clear all;

close all;

clc;

%========= params de simulations

N\_tirage = 1e4; %nbre d'echantillons

SNR = [0:13];%dB

snr = 10.^(SNR/10);%SNR en val lineaire

s0 = -1; % signal pour le bit '0'

s1 = +1; % signal pour le bit '1'

p0 = 1/2 %probabilite s0

p1 = 1/2 %probabilite s1

p\_soi = 1; %puissance signal utile

sigma\_n = sqrt(p\_soi./snr); %ecart type du broit

%=======================

% recepteur optimal

%=======================

%barr

for i\_SNR =1:length(SNR)

Nb\_segment = 0; %nbre de segments TX

%Nb\_segment\_FA=0;

%Nb\_segment\_ND =0;

Nb\_erreur = 0; %nbre d'err total

%Nb\_erreur\_FA=0;

%Nb\_erreur\_ND =0;

temp\_Nb\_erreur = 0; %nbre d'err par seg

%temp\_Nb\_erreur\_FA=0;

%temp\_Nb\_erreur\_ND =0;

while Nb\_erreur < 500

Vecteur\_s = (-1).^random('Binomial',1,p0,[1, N\_tirage]);%generation soi

Vecteur\_n = sigma\_n(i\_SNR) \* randn(1,N\_tirage); %generation du bruit

Vecteur\_x = Vecteur\_s + Vecteur\_n;

x\_seuil(i\_SNR) = (((s0+s1)/2) + (log(p0/p1) / (s1-s0)) \* sigma\_n(i\_SNR)^2);%s

Vecteur\_s\_hat = (-1).^double(Vecteur\_x < x\_seuil(i\_SNR));%detecteur

temp\_Nb\_erreur = length(Vecteur\_s) - sum(Vecteur\_s\_hat == Vecteur\_s);%calcul

Nb\_erreur = Nb\_erreur + temp\_Nb\_erreur; %nbre d'err cumulé

Nb\_segment = Nb\_segment + 1; %compteur de seg

end

Perr\_hat(i\_SNR) = (Nb\_erreur) / (Nb\_segment \* N\_tirage);

%Pfa\_hat(i\_SNR) = (Nb\_erreur\_FA) / (Nb\_segment\_FA \* N\_tirage);

%Pnd\_hat(i\_SNR) = (Nb\_erreur\_ND) / (Nb\_segment\_ND \* N\_tirage);

Perr(i\_SNR) = p0 \* qfunc((x\_seuil(i\_SNR)-s0)/sigma\_n(i\_SNR)) + p1 \* qfunc((s1-x\_seuil(i\_SNR))/sigma\_n(i\_SNR));

%awaitbar(i\_SNR/length(SNR))

end

figure (1)

semilogy(SNR,Perr\_hat,'\*','Linewidth',2)

hold on

semilogy(SNR,Perr,'Linewidth',2)

xlabel('SNR [dB]'),ylabel('probabilite d''erreur')

legend('P\_(err) empirique','P\_(err) theorique')

axis([0 13 1e-6 1e0]),grid

%=======================

for i\_SNR =1:length(SNR)

Nb\_segment = 0; %nbre de segments TX

%Nb\_segment\_FA=0;

%Nb\_segment\_ND =0;

Nb\_erreur = 0; %nbre d'err total

%Nb\_erreur\_FA=0;

%Nb\_erreur\_ND =0;

temp\_Nb\_erreur = 0; %nbre d'err par seg

%temp\_Nb\_erreur\_FA=0;

%temp\_Nb\_erreur\_ND =0;

while Nb\_erreur\_FA < 500

Vecteur\_s = (-1).^random('Binomial',1,p0,[1, N\_tirage]);%generation soi

Vecteur\_n = sigma\_n(i\_SNR) \* randn(1,N\_tirage); %generation du bruit

Vecteur\_x = Vecteur\_s + Vecteur\_n;

x\_seuil(i\_SNR) = (((s0+s1)/2) + (log(p0/p1) / (s1-s0)) \* sigma\_n(i\_SNR)^2);%s

Vecteur\_s\_hat = (-1).^double(Vecteur\_x < x\_seuil(i\_SNR));%detecteur

temp\_Nb\_erreur = length(Vecteur\_s) - sum(Vecteur\_s\_hat == Vecteur\_s);%calcul

Nb\_erreur = Nb\_erreur + temp\_Nb\_erreur; %nbre d'err cumulé

Nb\_segment = Nb\_segment + 1; %compteur de seg

end

Perr\_hat(i\_SNR) = (Nb\_erreur) / (Nb\_segment \* N\_tirage);

%Pfa\_hat(i\_SNR) = (Nb\_erreur\_FA) / (Nb\_segment\_FA \* N\_tirage);

%Pnd\_hat(i\_SNR) = (Nb\_erreur\_ND) / (Nb\_segment\_ND \* N\_tirage);

Perr(i\_SNR) = p0 \* qfunc((x\_seuil(i\_SNR)-s0)/sigma\_n(i\_SNR)) + p1 \* qfunc((s1-x\_seuil(i\_SNR))/sigma\_n(i\_SNR));

%awaitbar(i\_SNR/length(SNR))

end

figure (1)

semilogy(SNR,Perr\_hat,'\*','Linewidth',2)

hold on

semilogy(SNR,Perr,'Linewidth',2)

xlabel('SNR [dB]'),ylabel('probabilite d''erreur')

legend('P\_(err) empirique','P\_(err) theorique')

axis([0 13 1e-6 1e0]),grid