GEBZE TECHNICAL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF ELECTRONICS ENGINEERING

ELEC365

FUNDAMENTALS OF DIGITAL COMMUNICATIONS

MATLAB PROJECT

NAME-SURNAME	UMUT MEHMET ERDEM
STUDENT NUMBER	200102002025

1. Introduction

This assignment aims to understand and interpret baseband communications. As per the homework question, two situations are examined for the baseband communication system: bits with equal probability and bits with unequal probability. In this context, it is aimed to find theoretical values by making analytical inferences of the system and to compare and interpret the two systems by obtaining bit error probability curves with these values in the MATLAB simulation environment.

2. Problems

Operations were carried out in the MATLAB environment according to the a1, a2, σ0^2, Eh, Es1, Es2 and γ0 values in the analytical section as shown in Table 1. Pb formulas, which differ according to equal and different probability values, were used in options a and b for theoretical BER graphs. Accordingly, a BER graph as in Figure 1 was found in option a. In option B, a BER graph as shown in Figure 2 was found.

For Bits with Equal Probability		For the Case P(1) = 1/4, P(0) = 3/4	
a_1	1	a_1	1
a_2	-1	a_2	-1
E_{S1}	1	E_{S1}	1
E_{S2}	1	E_{S2}	1
E_h	2	E_h	2
$σ_0^2$	N_0	$σ_0^2$	N_0
E_b	1	E_b	1
$γ_0$	0	$γ_0$	$0.5493*N_0$
P_b	$Q(1/\sqrt{N_0})$	P_b	$[1-Q(\frac{0.55N_0-1}{\sqrt{N_0}})]\times\frac{1}{4}+Q(\frac{0.55N_0+1}{\sqrt{N_0}})\times\frac{3}{4}$

Table 1. Theoretically Calculated Values

a)

Matlab Code For Option a

```
% ELM365 MATLAB PROJECT - Umut Mehmet Erdem - 200102002025

% option a -> for equal probability
syms x; N0 = [];
SNR = 0:15; % SNRdb values

%{
Using the solve command according to the given SNR values,
N0 values are found in vector form for 10*log10(1/N0) = 0-15.
%}
for i=0:15
    N0= double([N0, solve(10.*log10(1./x)== i, x)]);
end

% The amount of bits to be produced for each SNR value N = 10,000,000
N = 1e7;

% Pb = Q(1/√N0), is found with the qfunc function.
Pb_theoretical = qfunc(sqrt(1./N0));
```

```
%{
With the randi function, bits were obtained for each SNRdb value in a 16x10^7 matrix
size with uniform distribution and equal probability, giving a value of 0 or 1.
%}
bits = randi([0, 1], 16, N);
%{
In the ai variable, according to the theoretically found a1=1 and a2=-1 values, the 0
and 1 values created in the bits variable are given -1 value instead of 0. The
created variable ai contains equally probable values 1 and -1 and has a size of 16x10
martis.
%}
ai = ones(16, N);
ai(find(bits == 0)) = -1;
%{
The randn function was used when calculating n0 Gaussian noise for each SNR value.
randn function is a function that produces random values with mean 0 and standard
deviation 1. Since the variance we found theoretically is σ0^2=N0, its standard
deviation is σ0=√N0. Accordingly, when calculating n0 gaussian noise, its standard
deviation should be √N0. To do this, based on the random variable equation Y=α*X, it
can be said that μy = α*μx and σy^2 = α^2*σx^2. Accordingly, α=√N0 was found and the
equation n0 = √N0*randn() was used. The n0 gaussian noise variable created is a
matrix of size 16x10^7.
%}
n0 = [];
for i=1:16
    n0 = [n0; sqrt(N0(i)).*randn(1,N)];
end
%{
With the created matrices ai and n0, z = ai+n0 was created. z is a 16x10^7 matrix.
%}
z = ai + n0;
gama = 0; % theoretically, γ0(gamma) = 0.
bit_err = [];
%{
After the z matrix is found, the decision process is performed for z>γ0(gamma) and
z<γ0(gamma). According to this; It has been determined that the bit is transmitted
incorrectly when the values of the ai variable at that index are -1 for the indices
where the z>γ0(gamma) condition is true, and when the values of the ai variable at
that index are 1 for the indices where the z<γ0(gamma) condition is true. This
process was found for each of the SNRdb values (0-15) and the total erroneous bits
were written to the bit_err variable. Thus, the bit_err variable was created in
the form of a 16x1 vector.
%}
for i=1:16
    h1 = z(i,:); h2= ai(i,:);
    cond1 = sum(h2(find(h1>gama))==-1);
    cond2 = sum(h2(find(h1<gama))==1);
    bit_err = [bit_err; (cond1+cond2)];
end

semilogy(SNR, Pb_theoretical, "bo-",LineWidth=1.5);
hold on;
semilogy(SNR, bit_err./N, 'r--',LineWidth=1.5);
hold off;
legend("Theoretical BER curve","Simulation BER curve");
xlabel("SNR (db)");
ylabel("Pb");
title('P(1)=P(0)=1/2 | Theoretical BER curve vs. Simulation BER curve | Umut Mehmet
Erdem-200102002025');
```

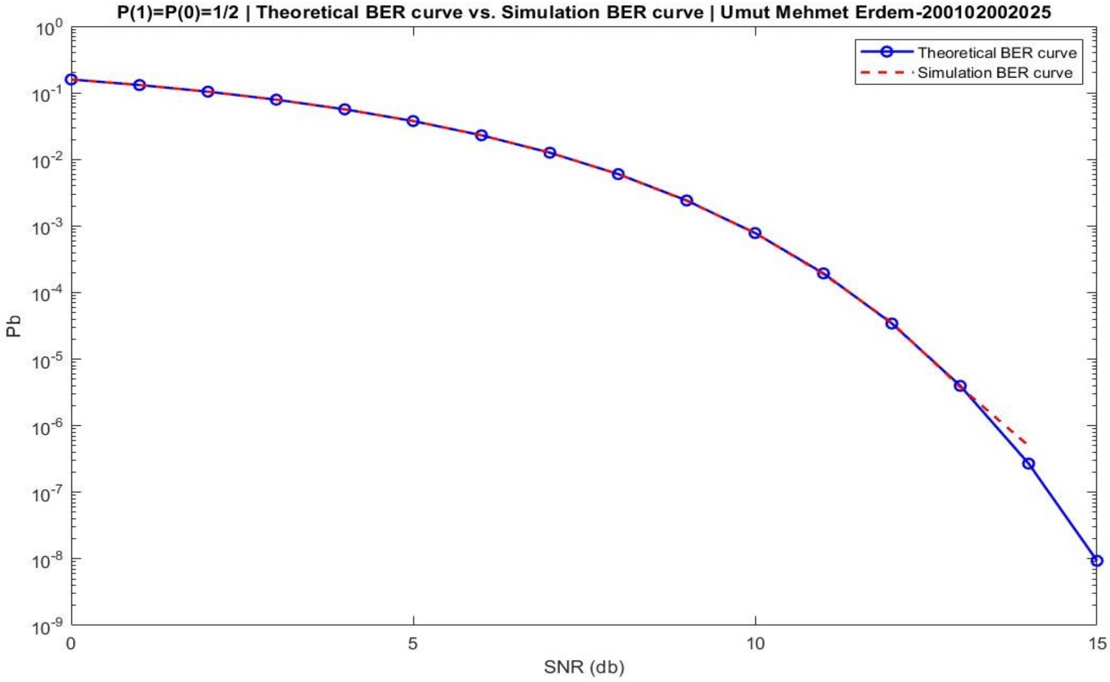

Figure 1. Graph Of Theoretical And Simulated BER Curves For Option a

b)

Matlab Code For Option b

```
%% option b -> for P(1)=1/4, P(0)=3/4
syms x; N0 = [];
SNR = 0:15; % SNRdb values

%{
Using the solve command according to the given SNR values,
N0 values are found in vector form for 10*log10(1/N0) = 0-15.
%}
for i=0:15
    N0= double([N0, solve(10.*log10(1./x)== i, x)]);
end

% The amount of bits to be produced for each SNR value N = 10,000,000
N = 1e7;

%{
theoretically, Pb = [1-Q((0,5493*N0-1)/√N0)]*(1/4) +
Q((0,5493*N0+1)/√N0)*(3/4) was found and the Q operation was performed
with the qfunc function.
%}
Pb_theoretical = (1 - qfunc((0.55*N0 - 1) ./ sqrt(N0))) * 1/4 +
qfunc((0.55*N0 + 1) ./ sqrt(N0))*3/4;

%{
With the randsrc function, bits were obtained for each SNRdb value in a
matrix size of 16x10^7, which randomly gives the value 1 with a
probability of 1/4 and the value of 0 with a probability of 3/4.
%}
bits = randsrc(16, N, [1,0;1/4,3/4]);
```

```matlab
%{
In the ai variable, according to the theoretically found a1=1 and a2=-1 values, the 0
and 1 values created in the bits variable are given -1 value instead of 0. The
created variable ai contains the values 1 and -1 with different probabilities and has
a size of 16x10 martis.
%}
ai = ones(16, N);
ai(find(bits == 0)) = -1;

%{
The randn function was used when calculating n0 Gaussian noise for each SNR value.
randn function is a function that produces random values with mean 0 and standard
deviation 1. Since the variance we found theoretically is σ0^2=N0, its standard
deviation is σ0=√N0. Accordingly, when calculating n0 gaussian noise, its standard
deviation should be √N0. To do this, based on the random variable equation Y=α*X, it
can be said that μy = α*μx and σy^2 = α^2*σx^2. Accordingly, α=√N0 was found and the
equation n0 = √N0*randn() was used. The n0 gaussian noise variable created is a
matrix of size 16x10^7.
%}
n0 = [];
for i=1:16
    n0 = [n0; sqrt(N0(i)).*randn(1,N)];
end

%{
With the created matrices ai and n0, z = ai+n0 was created. z is a 16x10^7 matrix.
%}
z = ai + n0;
% theoretically, γ0(gamma) = 0.5493*N0.
% γ0(gamma) has a matrix size of 1x16.
gama = 0.55*N0;

%{
After the z matrix is found, the decision process is performed for z>γ0(gamma) and
z<γ0(gamma). According to this; It has been determined that the bit is transmitted
incorrectly when the values of the ai variable at that index are -1 for the indices
where the z>γ0(gamma) condition is true, and when the values of the ai variable at
that index are 1 for the indices where the z<γ0(gamma) condition is true. This
process was found provided that the γ0(gamma) value was different for each SNRdb
values (0-15) and the total erroneous bits were written to the bit_err variable.
Thus, the bit_err variable was created in the form of a 16x1 vector.
%}
bit_err = [];
for i=1:16
    h1 = z(i,:); h2= ai(i,:);
    cond1 = sum(h2(find(h1>gama(i)))==-1);
    cond2 = sum(h2(find(h1<gama(i)))==1);
    bit_err = [bit_err; (cond1+cond2)];
end

semilogy(SNR, Pb_theoretical, "b--",LineWidth=1.5);
hold on;
semilogy(SNR, bit_err./N, 'go-',LineWidth=1.5);
hold off;
legend("Theoretical BER curve","Simulation BER curve");
xlabel("SNR (db)");
ylabel("Pb");
title('P(1)=1/4 and P(0)=3/4 | Theoretical BER curve vs. Simulation BER curve | Umut
Mehmet Erdem-200102002025');
```

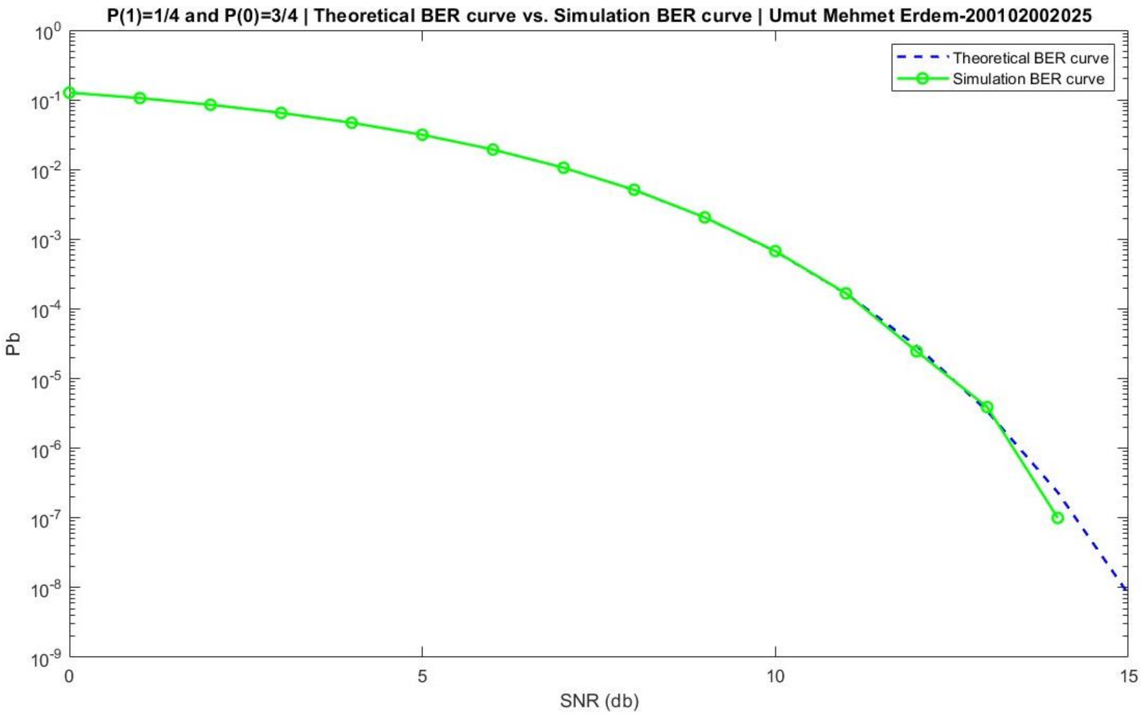

Figure 2. Graph Of Theoretical And Simulated BER Curves For Option b

3. Conclusion

Within the scope of the assignment, the skills of making analytical inferences of the data that should be used in a baseband communication and transferring the inference results to the simulation environment were acquired. When Figure 1 and Figure 2 are compared, bit error probability graphs with the same characteristics are observed. Moreover, according to the analytical calculations made for both graphs, the fact that the theoretical and simulation BER curves are similar shows that the calculations are correct.