DC AHP-10

Name: C.P.Sindhu

SRN: PES1UG21EC071

Semester: 4th

Section: 'B'

Code:

```
% To modulate and demodulate DBPSK constellation signals, and to compare
% theoretical and simulated BER values

clc;
clear all;
close all;

%——Input Fields————————
Nbit = 1e5;                          % number of bits ie 10^5
EbN0dB = 0:1:10;                     % multiple Eb/N0 values in dB
k=1;                                 % Number of bits per symbol
EsN0dB = EbN0dB + 10*log10(k);       % For BPSK, EbN0dB = EsN0dB
a=[ones(1,1);zeros(1,1)];            % To make a=[0 1]' for BPSK symbols
constellation = exp(i*2*pi.*a/2);    % For 1+0j and -1+0j as complex double

%-------Input signal----------
input=randi([0, 1], 1,Nbit);

%–generating differential symbols
%Initially we transmit 0 and then use the input bits and current input to get the
differential symbols
mod=zeros(1,Nbit+1);                 % Size of differential symbols, initialize all
0
for i=1:Nbit
    mod(i+1)=xor(mod(i),input(i));   % To take xor of past signal with current
input
end
Nbit=Nbit+1;

%--------Input constellation---------
input_mod=constellation(mod+1);      % This modulated signal is transmitted

length_snr=length(EbN0dB);           % Total number of Eb/No (dB) values
perr_estimate=zeros(1,length_snr);   % Initializing BER estimate to zeros
for x = 1:length_snr                 % For each SNR or Eb/No value
    snr_now = EbN0dB(x);             % Current SNR value
    ebno=10^(snr_now/10);            % Eb/No value in linear scale
    sigma=sqrt(1/(ebno));            % Set the noise variance (or signal strength)
accordingly
    received = input_mod + sigma*randn(Nbit,1)+1i*sigma*randn(Nbit,1);  % Received
signal with AWGN
    decisions=zeros(1,Nbit);         % Initialize all decisions to zero
    for n=1:Nbit                     % For each symbol
        distances = abs(received(n)-constellation);     % Calculate distance
```

```
        [min_dist,decisions(n)] = min(distances);       % Minimum distance
receiver
    end

    for n=1:Nbit
       if decisions(n)==1
           decisions(n)=0;  % Replace all 1 with 0, since BPSK can only have 1 or
-1
       end
    end
    for n=1:Nbit
       if decisions(n)==2
           decisions(n)=1;  % Replace all 2 with 1, since BPSK can only have 1 or
-1
       end
    end

    % To differentially decode the string using another loop and xor gate.
    decoded=zeros(1,Nbit);  % Initialize as all zeros
    for d=2:Nbit
        decoded(d)=(xor(decisions(d),decisions(d-1))); %It compares past decided
                        % binary character with current character to get the
decoded character
    end  % For example : if current decided character is 1
                        % and previous was 0 then the decoded string shall have 1
    decoded=decoded(2:end); % As the starting character of transmitted string was
0 (System initialization)
                        % it was not part of origional signal so we remove that.
    errors = (decoded ~=input );    % All decoded bits different from input are
errors and stored in errors.
    perr_estimate(x) = sum(errors)/Nbit; % This gives BER.
end


%------Plotting commands---------
%using Eb/No (dB)
figure
semilogy(EbN0dB,perr_estimate);     % To plot the BER per nsymbols with SNR as
Eb/No.
hold on;                            % To add both data in the same plot
semilogy(EbN0dB,0.5*exp(-(10.^(EbN0dB/10)))); % To plot BER theoretical using Q-
function .
legend("Experimental BER", "Theoretical BER"); % To all legend
xlabel("Eb/N0(dB)");                % To add SNR label to x axis
ylabel("BER (Bit Error Rate)");     % To add BER label to y axis. its BER per
symbol.
title("DBPSK Bit Error Plot with Eb/N0 using "+string(Nbit-1)+"bits")
hold off;                           % Note the difference between theoretical and
simulated BER
```

Output:

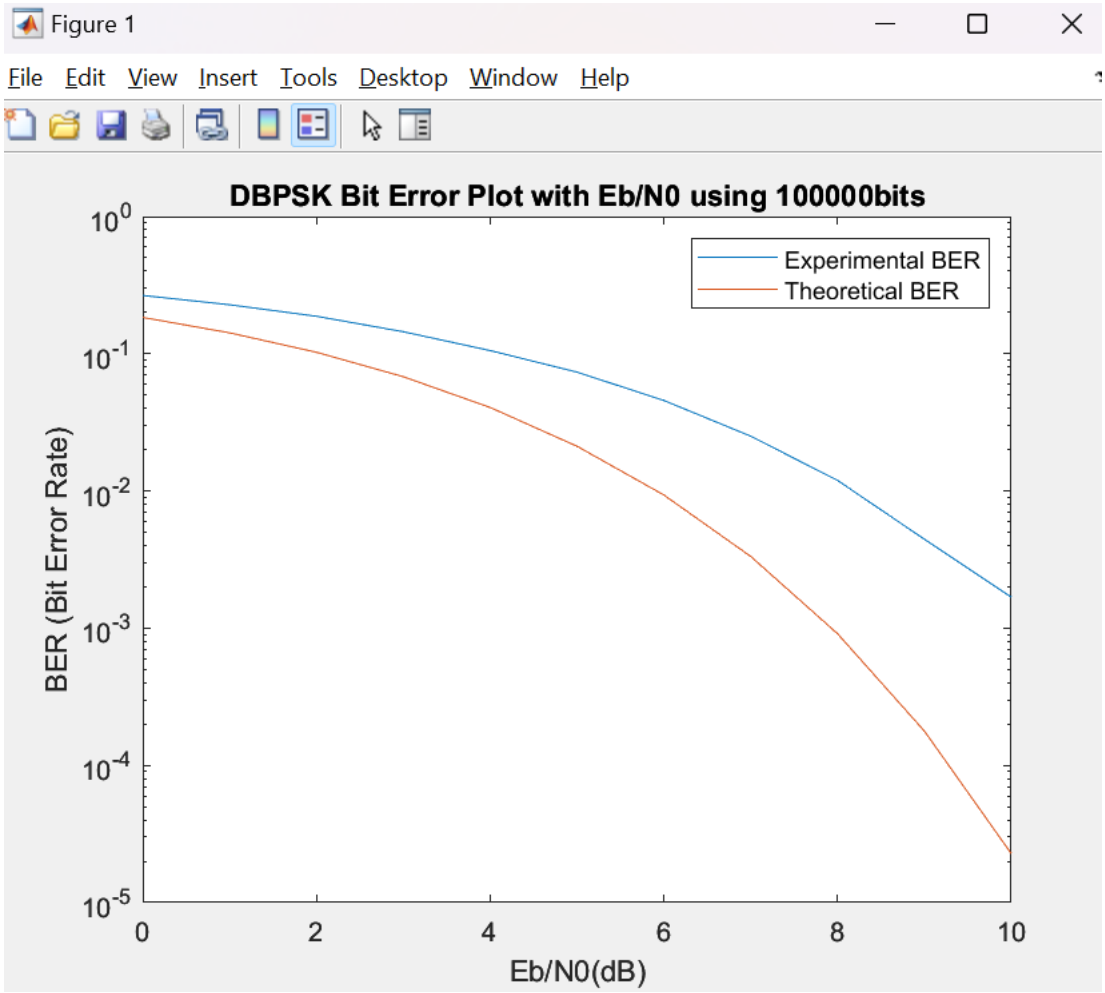
Figure 1
File Edit View Insert Tools Desktop Window Help
DBPSK Bit Error Plot with Eb/N0 using 100000bits
Experimental BER
Theoretical BER
BER (Bit Error Rate)
Eb/N0(dB)
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
0
2
4
6
8
10