title ('do thi ham tu tuong quan'); %dat ten do thi

xlabel('n'); %ten truc hoanh

ylabel('xcorr\_x'); %ten truc tung

Figure



**2.2. Hàm mật độ phổ năng lượng**

Code

x=randn(1,100); %tao mot vecto ngau nhien co 100 phan tu

y=xcorr(x); %tinh y=xcorr(x)

esd\_x=(fft(x,1000)).^2;

ft\_acorr\_x=fft(y,1000);

stem(esd\_x);

hold on;

stem(ft\_acorr\_x);

Figure



**Bài số 3**

**3.1. Mã đường dây NRZ**

Code

N = 10^6 ;

u = rand(1,N)>0.5; % tạo một vecto N bit ngẫu nhiên

s = 2\*u-1; % điều chế BPSK, u

n = 1/sqrt(2)\*[randn(1,N) + j\*randn(1,N)]; % tạo một nhiễu phức Gauss

SNR = 0:2:6; % tỉ lệ tín hiệu trên tạp âm [dB]

for i = 1:length(SNR)

y = s + 10^(-SNR(i)/20)\*n; % cộng nhiễu trắng

uHat = real(y)>0; % kí hiệu thu được

c(i) = size(find([u- uHat]),2);

end

BER = c/N;

theoryBer = 0.5\*erfc(sqrt(10.^(SNR/10))); % xác suất lỗi bit lý thuyết

save bpsk BER; % lưu file để phục vụ bài 3.2

semilogy(SNR,BER,'b.-');

xlabel('SNR[dB]');

ylabel('BER');

title('ti le BER/SNR');

Figure



**3.2 So sánh với công thức lý thuyết**

Code

SNR\_db=0:2:6;

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--');% vẽ đồ thị dB

xlabel('SNR[dB]');

ylabel('Pe');

title('Do thi so sanh ly thuyet va mo phong');% đồ thị xác suất lỗi lý thuyết

load bpsk BER% load file bpsk bài 3.1

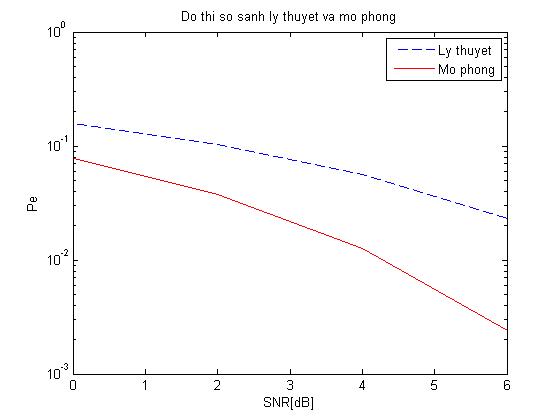
hold on;

semilogy(SNR\_db,BER,'r-'); % vẽ đồ thị với trục dB

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

hold off;

Figure



**Bài số 4**

**4.1. Biểu đồ chòm sao của tín hiệu điều chế QPSK mã hóa theo mã Gray** :

Code

x=round(rand(1,10000));% tạo một vecto bit ngẫu nhiên

for i=1:2:length(x) % vòng lặp từ 1 tới độ dài x, bước nhảy 2

switch x(i) % sử dụng switch case để lựa chọn mã hóa

case 0

if x(i+1)==0

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

else

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

end

case 1

if x(i+1)==0

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

else

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

end

end

end

save Bai4p1 s x;

plot(s,'\*');

hold on;

t=0:0.01:2\*pi; % khai báo biến t

plot(exp(j\*t),'k--'); % vẽ đường tròn đơn vị, màu đen, nét đứt

xlabel('\phi(t)');

ylabel('s\_m');

title('Bieu do khong gian trang thai tin hieu dieu che QPSK voi ma Gray');

Figure

****

**4.2 Cho tín hiệu ở trên qua kênh nhiễu trắng AWGN :**

Code

clear;

x=round(rand(1,10000));

for i=1:2:length(x) % vòng lặp từ 1 tới độ dài x, bước nhảy 2

switch x(i) % sử dụng switch case để lựa chọn mã hóa

case 0

if x(i+1)==0

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

else

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

end

case 1

if x(i+1)==0

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

else

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

end

end

end

Es=var(s); % Năng lượng của một symbol

Eb=Es/2;

SNR\_db=6; % giả thiết

N\_0=Eb/10.^(SNR\_db/10);

N=sqrt(N\_0/2)\*(randn(size(s))+j\*randn(size(s)));% nhiễu trắng phức cùng chiều dài với tín hiệu QPSK

R=s+N; % tín hiệu thu được

plot(R,'.');

hold on;

plot(s,'r\*');% vẽ biểu đồ chòm sao của tí nhiệu QPSK

hold on;

t=0:0.01:2\*pi;

plot(exp(j\*t),'r--');% vẽ đường tròn đơn vị, nét đứt, màu đỏ

legend('S\_m','S');

%S\_m chòm sao của tín hiệu thu được

% S chòm sao của tín hiệu QPSK

title('Bieu do chom sao tin hieu thu duoc');

xlabel('I');

ylabel('Q');

hold off;

Figure



**Bài số 5**

Mô phỏng hệ thống truyền dẫn số băng tần cơ sở QPSK :

Code

% Hàm giải điều chế QPSK và đếm lỗi bit xuất hiện trong quá trình phát, lưu với tên %'cha'

function y =cha(SNR\_db,S,x)

Es=var(S);

Eb=Es/2;

N\_0=Eb/10^(SNR\_db/10); % từ SNR\_db=10log(Eb/N\_0)

N0=sqrt(N\_0/2)\*(randn(size(S))+j\*randn(size(S)));% tạo nhiễu trắng phức

NS=S+N0;% tín hiệu thu được

theta\_m=[pi/4,3\*pi/4,5\*pi/4,7\*pi/4];

S\_m=exp(j\*theta\_m);% vòng lặp so sánh độ lệch của kí hiệu thu được

% với các giá trị kí hiệu chuẩn.

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 bít bằng 0

for i=1:length(x)

if R(i)~=x(i);

c=c+1;

end

end

y=c; % trả về y

end

%Main function:

clear all

load Bai4p1 s x% load file exp5p1\_Res đã được lưu bài 4.1

SNR\_db=0:2:8;

for i=1:length(SNR\_db)% vòng lặp xác định số lỗi bítứng với giá trị SNR[dB] khác nhau

c(i)=cha(SNR\_db(i),s,x);

end

BEP=c/length(x);% xác suất lỗi bít

semilogy(SNR\_db,BEP,'.--');% vẽ đồ thị nét '.--'

title('Do thi ti le loi bit voi ti le tin hieu tren nhieu');

xlabel('SNR[dB]');

ylabel('Pb');

save Bai5 c BEP;

Figure



**Bài số 6**

Tính tỉ lệ lỗi bit lý thuyết

Code

clear all;

SNR\_db=0:8;% SNR[dB] lý thuyết

SNR\_db\_mp=0:2:8;% SNR[dB] mô phỏng

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,'--')% vẽ đồ thị lý thuyết

hold on; % vẽ thêm hình

load Bai5 c BEP;% load hàm mô phỏng bài 5

semilogy(SNR\_db\_mp,BEP,'o--') % vẽ đồ thị mô phỏng

title('So sanh ket qua ly thuyet và mo phong');

xlabel('SNR[dB]');

ylabel('Pb');

legend('ket qua ly thuyet','ket qua mo phong');

Figure

