**BPSK with AWGN channel**

clc;

clear all;

close all;

num\_bit=1000000;%number of bit

data=randi ([0,1], 1,num\_bit); %random bit generation (1 or 0)

s=2\*data-1;%conversion of data for BPSK modulation

SNRdB=0:10; % SNR in dB

SNR=10.^(SNRdB/10);

for k=1:length(SNRdB) %BER (error/bit) calculation for different SNR

y=awgn(complex(s),SNRdB(k));

error=0;

for c=1:1:num\_bit

if (y(c)>0&&data(c)==0) || (y(c)<0&&data(c)==1)%logic acording to BPSK

error=error+1;

end

end

error=error/num\_bit %Calculate error/bit

m(k)=error

end

figure(1)

%plot start

semilogy(SNRdB,m,'\*','linewidth',2.5),grid on,hold on;

BER\_th=(1/2)\*erfc(sqrt(SNR));

semilogy(SNRdB,BER\_th,'r','linewidth',2.5);

title(' curve for Bit Error Rate verses SNR for Binary PSK modulation');

xlabel(' SNR(dB)');

ylabel('BER');

legend('simulation','theorytical')

axis([0 10 10^-5 1]);

**BASK with AWGN channel**

clc;

clear all;

close all;

num\_bit=10000;%number of bit

data=randi ([0,1], 1,num\_bit);%random bit generation (1 or 0)

SNRdB=0:10; % SNR in dB

SNR=10.^(SNRdB/10);

for(k=1:length(SNRdB))%BER (error/bit) calculation for different SNR%

y=awgn(complex(data),SNRdB(k));

error=0;

R=0;

M=[];

for(c=1:1:num\_bit)

if (y(c)>.5&&data(c)==0)||(y(c)<.5&&data(c)==1)%logic acording to BASK

error=error+1;

M=[M ~data(c)];

else

M=[M data(c)];

end

end

error=error/num\_bit; %Calculate error/bit

m(k)=error;

end

semilogy(SNRdB,m,'o','linewidth',2.5),grid on,hold on;

BER\_th=(1/2)\*erfc(.5\*sqrt(SNR));

semilogy(SNRdB,BER\_th,'r','linewidth',2.5),grid on,hold on;

title(' curve for Bit Error Rate verses SNR for Binary ASK modulation');

xlabel(' SNR(dB)');

ylabel('BER');

legend('simulation','theorytical')

axis([0 10 10^-5 1]);

**BFSK with AWGN channel**

clear all; close all; clc;

num\_bit=10000; %Signal length

max\_run=20; %Maximum number of iterations for a single SNR

Eb=1; %Bit energy

SNRdB=0:1:10; %Signal to Noise Ratio (in dB)

SNR=10.^(SNRdB/10);

hand=waitbar(0,'Please Wait....');

for count=1:length(SNR) %Beginning of loop for different SNR

avgError=0;

No=Eb/SNR(count); %Calculate noise power from SNR

for run\_time=1:max\_run %Beginning of loop for different runs

waitbar((((count-1)\*max\_run)+run\_time-1)/(length(SNRdB)\*max\_run));

Error=0;

data=randi ([0,1], 1,num\_bit); %Generate binary data source

s=data+j\*(~data); %Baseband BFSK modulation

NI=sqrt(No/2)\*randn(1,num\_bit);

NQ=sqrt(No/2)\*randn(1,num\_bit);

N=NI+j\*NQ; %Generate complex AWGN

Y=s+N; %Received Signal

for k=1:num\_bit %Decision device taking hard decision and deciding error

Z(k)=real(Y(k))-imag(Y(k));

if ((Z(k)>0 && data(k)==0)||(Z(k)<0 && data(k)==1))

Error=Error+1;

end

end

Error=Error/num\_bit; %Calculate error/bit

avgError=avgError+Error; %Calculate error/bit for different runs

end %Termination of loop for different runs

BER\_sim(count)=avgError/max\_run; %Calculate BER for a particular SNR

end %Termination of loop for different SNR

BER\_th=(1/2)\*erfc(sqrt(SNR/2)); %Calculate analytical BER

close(hand);

semilogy(SNRdB,BER\_th,'k'); %Plot BER

hold on

semilogy(SNRdB,BER\_sim,'k\*');

legend('Theoretical','Simulation',3);

axis([min(SNRdB) max(SNRdB) 10^(-5) 1]);

hold off